Équipe d'ingénierie de l'Internet ((IETF)
Request for Comments : 7143
RFC rendues obsolètes : 3720, 3980, 4850, 5048
RFC mise à jour : 3721
Catégorie : En cours de normalisation
ISSN: 2070-1721

M. Chadalapaka, Microsoft
J. Satran, Infinidat Ltd.
K. Meth, IBM
D. Black, EMC
avril 2014
Traduction Claude Brière de L'Isle

Protocole (consolidé) d'interface Internet de système de petit ordinateur (iSCSI)

Résumé
Le présent document décrit un protocole de transport pour SCSI qui fonctionne par dessus TCP. Le protocole iSCSI vise à être pleinement conforme au modèle d'architecture SCSI normalisé (SAM-2, *SCSI Architecture Model*). La RFC 3720 définissait le protocole iSCSI d'origine. La RFC 3721 expose des exemples de dénomination iSCSI et des techniques de découverte. Par la suite, la RFC 3980 a ajouté un format de dénomination supplémentaire au protocole iSCSI. La RFC 4850 a suivi en ajoutant une nouvelle clé d'extension publique à iSCSI. La RFC 5048 a proposé un certain nombre de précisions ainsi que des améliorations et corrections au protocole iSCSI original.

Le présent document rend obsolètes les RFC 3720, 3980, 4850, et 5048 en les consolidant en un seul document et en faisant des mises à jour supplémentaires à la spécification consolidée. Le présent document met aussi à jour la RFC 3721. Le texte du présent document se substitue donc au texte de toutes les RFC notées chaque fois qu'il y a une différence de sémantique.

Statut de ce mémoire
Ceci est un document de l'Internet en cours de normalisation.

Le présent document a été produit par l'équipe d'ingénierie de l'Internet (IETF). Il représente le consensus de la communauté de l'IETF. Il a subi une révision publique et sa publication a été approuvée par le groupe de pilotage de l'ingénierie de l'Internet (IESG). Plus d'informations sur les normes de l'Internet sont disponibles à la Section 2 de la [RFC5741].

Les informations sur le statut actuel du présent document, tout errata, et comment fournir des réactions sur lui peuvent être obtenues à http://www.rfc-editor.org/info/rfc7143

Table des matières

1. Introduction

Interface de petits systèmes informatiques (SCSI, *Small Computer Systems Interface*) est une famille populaire de protocoles pour communiquer avec des appareils d'entrée/sortie, en particulier des appareils de mémorisation/stockage. SCSI est une architecture client-serveur. Les clients d'une interface SCSI sont appelés "initiateurs". Les initiateurs produisent des "commandes" SCSI pour demander des services à des composants, unités logiques d'un serveur qu'on appelle une "cible". Un "transport SCSI" transpose le protocole client-serveur SCSI en une interconnexion spécifique. Un initiateur est un point d'extrémité d'un transport SCSI et une cible est l'autre point d'extrémité.

Le protocole SCSI a été transposé sur divers transports, y compris SCSI parallèle, interface périphérique intelligente (IPI, *Intelligent Peripheral Interface*), IEEE-1394 (firewire) et canal fibre. Ces transports sont spécifiques d'entrée/sortie et ont des capacités de distance limitées.

Le protocole iSCSI défini dans le présent document décrit les moyens de transporter des paquets SCSI sur TCP/IP, assurant une solution interopérable qui peut tirer parti de l'infrastructure Internet existante, des facilités de gestion de l'Internet, et des limitations en distance des adresses.

2. Acronymes, définitions, et résumé du document

2.1 Acronymes

Acronyme	Développement	Signification
3DES	Triple Data Encryption Standard	Norme de triple chiffrement des données
ACA	Auto Contingent Allegiance	Obéissance auto-limitée
AEN	Asynchronous Event Notification	notification d'événement asynchrone
AES	Advanced Encryption Standard	Norme de chiffrement évolué
AH	Additional Header (pas l'AH IPsec !)	En-tête supplémentaire
AHS	Additional Header Segment	Segment d'en-tête supplémentaire
API	Application Programming Interface	Interface de programmation d'application
ASC	Additional Sense Code	Code de sens supplémentaire
ASCII	American Standard Code for Information Interchange	Code standard américain pour les échanges d'information
ASCQ	Additional Sense Code Qualifier	Qualificatif de code de sens supplémentaire
ATA	AT Attachment	Rattachement AT
BHS	Basic Header Segment	Segment d'en-tête de base
CBC	Cipher Block Chaining	Chaînage de bloc de chiffrement
CD	Compact Disk	Disque compact
CDB	Command Descriptor Block	Bloc descripteur de commande
CHAP	Challenge Handshake Authentication Protocol	Protocole d'authentification par dialogue à énigme
CID	Connection ID	Identifiant de connexion
CO	Connection Only	Connexion seule
CRC	Cyclic Redundancy Check	Contrôle de redondance cyclique
CRL	Certificate Revocation List	Liste de révocation de certificats
CSG	Current Stage	Étape en cours
CSM	Connection State Machine	Automate à états de connexion
DES	Data Encryption Standard	Norme de chiffrement des données
DNS	Domain Name Server	Serveur de noms de domaine
DOI	Domain of Interpretation	Domaine d'interprétation,
DVD	Digital Versatile Disk	Disque numérique enregistrable
EDTL	Expected Data Transfer Length	Longueur attendue du transfert de données
ESP	Encapsulating Security Payload	Encapsulation de charge utile de sécurité
EUI	Extended Unique Identifier	Identifiant univoque étendu
FFP	Full Feature Phase	Phase de pleines caractéristiques
FFPO	Full Feature Phase Only	Phase de pleines caractéristiques seules
HBA	Host Bus Adapter	Adaptateur de bus d'hôte
HMAC	Hashed Message Authentication Code	Code d'authentification de message par hachage de clé
I_T	Initiator_Target	Initiateur_Cible
I_T_L	Initiator_Target_Logical-Unit-Number	Numéro-d'unité-logique_d'initiateur_cible
IANA	Internet Assigned Numbers Authority	Autorité d'allocation des numéros de l'Internet
IB	InfiniBand	Bande infinie
ID	Identifier	Identifiant
IDN	Internationalized Domain Name	Nom de domaine internationalisé
IEEE	Institute of Electrical & Electronics Engineers	Institut des ingénieurs en électricité et électronique
IETF	Internet Engineering Task Force	Équipe d'ingénierie de l'Internet
IKE	Internet Key Exchange	Échange de clés Internet
I/O	Input – Output	Entrée/sortie
IO	Initialize Only	En initialisation seule
IP	Internet Protocol	Protocole Internet
IPsec	Internet Protocol Security	Sécurité du protocole Internet
IPv4	Internet Protocol Version 4	Protocole Internet version 4
IPv6	Internet Protocol Version 6	Protcole Internet version 6
IQN	iSCSI Qualified Name	Nom qualifié iSCSI
iSCSI	Internet SCSI	SCSI Internet
iSER	iSCSI Extensions for RDMA (voir [RFC7145])	Extensions iSCSI pour RDMA
ISID	Initiator Session ID	Identifiant d'initiateur de session
iSNS	Internet Storage Name Service (voir [RFC4171])	Service de nom de mémorisation Internet

ITN	iSCSI Target Name	Nom de cible iSCSI
ITT	Initiator Task Tag	Étiquette de tâche d'initiateur
KRB5	Kerberos V5	Kerberos version 5
LFL	Lower Functional Layer	Couche fonctionnelle inférieure
LTDS	Logical-Text-Data-Segment	Segment logique de texte/données
LO	Leading Only	Seulement en tête
LU	Logical Unit	Unité logique
LUN	Logical Unit Number	Numéro d'unité logique
MAC	Message Authentication Code	Code d'authentification de message
NA	Not Applicable	Non applicable
NAA	Network Address Authority	Autorité des adresses du réseau
NIC	Network Interface Card	Carte d'interface réseau
NOP	No Operation	Non fonctionnement
NSG	Next Stage	Prochaine étape
OCSP	Online Certificate Status Protocol	Protocole d'état de certificat en ligne
OS	Operating System	Système d'exploitation
PDU	Protocol Data Unit	Unité de données de protocole
PKI	Public Key Infrastructure	Infrastructure de clé publique
R2T	Ready To Transfer	Prêt au transfert
R2TSN	Ready To Transfer numéro de séquence	Numéro de séquence de Prêt au transfert
RDMA	Remote Direct Memory Access	Accès mémoire direct à distance
RFC	Request For Comments	Appel à commentaires
SA	Security Association	Association de sécurité
SAM	SCSI Architecture Model	Modèle d'architecture SCSI
SAM2	SCSI Architecture Model - 2	Modèle 2 d'architecture SCSI
SAN	Storage Area Network	réseau à zone de mémorisation
SAS	Serial Attached SCSI	SCSI rattachée en série
SATA	Serial AT Attachment	Rattachement AT en série
SCSI	Small Computer Systems Interface	Interface de petits systèmes informatiques
SLP	Service Location Protocol	Protocole de localisation de service
SN	numéro de séquence	Numéro de séquence
SNACK	Selective Negative Acknowledgment	Accusé de non réception sélectif - aussi accusé de réception de numéro de séquence pour des données
SPDTL	SCSI-Presented Data Transfer Length	Longueur de transfert de données présentées à la SCSI
SPKM	Simple Public-Key Mechanism	Mécanisme simple de clé publique
SRP	Secure Remote Password	Mot de passe distant sécurisé
SSID	Session ID	Identifiant de session
SW	Session Wide	Pour toute la session
TCB	Task Control Block	Bloc de contrôle de tâche
TCP	Transmission Control Protocol	Protocole de contrôle de transmission
TMF	Task Management Function	Fonction de gestion de tâches
TPGT	Target Portal Group Tag	Étiquette de groupe de portail cible
TSIH	Target Session Identifying Handle	Descripteur identifiant de session cible
TTT	Target Transfer Tag	Étiquette de transfert de cible
UA	Unit Attention	
UFL	Upper Functional Layer	Couche fonctionnelle supérieure
ULP	Upper Level Protocol	Protocole de niveau supérieur
URN	Uniform Resource Name [RFC2396]	Nom de ressource universel
UTF	Universal Transformation Format	Format de transformation universel
WG	Working Group	Groupe de travail

2.2 Définitions

Alias : Une chaîne d'alias peut aussi être associée à un nœud iSCSI. L'alias permet à une organisation d'associer au nom iSCSI une chaîne facilement mémorisable. Cependant, la chaîne d'alias n'est pas un substitut du nom iSCSI.

Identifiant de connexion (CID) : Au sein d'une session, les connexions sont identifiées par un identifiant de connexion. C'est un identifiant univoque pour cette connexion au sein de la session pour l'initiateur. Il est généré par l'initiateur et présenté à la cible durant les demandes de connexion et durant les déconnexions qui closent les connexions.

Connexion : Une connexion est une connexion TCP. La communication entre l'initiateur et la cible se fait sur une ou plusieurs connexions TCP. Les connexions TCP portent des messages de contrôle, des commandes SCSI, des

paramètres, et des données au sein des unités de données de protocole iSCSI (PDU iSCSI, *iSCSI Protocol Data Unit*).

Mémoire tampon I/O : utilisée dans une opération SCSI de lecture ou écriture de façon que les données de la SCSI puissent être envoyées ou reçues par cette mémoire tampon. Pour qu'un transfert de données en lecture ou écriture ait lieu pour une tâche, une mémoire tampon I/O est nécessaire chez l'initiateur et il en est exigé au moins une chez la cible.

INCITS *(InterNational Committee for Information Technology Standards)* : Le comité international pour les normes des technologies de l'information (INCITS) a un large domaine de normalisation dans le secteur des technologies de l'information et des communications (ICT), englobant la mémorisation, le traitement, le transfert, l'affichage, la gestion, l'organisation, et la restitution de l'information. L'INCITS sert de groupe de conseil technique de l'ANSI pour le comité technique conjoint n° 1 de l'ISO/CEI (JTC 1). Voir <http://www.incits.org >.

InfiniBand : InfiniBand est une architecture I/O destinée à l'origine à remplacer l'interconnectivité d'interconnexion de composants périphériques (PCI, *Peripheral Component Interconnect*) et de serveur d'adresse hautes performances [IB].

Appareil iSCSI : C'est un appareil SCSI qui utilise un sous-système de livraison de service iSCSI. Un sous-système de livraison de service est défini par [SAM2] comme un mécanisme de transport pour les commandes et réponses SCSI.

Nom d'initiateur iSCSI : Il spécifie le nom unique au monde de l'initiateur.

Nœud initiateur iSCSI : C'est "l'initiateur". Le mot "initiateur" a été bien qualifié comme un accès ou un appareil dans le reste de ce document lorsque le contexte est ambigu. Toute utilisation non qualifiée de "initiateur" se réfère à un accès (ou appareil) initiateur selon le contexte.

Couche iSCSI : Cette couche construit/reçoit des PDU iSCSI et les relaye/reçoit de/vers une ou plusieurs connexions TCP qui forment une "session" initiateur-cible.

Nom iSCSI : C'est le nom d'un initiateur iSCSI ou d'une cible iSCSI.

Nœud iSCSI : Il représente un seul initiateur ou cible iSCSI. Il y a un ou plusieurs nœuds iSCSI au sein d'une entité réseau. Le nœud iSCSI est accessible via un ou plusieurs portails réseau. Un nœud iSCSI est identifié par son nom iSCSI. La séparation du nom iSCSI des adresses utilisées par et pour le nœud iSCSI permet que plusieurs nœuds iSCSI utilisent la même adresse, et que le même nœud iSCSI utilise plusieurs adresses.

Nom de cible iSCSI : Il spécifie le nom unique au monde de la cible.

Nœud cible iSCSI : C'est l'appareil "cible". Le mot "cible" a été qualifié de façon appropriée comme un accès ou un appareil dans le reste de ce document lorsque le contexte est ambigu. Toutes les utilisations non qualifiées de "cible" se réfèrent à un accès cible (ou appareil) selon le contexte.

Tâche iSCSI : C'est une demande iSCSI pour laquelle une réponse est attendue.

Direction de transfert iSCSI : Elle est définie par rapport à l'initiateur. Les transferts sortants sont des transferts de l'initiateur à la cible, tandis que les transferts entrants sont de la cible à l'initiateur.

ISID : Partie initiateur de l'identifiant de session. Elle est explicitement spécifiée par l'initiateur durant l'établissement de connexion.

Lien I_T : Selon [SAM2], le lien I_T est une relation entre un accès d'initiateur SCSI et un accès cible SCSI. Pour iSCSI, cette relation est une session, définie comme une relation entre l'extrémité de la session de l'initiateur iSCSI (accès d'initiateur SCSI) et le groupe portail de la cible iSCSI. Le I_T lien peut être identifié par la conjonction des noms d'accès SCSI ; c'est-à-dire que l'identifiant de I_T lien est le tuplet (Nom d'initiateur iSCSI + ',i,' + ISID, nom de cible iSCSI + ',t,' + étiquette de groupe portail).

Nexus I_T_L : C'est un concept SCSI qui est défini comme la relation entre un accès d'initiateur SCSI, un accès cible SCSI, et une unité logique (LU, *Logical Unit*).

NAA *(Network Address Authority)* : "NAA" se réfère à une autorité d'adresse réseau, un format de désignation défini par les protocoles de canal fibre T11 de l'INCITS [FC-FS3].

Entité réseau : Elle représente un appareil ou passerelle qui est accessible à partir du réseau IP. Une entité réseau doit avoir un ou plusieurs portails réseau, dont chacun peut être utilisé pour obtenir l'accès au réseau IP par des nœuds iSCSI contenus dans cette entité réseau.

Portail réseau : C'est un composant d'une entité réseau qui a une adresse réseau TCP/IP et qui peut être utilisé par un nœud iSCSI au sein de cette entité réseau pour la ou les connexions au sein d'une de ses sessions iSCSI. Un portail réseau dans un initiateur est identifié par son adresse IP. Un portail réseau dans une cible est identifié par son adresse IP et son accès TCP d'écoute.

Origine : Dans une négociation ou un échange, c'est la partie qui initie la négociation ou l'échange.

PDU (Unité de données de protocole) : L'initiateur et la cible divisent leurs communications en messages. Le terme de "unité de données de protocole iSCSI" (PDU iSCSI) est utilisé pour ces messages.

Groupes de portails : iSCSI prend en charge plusieurs connexions au sein de la même session ; certaines mises en œuvre vont avoir la capacité de combiner les connexions dans une session à travers plusieurs portails réseau. Un groupe de portails définit un ensemble de portails réseau au sein d'une entité réseau iSCSI qui prend en charge collectivement la capacité de coordination d'une session avec les connexions qui traversent ces portails. Tous les portails réseau au sein d'un groupe de portails n'ont pas besoin de participer à toutes les sessions connectées à travers ce groupe de portails. Un ou plusieurs groupes de portails peuvent fournir l'accès à un nœud iSCSI. Chaque portail réseau, tel qu'utilisé par un certain nœud iSCSI, appartient à exactement un groupe portail au sein de ce nœud.

Étiquette de groupe de portails : Cette quantité de 16 bits identifie un groupe de portails au sein d'un nœud iSCSI. Tous les portails réseau avec la même étiquette de groupe de portails dans le contexte d'un certain nœud iSCSI sont dans le même groupe de portails.

R2T de récupération : R2T généré par une cible à la détection de la perte d'une ou plusieurs PDU de données sortantes à travers un des moyens suivants : une erreur de résumé, une erreur de numéro de séquence, ou une fin de temporisation de réception de numéro de séquence. Un R2T de récupération porte le prochain R2TSN non utilisé, mais demande tout ou partie de la salve de données qu'un R2T antérieur (avec un R2TSN inférieur) avait déjà demandé.

Répondant : Dans une négociation ou échange, la partie qui répond à l'origine de la négociation ou échange.

SAS *(Serial Attached SCSI)* : La norme SCSI rattachée en série (SAS) contient une couche physique compatible avec l'ATA en série, et des protocoles pour transporter les commandes SCSI aux appareils SAS et les commandes ATA aux appareils SATA [SAS], [SPL].

Appareil SCSI : C'est le terme SAM2 pour une entité qui contient un ou plusieurs accès SCSI qui sont connectés à un sous-système de livraison de service et qui prennent en charge un protocole d'application SCSI. Par exemple, un appareil initiateur SCSI contient un ou plusieurs accès d'initiateur SCSI et zéro, un ou plusieurs clients d'application. Un appareil cible contient un ou plusieurs accès de cible SCSI et un ou plusieurs serveurs d'appareils et les unités logiques associées. Pour iSCSI, l'appareil SCSI est le composant au sein d'un nœud iSCSI qui fournit la fonction de SCSI. À ce titre, il ne peut y avoir au plus qu'un appareil SCSI au sein d'un certain nœud iSCSI. L'accès à l'appareil SCSI ne peut être réalisé que dans une session de fonctionnement normal iSCSI. Le nom d'appareil SCSI est défini comme étant le nom iSCSI du nœud.

Couche SCSI : Elle construit/reçoit les blocs de descripteur de commande SCSI (CDB, *Command Descriptor Block*) et les relaye/reçoit avec les paramètres d'exécution de commande restants [SAM2] de/vers la couche iSCSI.

Session : Le groupe de connexions TCP qui relie un initiateur à une cible forme une session (en gros équivalente à un lien I-T SCSI). Les connexions TCP peuvent être ajoutées et retirées d'une session. À travers toutes les connexions au sein d'une session, un initiateur voit une seule et même cible.

Accès SCSI : C'est le terme SAM2 pour une entité dans un appareil SCSI qui fournit la fonction SCSI à l'interface avec un sous-système de livraison de service. Pour iSCSI, la définition de l'accès d'initiateur SCSI et de l'accès de cible SCSI est différente.

Accès d'initiateur SCSI : Cela se transpose en point d'extrémité d'une session opérationnelle normale iSCSI. Une session opérationnelle normale iSCSI est négociée à travers le processus de connexion entre un nœud initiateur iSCSI et un nœud cible iSCSI. À l'achèvement réussi de ce processus, un accès d'initiateur SCSI est créé au sein de l'appareil initiateur SCSI. Le nom de l'accès d'initiateur SCSI et l'identifiant d'accès d'initiateur SCSI sont tous deux définis

comme étant le nom d'initiateur iSCSI ainsi que (a) une étiquette qui l'identifie comme nom/identifiant d'accès d'initiateur et (b) la portion ISID de l'identifiant de session.

Nom d'accès SCSI : C'est un nom consistant en codage UTF-8 [RFC3629] de caractères Unicode [UNICODE] et qui inclut le nom iSCSI + 'i' ou 't' + ISID ou l'étiquette de groupe portail.

SPDTL *(SCSI-Presented Data Transfer Length)* : La longueur du transfert de données présentées à SCSI est la longueur de données agrégée des données que la couche SCSI "présente" logiquement à la couche iSCSI pour un transfert de données entrantes ou sortantes dans le contexte d'une tâche SCSI. Pour une tâche bidirectionnelle, il y a deux valeurs de SPDTL -- une pour les données entrantes (Data-In) et une pour les données sortantes (Data-Out). Noter que la notion de "présentation" inclut les données immédiates selon le modèle de transfert de données dans [SAM2] et exclut les transferts de données qui se chevauchent, s'il en est, demandés par la couche SCSI.

Accès de cible SCSI : Ceci se transpose en groupe portail cible iSCSI.

Nom d'accès cible SCSI et identifiant d'accès cible SCSI : Ils sont tous deux définis comme étant le nom de cible iSCSI avec (a) une étiquette qui l'identifie comme un nom/identifiant d'accès de cible et (b) l'étiquette de groupe portail.

Identifiant de session (SSID) : C'est une session entre un initiateur iSCSI et une cible iSCSI qui est définie par un identifiant de session qui est un tuplet composé d'une partie initiateur (ISID) et une partie cible (étiquette de groupe portail cible). L'ISID est explicitement spécifié par l'initiateur à l'établissement de la session. L'étiquette de groupe portail cible est impliquée par l'initiateur par le choix du point d'extrémité TCP à l'établissement de la connexion. La clé TargetPortalGroupTag doit aussi être retournée par la cible comme confirmation durant l'établissement de la connexion.

T10 : C'est un comité technique de l'INCITS qui développe des normes et rapports techniques sur les interfaces I/O, en particulier la série des normes SCSI *(Small Computer System Interface)*. Voir < http://www.t10.org >.

T11 : C'est un comité technique de l'INCITS qui est charge du développement des normes dans les domaines des interfaces périphériques intelligentes (IPI, *Intelligent Peripheral Interface*), des interfaces parallèles hautes performances (HIPPI, *High-Performance Parallel Interface*), et du canal fibre (FC, *Fibre Channel*). Voir < http://www.t11.org >.

Étiquette de groupe portail cible : C'est un identifiant numérique (16 bits) pour un groupe portail cible iSCSI.

TTT *(Target Transfer Tag)* : L'étiquette de transfert de données (TTT) est un champ de protocole iSCSI utilisé dans certaines PDU iSCSI (par exemple, R2T, NOP-In) qui sont toujours envoyées d'abord de la cible à l'initiateur et ensuite citées comme référence dans les PDU renvoyées de l'initiateur à la cible et qui se rapportent à la même tâche ou échange. Donc, le TTT agit effectivement comme une bride opaque d'une tâche/échange existant pour aider la cible à associer les PDU entrantes venant de l'initiateur avec le contexte d'exécution approprié.

Tiers : Ce terme est utilisé dans le document comme un qualificatif des objets lien (I_T ou I_T_L) et des sessions iSCSI, pour indiquer que ces objets et sessions récoltent les effets collatéraux des actions qui ont lieu dans le contexte d'une session iSCSI séparée. Un exemple de session tierce est une session iSCSI qui découvre que son lien I_T_L à une LU a été réinitialisé à cause d'une opération de réinitialisation de LU orchestrée via un autre lien I_T.

Descripteur identifiant de session cible (TSIH) : Étiquette allouée à une cible pour une session avec un initiateur désigné spécifique. La cible la génère durant l'établissement de session. À part de la définir comme une chaîne binaire de 16 bits, son format et contenu interne ne sont pas définis par ce protocole sauf pour la valeur avec tous les bits à 0 qui est réservée et est utilisée par l'initiateur pour indiquer une nouvelle session. Elle est donnée à la cible durant un établissement de connexion supplémentaire pour la même session.

2.3 Résumé des changements

1) Consolidation des RFC 3720, 3980, 4850, et 5048, et insertion des corrections rédactionnelles nécessaires.
2) Spécification de iSCSIProtocolLevel = "1" au paragraphe 13.24 et ajout d'une référence normative à la [RFC7144].
3) Suppression des marqueurs et des clés qui s'y rapportent.
4) Suppression de l'authentification SPKM et des clés qui s'y rapportent.
5) Ajout d'un nouveau paragraphe 13.25 sur la réponse à des clés obsolètes.
6) Admission explicite tout au long du texte des mises en œuvre d'initiateur + cible.
7) Précision au paragraphe 4.2.7 que les mises en œuvre NE DEVRAIENT PAS s'appuyer sur la découverte fondée sur SLP.

8) Ajout de diagrammes en langage de modélisation unifiée (UML, *Unified Modeling Language*) et des conventions qui s'y rapportent à la Section 3.
9) Transformation de l'exigence de la mise en œuvre de FastAbort en "DEVRAIT" au paragraphe 4.2.3.4, à la place du "DOIT" précédent.
10) Au paragraphe 4.2.7.1, il est exigé que le nom de cible iSCSI soit le même que le nom d'initiateur iSCSI pour les appareils SCSI (composite) qui ont les deux rôles.
11) Changement du "NE DOIT PAS" en "devrait être évité" au paragraphe 4.2.7.2 concernant l'usage de caractères tels que des marques de ponctuation dans les noms iSCSI.
12) Mise à jour du paragraphe 9.3 pour exiger ce qui suit : DOIT mettre en œuvre IPsec, les RFC de la série 2400 (IPsec v2, IKEv1) ; et DEVRAIT mettre en œuvre IPsec, les RFC de la série 4300 (IPsec v3, IKEv2).
13) Précision au paragraphe 10.2 que ACA est un "DEVRAIT" seulement pour les cibles iSCSI.
14) Interdiction de l'usage du préfixe de nom X# pour les nouvelles clés publiques au paragraphe 6.2.
15) Interdiction de l'usage du préfixe de nom Y# pour les nouvelles extensions de résumé au paragraphe 13.1 et du préfixe de nom Z# pour les nouvelles extensions de méthode d'authentification au paragraphe 12.1.
16) Ajout d'un "DEVRAIT" au paragraphe 6.2 que les initiateurs et les cibles prennent en charge au moins six (6) échanges durant la négociation de texte.
17) Ajout de la précision que l'Appendice C est normatif.
18) Ajout d'une exigence normative sur la [RFC7146] et de quelques changements en découlant au paragraphe 9.3 pour aligner le texte de ce document avec celui de la [RFC7146].
19) Ajout d'un nouveau paragraphe 9.2.3 couvrant les considérations d'authentification Kerberos.
20) Ajout de texte au paragraphe 9.3.3 pour noter que OCSP est maintenant permis pour vérifier les certificats utilisés avec IPsec en plus de l'utilisation des CRL.
21) Ajout de texte au paragraphe 9.3.1 pour spécifier que les numéros de séquence étendus (ESN, *extended sequence number*) sont maintenant exigés pour ESPv2 (partie de IPsec v2).

2.4 Conventions

Dans les exemples, "I->" et "T->" montrent des PDU iSCSI envoyés respectivement par l'initiateur et la cible.
Les mots clés "DOIT", "NE DOIT PAS", "EXIGE", "DEVRA", "NE DEVRA PAS", "DEVRAIT", "NE DEVRAIT PAS", "RECOMMANDE", "PEUT", et "FACULTATIF" dans ce document sont à interpréter comme décrit dans la [RFC2119].

3. Conventions UML

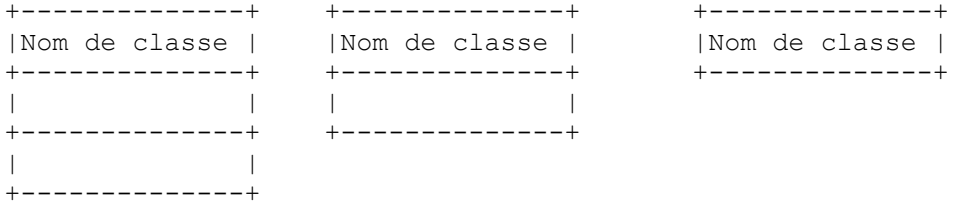
3.1 Généralités sur les conventions UML

Le modèle d'architecture SCSI (SAM, *SCSI Architecture Model*) utilise des diagrammes de classe et des diagrammes d'objets avec une notation qui se fonde sur le langage de modélisation unifié (UML, *Unified Modeling Language*) [UML]. Donc, le présent document utilise aussi UML pour modéliser les relations des objets SCSI et iSCSI.
Il n'appartient pas au présent document de rédiger un traité sur la notation graphique utilisée dans UML. Cependant, étant donné l'utilisation des dessins ASCII pour les diagrammes de classe statique UML, une description des conventions de notation utilisées dans ce document est incluse dans la suite de cette section.

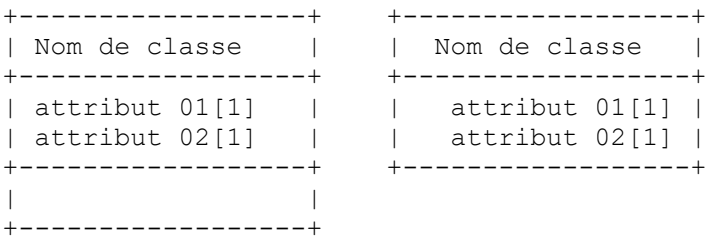
3.2 Notion de multiplicité

Non spécifiée. Le nombre d'instances d'un attribut n'est pas spécifié.
1 : une instance de la classe ou de l'attribut existe.
0..* : zéro, une ou plusieurs instances de la classe ou de l'attribut existent.
1..* : une ou plusieurs instances de la classe ou de l'attribut existent.
0..1 : zéro ou une instance de la classe ou de l'attribut existe.
n..m : n à m instances de la classe ou de l'attribut existent (par exemple, 2..8).
x, n..m : plusieurs instances disjointes de la classe ou de l'attribut existent (par exemple, 2, 8..15).

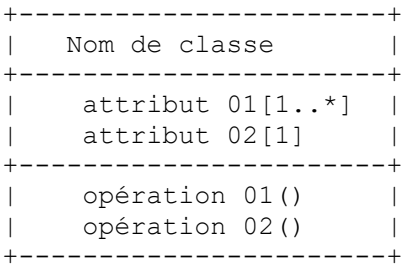
3.3 Conventions de diagramme de classe



Les trois précédents diagrammes sont des exemples d'une classe sans attribut et sans opération.

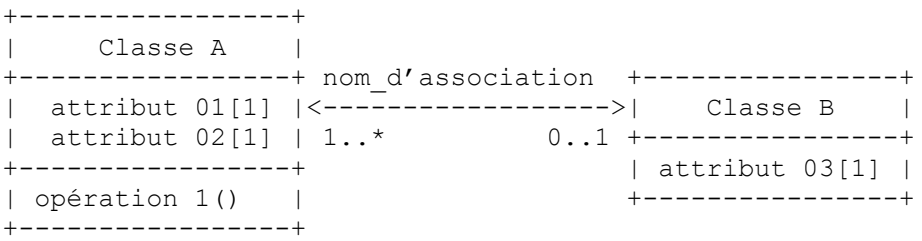


Les deux diagrammes précédents sont des exemples d'une classe avec des attributs et pas d'opération.

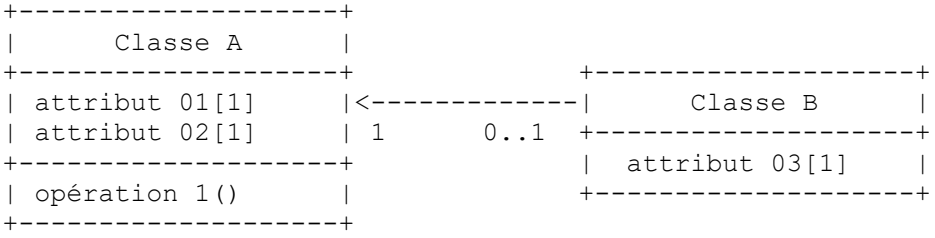


Le diagramme précédent est un exemple d'une classe avec des attributs qui ont une multiplicité spécifiée et des opérations.

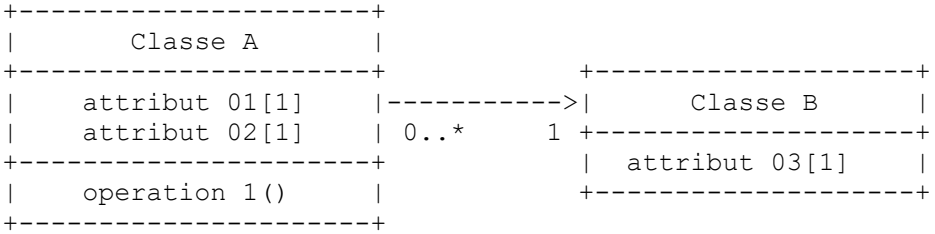
3.4 Notation de diagramme de classe pour des associations



Le diagramme précédent est un exemple où la classe A connaît la classe B (c'est-à-dire, se lit comme "Classe A nom_d'association Classe B") et la classe B connaît la classe A (c'est-à-dire, se lit comme "Classe B nom_d'association Classe A"). L'utilisation de nom_d'association est facultative. la notation de la multiplicité (1..* et 0..1) indique le nombre d'instances de l'objet.

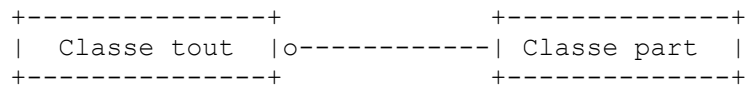


Le diagramme précédent est un exemple où la classe A connaît la classe B (c'est-à-dire, se lit comme "Classe B connaît Classe A") mais la classe A ne connaît pas la classe B.

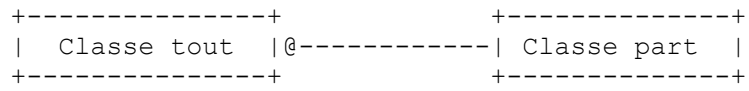


Le diagramme précédent est un exemple où la classe A connaît la classe B (c'est-à-dire, se lit comme "Classe A connaît Classe B") mais Classe B ne connaît pas Classe A.

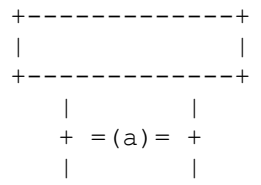
3.5 Notation de diagramme de classe pour l'agrégation



Le diagramme précédent est un exemple où Classe tout est un agrégat qui contient Classe part et où Classe part peut continuer d'exister même si Classe tout est supprimé (c'est-à-dire, lire comme "le tout contient la partie").

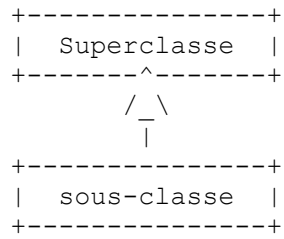


Le diagramme précédent est un exemple où Classe tout est un agrégat qui contient Classe part où Classe part n'appartient qu'à une Classe tout, et où la Classe part ne continue pas d'exister si la Classe tout est supprimée (c'est-à-dire, se lit comme "le tout contient la partie").



Le diagramme précédent est un exemple où il y a une contrainte entre les associations, où la note de bas de page (a) décrit la contrainte.

3.6 Notation de diagramme de classe pour les généralisations



Le diagramme précédent est un exemple où la sous-classe est une sorte de superclasse. Une sous-classe partage tous les attributs et toutes les opérations de la superclasse (c'est-à-dire que la sous-classe hérite de la superclasse).

4. Vue d'ensemble

4.1 Concepts de SCSI

Le modèle 2 d'architecture de SCSI [SAM2] décrit en détail l'architecture de la famille SCSI de protocoles d'entrée/sortie. La présente section donne brièvement les fondements de l'architecture SCSI et est destinée à familiariser le lecteur avec sa terminologie.

Au plus haut niveau, SCSI est une famille d'interfaces pour demander des services à partir d'appareils d'entrée/sortie, incluant des pilotes matériels, des lecteurs de bandes, des pilotes de CD et DVD, des imprimantes, et des scanneurs. Dans la terminologie SCSI, un appareil d'entrée/sortie individuel est appelé une "unité logique" (LU, *logical unit*).

SCSI est une architecture client-serveur. Les clients d'une interface SCSI sont appelés des "initiateurs". Les initiateurs produisent des "commandes" SCSI pour demander des services à des composants, les unités logiques, d'un serveur appelé une "cible". L'appareil serveur sur l'unité logique accepte les commandes SCSI et les traite.

Un "transport SCSI" transpose le protocole SCSI client-serveur sur une interconnexion spécifique. Les initiateurs sont un point d'extrémité d'un transport SCSI. La "cible" est l'autre point d'extrémité. Une cible peut contenir plusieurs unités

logiques (LU). Chaque unité logique a une adresse au sein d'une cible, appelée numéro d'unité logique (LUN, *Logical Unit Number*).

Une tâche SCSI est une commande SCSI ou éventuellement un ensemble relié de commandes SCSI. Certaines LU prennent en charge plusieurs tâches en cours (en file d'attente) mais la file d'attente des tâches est gérée par l'unité logique. La cible utilise une "étiquette de tâche" fournie par l'initiateur pour distinguer les tâches. Une seule commande dans une tâche peut être en suspens à un moment donné.

Chaque commande SCSI résulte en une phase de données facultative et une phase de réponse obligatoire. Dans la phase de données, les informations peuvent voyager de l'initiateur à la cible (par exemple, WRITE), de la cible à l'initiateur (par exemple, READ) ou dans les deux directions. Dans la phase de réponse, la cible retourne l'état final de l'opération, y compris toutes erreurs.

Les blocs de descripteur de commande (CDB, *Command Descriptor Block*) sont les structures de données utilisées pour contenir les paramètres de commande qu'un initiateur envoie à une cible. Le contenu et la structure du CDB sont définis par [SAM2] et les normes SCSI spécifques de type d'appareil.

4.2 Concepts iSCSI et vue fonctionnelle d'ensemble

Le protocole iSCSI est une transposition du modèle d'invocation de procédure distante SCSI (voir [SAM2]) sur le protocole TCP. Les commandes SCSI sont portées par les demandes iSCSI et les réponses SCSI et les états sont portés par les réponses iSCSI. iSCSI utilise aussi le mécanisme demande-réponse comme mécanisme de protocole iSCSI.

Dans la suite de ce document, les termes "initiateur" et "cible" se réfèrent au "nœud initiateur iSCSI" et au "nœud cible iSCSI", respectivement (voir le paragraphe 3.4.1 "Modèle d'architecture iSCSI") sauf qualification contraire.

Comme son titre le suggère, la Section 4 présente une vue d'ensemble des concepts iSCSI, et les sections suivantes du reste de la spécification contiennent les exigences normatives – couvrant dans de nombreux cas les mêmes concepts exposés dans la Section 4. Le texte de ces exigences normatives a la préséance sur le texte de la présentation générale de la Section 4 en cas de désaccord entre les deux.

À l'instar de protocoles similaires, l'initiateur et la cible divisent leurs communications en messages. Le présent document utilise le terme de "unité de données de protocole iSCSI" (iSCSI PDU) pour ces messages.

Pour des raisons de performances, iSCSI permet un "collapsus de phase". Une commande et ses données associées peuvent être transportées ensemble de l'initiateur à la cible, et les données et réponses peuvent être transportées ensemble à partir des cibles.

La direction de transfert iSCSI est définie par rapport à l'initiateur. Les transferts sortants ou externes sont des transferts d'un initiateur à une cible, tandis que les transferts entrants ou internes sont d'une cible à un initiateur.

Une tâche iSCSI est une demande iSCSI pour laquelle une réponse est attendue.

Dans ce document "demande iSCSI", "commande iSCSI", demande, ou commande (non qualifiée) ont la même signification. Aussi, sauf spécification contraire, état, réponse, ou réponse numérotée ont la même signification.

4.2.1 Couches et sessions

Le modèle de mise en couches conceptuel suivant est utilisé pour spécifier les actions d'initiateur et de cible et la façon dont elles se rapportent aux unités de données de protocole transmises et reçues :

- la couche SCSI construit/reçoit les blocs descripteurs de commande (CDB, *Command Descriptor Block*) SCSI et les passe/reçoit avec le reste des paramètres de commande d'exécution [SAM2] à/de
- la couche iSCSI qui construit/reçoit les PDU iSCSI et les relaye/reçoit vers/de une ou plusieurs connexions TCP ; le groupe de connexions forme une "session" initiateur-cible.

La communication entre l'initiateur et la cible survient sur une ou plusieurs connexions TCP. Les connexions TCP portent des messages de contrôle, des commandes SCSI, des paramètres, et des données au sein des unités de données de protocole iSCSI (iSCSI PDU). Le groupe de connexions TCP qui relie un initiateur à une cible forme une session (en gros équivalente à un lien I_T SCSI, voir au paragraphe 3.4.2 "Modèle d'architecture SCSI). Une session est définie par un identifiant de session qui se compose d'une partie initiateur et d'une partie cible. Les connexions TCP peuvent être ajoutées et retirées d'une session. Chaque connexion au sein d'une session est identifiée par un identifiant de connexion (CID).

À travers toutes les connexions au sein d'une session, un initiateur voit une "image". Les éléments qui identifient une cible, comme LUN, sont les mêmes. Une cible voit aussi une "image d'initiateur" à travers toutes les connexions au sein d'une session. Les éléments qui identifient l'initiateur, comme l'étiquette de tâche d'initiateur sont globaux au sein de la session sans considération de la connexion sur laquelle ils sont envoyés ou reçus.

Les cibles et initiateurs iSCSI DOIVENT prendre en charge au moins une connexion TCP et PEUVENT prendre en charge plusieurs connexions dans une session. Pour les besoins de la récupération d'erreur, les cibles et initiateurs qui prennent en charge une seule connexion active dans une session DEVRAIENT prendre en charge deux connexions durant la récupération.

4.2.2 Ordre et numérotation iSCSI

iSCSI utilise des schémas de numérotation de commandes et d'états et un schéma de séquençage des données.

Le numérotage des commandes est au niveau session et est utilisé pour une livraison ordonnée des commandes sur plusieurs connexions. Il peut aussi être utilisé comme mécanisme de contrôle des flux de commandes sur une session.

Le numérotage d'état est fait par connexion et est utilisé pour permettre la détection et la récupération des états manquants en présence d'erreurs de communication transitoires ou permanentes.

Le séquençage des données est par commande ou partie d'une commande (séquence déclenchée par R2T) et est utilisé pour détecter des données manquantes et/ou PDU R2T dues à des erreurs de résumé d'en-tête.

Normalement, les champs dans les PDU iSCSI communiquent les numéros de séquence entre l'initiateur et la cible. Durant les périodes où le trafic sur une connexion est unidirectionnel, les PDU iSCSI NOP-Out/In peuvent être utilisées pour synchroniser les compteurs d'ordre de commandes et d'états de la cible et de l'initiateur.

L'abstraction de session iSCSI est équivalente au lien I_T SCSI, et la session iSCSI fournit une livraison de commande ordonnée de l'initiateur SCSI à la cible SCSI. Pour les considérations détaillées de conception qui ont conduit au modèle de session iSCSI comme il est défini ici et comment il met en rapport les caractéristiques d'arrangement des commandes SCSI définies dans les spécifications SCSI avec les concepts iSCSI, voir la [RFC3783].

4.2.2.1 Numérotation et accusé de réception des commandes

iSCSI effectue la livraison ordonnée des commandes au sein d'une session. Toutes les commandes (PDU d'initiateur à cible) en transit de l'initiateur à la cible sont numérotées.

iSCSI considère qu'une tâche est à instancier sur la cible en réponse à chaque demande produite par l'initiateur. Un ensemble d'opérations de gestion de tâches incluant l'interruption et la réallocation (voir le paragraphe 10.5 "Demande de fonction de gestion de tâche") peut être effectué sur toute tâche iSCSI.

Certaines tâches iSCSI sont des tâches SCSI, et de nombreuses activités de SCSI se rapportent à une tâche de SCSI ([SAM2]). Dans tous les cas, la tâche est identifiée par l'étiquette de tâche d'initiateur (ITT, *Initiator Task Tag*) pour la vie de la tâche.

Le numéro de commande est porté par la PDU iSCSI comme numéro de séquence de commande (CmdSN, *Command sequence number*). La numérotation se fait au sein de la session. Les PDU iSCSI sortantes portent ce numéro. L'initiateur iSCSI alloue les CmdSN avec un compteur de 32 bits non signés (modulo 2**32). Les comparaisons et l'arithmétique sur CmdSN utilisent l'arithmétique des numéros de série définie dans la [RFC1982] où SERIAL_BITS = 32.

Les commandes destinées à une livraison immédiate sont marquées avec un fanion de livraison immédiate ; elles DOIVENT aussi porter le CmdSN en cours. Le CmdSN n'avance pas après l'envoi d'une commande marquée pour livraison immédiate.

Le numérotage de commande commence avec la première demande de connexion sur la première connexion d'une session (l'identifiant de tête sur la connexion de tête) et les numéros de commande sont incrémentés de 1 pour chaque commande non immédiate produite ensuite.

Si la livraison immédiate est utilisée avec des commandes de gestion de tâches, ces commandes peuvent atteindre la cible avant des tâches sur lesquelles elles sont supposées agir. Cependant leur CmdSN sert de marqueur de leur position dans le flux de commandes. L'initiateur et la cible doivent s'assurer que les commandes de gestion de tâche agissent comme spécifié par [SAM2]. Par exemple, les commandes et les réponses apparaissent toutes deux si elles sont livrées dans l'ordre.

Chaque fois que le CmdSN pour une PDU sortante n'est pas spécifié par une règle explicite, le CmdSN va porter la valeur courante de la variable CmdSN locale (voir plus loin dans ce paragraphe).

Les moyens par lesquels une mise en œuvre décide de marquer une PDU pour livraison immédiate ou par lesquels iSCSI décide de lui-même de marquer une PDU pour livraison immédiate sortent du domaine d'application du présent document.

Le nombre de commandes utilisées pour livraison immédiate n'est pas limité et leur livraison pour exécution n'est pas reconnue par le schéma de numérotation. Les commandes immédiates PEUVENT être rejetées par la couche cible iSCSI à cause d'un manque de ressources. Une cible iSCSI DOIT être capable de traiter à tout moment par connexion au moins une commande de gestion de tâche immédiate et une commande iSCSI non de gestion de tâche immédiate.

Dans le présent document, livraison pour exécution signifie livraison au moteur d'exécution SCSI ou à un moteur d'exécution iSCSI spécifique du protocole (par exemple, pour les demandes de texte avec des clés d'extension publiques ou privées qui impliquent un composant d'exécution). À l'exception des commandes marquées pour livraison immédiate, la couche cible iSCSI DOIT livrer les commandes pour exécution dans l'ordre spécifié par le CmdSN. Les commandes marquées pour livraison immédiate peuvent être délivrées par la couche cible iSCSI pour exécution aussitôt que détectées. iSCSI peut éviter de livrer certaines commandes à la couche cible SCSI si c'est exigé par une action SCSI ou iSCSI antérieure (par exemple, une demande de gestion de tâche CLEAR TASK SET *(nettoyer l'ensemble de tâches)* reçue avant toutes les commandes sur lesquelles elle est supposée agir).

Sur toute connexion, l'initiateur iSCSI DOIT envoyer les commandes en ordre croissant de numéro de séquence de commande, sauf pour les commandes qui sont retransmises du fait d'une récupération d'erreur de résumé et d'une récupération de connexion.

Pour le mécanisme de numérotage, l'initiateur et la cible entretiennent les trois variables suivantes pour chaque session :

- CmdSN – Numéro de séquence de la commande en cours, avancé de 1 à chaque commande envoyée excepté pour les commandes marquées pour livraison immédiate. CmdSN contient toujours le numéro à allouer à la PDU de prochaine commande.
- ExpCmdSN – Numéro de séquence de la prochaine commande attendue par la cible. La cible accuse réception de toutes les commandes jusqu'à, non compris, ce numéro. L'initiateur traite toutes les commandes avec le numéro de séquence de commande inférieur à ExpCmdSN comme acquittées. La couche iSCSI cible règle ExpCmdSN au plus grand numéro de séquence de commande non immédiat qu'il peut livrer pour exécution plus 1 (pas de trou dans la séquence CmdSN).
- MaxCmdSN – Numéro de séquence de commande maximum à envoyer. La capacité de mise en file d'attente de la couche iSCSI receveuse est MaxCmdSN - ExpCmdSN + 1.

Les ExpCmdSN et MaxCmdSN de l'initiateur sont déduits des champs de PDU de cible à initiateur. Les comparaisons et l'arithmétique sur ExpCmdSN et MaxCmdSN DOIVENT utiliser l'arithmétique des numéros de série définie dans la [RFC1982] où SERIAL_BITS = 32.

La cible NE DOIT PAS transmettre un MaxCmdSN inférieur à ExpCmdSN-1. Pour les commandes non immédiates, le champ CmdSN peut prendre toute valeur de ExpCmdSN à MaxCmdSN inclus. La cible DOIT ignorer en silence toute commande non immédiate en dehors de cette gamme ou des dupliqués non immédiats dans la gamme. Le CmdSN porté par des commandes immédiates peut se tenir en dehors de la gamme de ExpCmdSN à MaxCmdSN. Par exemple, si l'initiateur a envoyé précédemment une commande non immédiate portant le CmdSN égal à MaxCmdSN, la fenêtre cible est close. Pour les commandes de gestion de tâche de groupe produites comme commandes immédiates, le CmdSN indique la portée de l'action de groupe (par exemple, sur ABORT TASK SET, il indique quelles commandes sont interrompues).

Les champs MaxCmdSN et ExpCmdSN sont traités comme suit par l'initiateur :
- Si le MaxCmdSN de la PDU est inférieur à ExpCmdSN-1 de la PDU (au sens de l'arithmétique des numéros de série), ils sont tous deux ignorés.
- Si le MaxCmdSN de la PDU est supérieur au MaxCmdSN local (au sens de l'arithmétique des numéros de série) il met à jour le MaxCmdSN local ; autrement, il est ignoré.
- Si le ExpCmdSN de la PDU est supérieur au ExpCmdSN local (au sens de l'arithmétique des numéros de série) il met à jour le ExpCmdSN local ; autrement, il est ignoré.

Cette séquence est exigée parce que les mises à jour peuvent arriver en désordre (par exemple, les mises à jour sont envoyées sur des connexions TCP différentes).

Les initiateurs et cibles iSCSI DOIVENT prendre en charge le schéma de numérotation des commandes.

Une demande iSCSI numérotée ne va jamais changer le CmdSN qui lui est alloué, sans considération du nombre de fois et des circonstances dans lesquelles il est réémis (voir au paragraphe 6.2.1 "Usage du réessai). À la cible, CmdSN n'est pertinent que lorsque la commande n'a pas créé d'état en rapport avec son exécution (état d'exécution) ; ensuite, le CmdSN n'est plus pertinent. La vérification de l'état d'exécution (représenté par l'identification de l'étiquette Tâche d'initiateur) DOIT précéder toute autre action à la cible. Si aucun état d'exécution n'est trouvé, il est suivi par le rangement et la livraison. Si un état d'exécution est trouvé, il est suivi par la livraison.

Si un initiateur produit une commande réessayer pour une commande avec le CmdSN R sur une connexion lorsque la valeur du CmdSN de la session est Q, il NE DOIT PAS avancer le CmdSN au delà de R + 2**31 – 1, sauf si la connexion n'est plus opérationnelle (c'est-à-dire, si il a retourné l'état FREE, voir le paragraphe 7.1.3 "Diagramme d'état de connexion standard pour un initiateur") si la connexion a été réinstanciée (voir au paragraphe 5.3.4 "Réinstanciation de connexion") ou si une commande non immédiate avec CmdSN égal ou supérieur à Q a été produite suite à la commande réessayer sur la même connexion et si la réception de cette commande est reconnue par la cible (voir au paragraphe 9.4 "Réessai de commande et purge des vieilles instances de commande").

Une cible NE DOIT PAS produire une réponse de commande ou une PDU Data-In avec état avant d'accuser réception de la commande. Cependant, l'accusé de réception peut être inclus dans la réponse ou la PDU Data-In.

4.2.2.2 Numérotation et accusé de réception de réponse/état

Les réponses en transit de la cible à l'initiateur sont numérotées. Le numéro de séquence d'état (StatSN, *Status Sequence Number*) est utilisé à cette fin. Numéro de séquence d'état est un compteur entretenu par connexion. ExpStatSN est utilisé par l'initiateur pour accuser réception des états. L'espace de numéro de séquence d'état est d'entiers de 32 bits non signés et les opérations arithmétiques sont l'arithmétique régulière modulo (2**32).

Le numérotage d'état commence par la réponse Établissement à la première demande d'établissement de la connexion. La réponse d'établissement comporte une valeur initiale pour le numérotage d'état (toute valeur initiale est valide).

Pour permettre la récupération de commande, la cible PEUT conserver assez d'informations d'état pour la récupération des données et de l'état après une défaillance de connexion. Une cible qui fait cela peut en toute sécurité éliminer toutes les informations d'état conservées pour la récupération d'une commande après que la livraison de l'état pour la commande (StatSN numéroté) a été acquittée par un ExpStatSN.

Une grosse différence absolue entre Numéro de séquence d'état et Numéro attendu de séquence d'état peut indiquer une connexion défaillante. Les initiateurs DOIVENT entreprendre des actions de récupération si la différence est supérieure à une constante, définie par la mise en œuvre, qui NE DOIT PAS excéder 2**31-1.

Les initiateurs et les cibles DOIVENT prendre en charge le schéma de numérotation de réponse.

4.2.2.3 Réponse Ordering

4.2.2.3.1 Besoin de la réponse Ordering

Chaque fois qu'une session iSCSI se compose de multiples connexions, les PDU de réponse (réponses de tâches ou réponses de TMF) générées dans la couche SCSI de la cible sont distribuées aux diverses connexions par la couche iSCSI de la cible conformément aux règles d'allégeance de connexion iSCSI. Ce processus ne peut généralement pas préserver l'ordre des réponses au moment où elles sont livrées à la couche SCSI de l'initiateur.

Comme de toutes façons on ne s'attend pas à ce que les PDU de réponse SCSI soient dans l'ordre, cette approche fonctionne bien dans le cas général. Cependant, pour régler des cas particuliers où un certain ordre est souhaité par la couche SCSI, on introduit la notion d'une "réponse close" *(response Fence)* : une réponse close est logiquement l'attribut/propriété d'un message de réponse SCSI passé à une couche iSCSI cible qui indique que des considérations d'ordre particulières de niveau SCSI sont associées à ce message de réponse particulier. Chaque fois qu'une réponse close est établie ou exigée sur un message de réponse SCSI, on définit la sémantique au paragraphe 4.2.2.3.2 par rapport au traitement de la couche iSCSI de la cible de tels messages de réponse SCSI.

4.2.2.3.2 Description du modèle de réponse Ordering

La couche de protocole SCSI cible passe les messages de réponse SCSI à la couche iSCSI de la cible en invoquant le service de données de protocole "Commande d'envoi achevée" ([SAM2], paragraphe 5.4.2) et le service "Fonction de gestion de tâche exécutée" ([SAM2], paragraphe 6.9). À réception du message de réponse SCSI, la couche iSCSI affiche le comportement de réponse close pour certains messages de réponse SCSI (le paragraphe 4.2.2.3.4 décrit les instances spécifiques où cette sémantique doit être réalisée).

Chaque fois que le comportement de réponse close est requis pour un message de réponse SCSI, la couche iSCSI de la cible DOIT s'assurer que les conditions suivantes sont satisfaites en livrant le message de réponse à la couche iSCSI de l'initiateur :
- Une réponse avec une réponse close DOIT être livrée chronologiquement après toutes les réponses "précédentes" sur le lien I_T_L, si les réponses précédentes sont bien livrées, à la couche iSCSI de l'initiateur.
- Une réponse avec une réponse close DOIT être livrée chronologiquement avant toutes les réponses "suivantes" sur le lien I_T_L.

Les notions de "précédente" et de "suivante" se réfèrent à l'ordre de passage d'un message de réponse de la couche de protocole SCSI de la cible à la couche iSCSI de la cible.

4.2.2.3.3 Sémantique iSCSI avec le modèle Interface

Chaque fois que la clé TaskReporting (paragraphe 13.23) est négociée à ResponseFence *(réponse close)* ou FastAbort pour une session iSCSI et que le comportement réponse close est exigé pour un message de réponse SCSI, la couche iSCSI de la cible DOIT effectuer les actions décrites dans le présent paragraphe pour cette session.

a) Si c'est une session à une seule connexion, aucun traitement particulier n'est nécessaire. Le processus standard SCSI de construction et de répartition de PDU de réponse se produit.

b) Si c'est une session multi connexions, la couche iSCSI de la cible prend note du dernier Numéro de séquence d'état envoyé et non acquitté sur chacune des connexions dans la session iSCSI, et attend un accusé de réception (des PDU NOP-In PEUVENT être utilisées pour solliciter des accusés de réception en tant que de besoin afin d'accélérer ce processus) de chacun de ces Numéro de séquence d'état pour supprimer la clôture. La PDU de réponse SCSI exigeant le comportement de réponse close NE DOIT PAS être envoyée à l'initiateur avant la réception des accusés de réception pour chacun des numéros de séquence d'état non acquittés.

c) La couche iSCSI cible doit attendre un accusé de réception de la PDU de réponse SCSI qui portait la réponse SCSI exigeant le comportement de réponse close. La clôture DOIT être considérée comme supprimée après la réception de l'accusé de réception.

d) Tous les autres traitements d'état pour la LU sont repris après la suppression de la clôture. Si une nouvelle réponse pour le lien I_T_L est reçue de la couche SCSI avant la suppression de la clôture, cette PDU de réponse DOIT être conservée et mise en file d'attente à la couche iSCSI jusqu'à la suppression de la clôture.

4.2.2.3.4 Liste actuelle de cas d'utilisation de réponse close

Ce paragraphe fait la liste des situations dans lesquelles le comportement de réponse close est EXIGÉ dans les mises en œuvre de cible iSCSI. Noter que cette liste est une énumération exhaustive de ce qui est actuellement identifié – on peut s'attendre à ce que la spécification du protocole SCSI évolue, les spécifications énuméreront l'exigence de réponse close au cas par cas.

Chaque fois que la clé TaskReporting (paragraphe 13.23) est négociée à ResponseFence ou FastAbort pour une session iSCSI, la couche iSCSI de la cible DOIT supposer que la réponse close est exigée pour les messages d'achèvement SCSI suivants :
a) Le premier message d'achèvement portant l'UA après l'interruption de multi-tâche sur les sessions productrices *(issuing)* et de tiers *(third-party)*. Voir au paragraphe 4.2.3.2 la discussion sur la fonction de tâche de gestion (TMF).
b) La réponse de TMF portant la réponse de TMF multi-tâche ou la session productrice.
c) Le message d'achèvement indiquant l'établissement d'ACA sur la session productrice.
d) Le premier message d'achèvement portant l'état ACA ACTIF après l'établissement d'ACA sur les sessions productrices et de tiers.
e) La réponse de TMF portant la réponse d'ACA CLEAR sur la session productrice.
f) La réponse à une commande PERSISTENT RESERVE OUT/PREEMPT AND ABORT.

Notes :
- Du fait de l'absence d'exigences de clôture relative à l'ACA dans la [RFC3720], les mises en œuvre d'initiateur NE DEVRAIENT PAS utiliser d'ACA sur les sessions multi connexion iSCSI avec ces cibles se conformant seulement à la [RFC3720]. Cela peut être déterminé via la négociation de clé TaskReporting (paragraphe 13.23) – lorsque la négociation résulte en "RFC3720" ou en "NonCompris".
- Les initiateurs qui veulent employer l'ACA sur des sessions multi connexion iSCSI DEVRAIENT d'abord s'assurer du comportement de clôture de réponse via la négociation de la valeur de "ResponseFence" ou de "FastAbort" pour la clé de TaskReporting (paragraphe 13.23).

4.2.2.4 Séquençage des données

Les données et les PDU R2T transférées au titre de l'exécution d'une commande DOIVENT être séquencées. Le champ DataSN *(numéro de séquence de données)* est utilisé pour le séquençage des données. Pour les PDU de données d'entrée (en lecture) DataSN commence à 0 pour la première PDU de données d'une commande d'entrée et avance de 1 pour chaque PDU de données suivante. Pour les PDU de données de sortie, DataSN commence par 0 pour la première PDU de données d'une séquence (la séquence initiale non sollicitée ou toute séquence de PDU de données produite pour satisfaire un R2T) et avance de 1 pour chaque PDU de données suivante. Les R2T sont aussi séquencés par commande. Par exemple, le premier R2T a un R2TSN de 0 et avance de 1 pour chaque R2T suivant. Pour les commandes bidirectionnelles, la cible utilise le DataSN/R2TSN pour séquencer les PDU Data-In et R2T dans une séquence continue (indifférenciée). À la différence des commandes et des états, les PDU de données et de R2T ne sont pas acquittées par un champ dans les PDU sortantes régulières. Les PDU Data-In peuvent être acquittées à la demande par une forme spéciale de la PDU SNACK. Les PDU de données et de R2T sont implicitement acquittées par l'état pour la commande. Le champ DataSN/R2TSN permet à l'initiateur de détecter les PDU de données ou de R2T manquantes.

Pour toute commande en lecture ou bidirectionnelle, une cible DOIT produire moins de 2**32 de PDU combinées de R2T et de Data-In. Toute séquence de données de sortie DOIT contenir moins de 2**32 PDU Data-Out.

4.2.3 Gestion des tâches iSCSI

4.2.3.1 Vue d'ensemble de la gestion des tâches

Les caractéristiques de gestion de tâche iSCSI permettent à un initiateur de contrôler les tâches iSCSI actives sur une session iSCSI opérationnelle qu'il a avec une cible iSCSI. Le paragraphe 11.5 définit la fonction de gestion des types de tâche que la présente spécification définit -- ABORT TASK, ABORT TASK SET, CLEAR ACA, CLEAR TASK SET, LOGICAL UNIT RESET, TARGET WARM RESET, TARGET COLD RESET, et TASK REASSIGN.

Dans ces types de fonctions, ABORT TASK et TASK REASSIGN gèrent une seule tâche active, tandis que ABORT TASK SET, CLEAR TASK SET, LOGICAL UNIT RESET, TARGET WARM RESET, et TARGET COLD RESET peuvent chacune affecter potentiellement plusieurs tâches actives.

4.2.3.2 Notion de tâche affectée

On définit ici la notion de "tâches affectées" dans les scénarios d'interruption multi tâches. Les définitions de portée de ce paragraphe s'appliquent aussi bien au comportement de la sémantique d'interruption de multi tâche standard (paragraphe 4.2.3.3) qu'à celui de la sémantique d'interruption de multi tâche FastAbort (paragraphe 4.2.3.4).

ABORT TASK SET *(interrompre l'ensemble de tâches)* : toutes les tâches en cours pour le lien I_T_L identifié par le champ LUN dans la PDU de demande de TMF ABORT TASK SET.

CLEAR TASK SET *(nettoyer l'ensemble de tâches)* : toutes les tâches en cours dans l'ensemble de tâches pour la LU identifiée par le champ LUN dans la PDU de la demande de TMF CLEAR TASK SET. Voir dans [SPC3] la définition d'un "ensemble de tâches".

LOGICAL UNIT RESET *(réinitialiser l'unité logique)* : toutes les tâches en cours provenant de tous les initiateurs pour la LU identifiée par le champ LUN dans la PDU de la demande LOGICAL UNIT RESET.

TARGET WARM/COLD RESET *(réinitialiser la cible à chaud/froid)* : toutes les tâches en cours provenant de tous les initiateurs à travers toutes les LU auxquels la session productrice de TMF a accès sur l'appareil cible SCSI qui héberge la session iSCSI.

Usage : une "PDU de demande de TMF ABORT TASK SET" dans le texte précédant est une PDU de demande de TMF iSCSI avec le champ "Fonction" réglé à "ABORT TASK SET" comme défini au paragraphe 11.5. Un usage similaire est employé pour les autres descriptions de portée.

4.2.3.3 Sémantique standard d'interruption de multi tâche

Toutes les mises en œuvre iSCSI DOIVENT prendre en charge le comportement de protocole défini dans ce paragraphe comme comportement par défaut. L'exécution des demandes de TMF ABORT TASK SET, CLEAR TASK SET, LOGICAL UNIT RESET, TARGET WARM RESET, et TARGET COLD RESET consiste en la séquence d'actions suivante dans l'ordre spécifié sur la partie spécifiée.

La couche iSCSI initiatrice :

a) DOIT continuer de répondre à chaque TTT reçue pour les tâches affectées.

b) DEVRAIT traiter toutes les réponses reçues pour les tâches affectées de façon normale. C'est acceptable parce que il est garanti que les réponses ont été envoyées avant la réponse de TMF.
c) DEVRAIT recevoir la réponse de TMF concluant toutes les tâches dans l'ensemble des tâches affectées, sauf si l'initiateur a fait quelque chose (par exemple, réinitialisation de LU, abandon de connexion) qui pourrait empêcher la réponse de TMF d'être envoyée ou reçue. L'initiateur DOIT donc conclure toutes les tâches affectées au titre de cette étape dans tous les cas et DOIT éliminer toute réponse de TMF reçue après la conclusion des tâches affectées.

La couche iSCSI cible :
a) DOIT attendre les réponses sur les étiquettes de transfert de cible (TTT) actuellement valides des tâches affectées provenant de l'initiateur qui les produit ; PEUT attendre des réponses sur les TTT actuellement valides des tâches affectées provenant d'initiateurs tiers.
b) DOIT attendre (concurremment avec l'attente de l'étape a) que toutes les commandes des tâches affectées soient reçues sur la base de l'ordre du CmdSN, NE DEVRAIT PAS attendre de nouvelles commandes sur les sessions affectées de tiers – seules les tâches instanciées doivent être considérées afin de déterminer les tâches affectées. Cependant, dans le cas de demandes de portée de cible (c'est-à-dire, TARGET WARM RESET et TARGET COLD RESET) toutes les commandes qui ne sont pas encore reçues sur la session productrice dans le flux de commandes peuvent être considérées comme ayant été reçues sans période d'attente de commande -- c'est-à-dire que l'espace entier de CmdSN jusqu'au CmdSN de la fonction de gestion des tâches peut être "bouché".
c) DOIT propager la demande de TMF à, et recevoir la réponse de, la couche cible SCSI.
d) DOIT fournir le comportement de réponse close pour la réponse de TMF sur la session productrice comme spécifié au paragraphe 4.2.2.3.2.
e) DOIT fournir le comportement de réponse close sur la première réponse post TMF sur les sessions de tiers, comme spécifié au paragraphe 4.2.2.3.3. Si certaines tâches ont pour origine des liens I_T_L non iSCSI, les moyens par lesquels la cible s'assure que toutes les tâches affectées ont retourné leur état à l'initiateur sont définis par le ou les protocoles de transport non iSCSI spécifiques.

Techniquement, le service de TMF est achevé à l'étape d). Les transferts de données correspondants aux tâches terminées peuvent, cependant, être encore en cours sur des sessions iSCSI de tiers même à la fin de l'étape e). La réponse de TMF NE DOIT PAS être envoyée par la couche iSCSI de la cible avant la fin de l'étape d) et PEUT être envoyée à la fin de l'étape d) en dépit de ces transferts de données en cours jusque après l'étape e).

4.2.3.4 Sémantique d'interruption de multi tâches FastAbort

Le comportement de protocole défini dans ce paragraphe DEVRAIT être respecté par toutes les mises en œuvre iSCSI qui se conforment au présent document, en notant que certaines étapes ci-dessous peuvent n'être pas compatibles avec la sémantique de la [RFC3720]. Cependant, le comportement de protocole défini dans ce paragraphe DOIT être suivi par les mises en œuvre iSCSI sur une session iSCSI quand elles négocient la clé TaskReporting (paragraphe 13.23) à "FastAbort" sur cette session. L'exécution des demandes de TMF ABORT TASK SET, CLEAR TASK SET, LOGICAL UNIT RESET, TARGET WARM RESET, et TARGET COLD RESET consiste en la séquence d'actions suivantes dans l'ordre spécifié sur la partie spécifiée.

La couche iSCSI initiatrice :
a) NE DOIT PAS envoyer d'autre PDU Data-Out pour des tâches affectées sur la connexion productrice de la session productrice iSCSI une fois que la TMF est envoyée à la cible.
b) DEVRAIT traiter toutes les réponses reçues pour des tâches affectées de la façon normale. C'est acceptable parce que il est garanti que les réponses ont été envoyées avant la réponse de TMF.
c) DOIT répondre à chaque PDU Message Async avec un AsyncEvent (5) "Terminaison de tâche" comme défini au paragraphe 11.9.
d) DOIT traiter la réponse de TMF comme terminant toutes les tâches affectées pour lesquelles des réponses n'ont pas été reçues et DOIT éliminer toute réponse pour des tâches affectées reçues après que la réponse de TMF a été passée à la couche SCSI (bien que la sémantique définie dans ce paragraphe assure qu'un tel scénario en désordre ne va jamais arriver avec une mise en œuvre de cible conforme).

La couche iSCSI cible :
a) DOIT attendre que toutes les commandes des tâches affectées soient reçues sur la base de l'ordre de CmdSN sur la session productrice. NE DEVRAIT PAS attendre de nouvelles commandes sur des sessions affectées de tiers – seules les tâches instanciées doivent être prises en compte pour déterminer les tâches affectées. Dans le cas de demandes à portée de cible (c'est-à-dire, TARGET WARM RESET et TARGET COLD RESET) toutes les commandes qui ne sont pas encore reçues sur la session productrice dans le flux de commandes peuvent être considérées comme ayant été reçues sans période d'attente de commande -- c'est-à-dire que l'espace CmdSN entier jusqu'au CmdSN de la fonction de gestion des tâches peut être "bouché".
b) DOIT propager la demande de TMF à, et recevoir la réponse de, la couche cible SCSI.
c) DOIT laisser tous les "TTT affectés" actifs (c'est-à-dire, les TTT actifs associés aux tâches affectées) valides.

d) DOIT envoyer une PDU Message asynchrone avec AsyncEvent = 5 (paragraphe 11.9) sur :
   1) chaque connexion de chaque session tierce à laquelle au moins une tâche affectée est soumise si TaskReporting=FastAbort est opérationnel sur cette session tierce, et
   2) chaque connexion sauf la connexion productrice de la session productrice qui a au moins une tâche affectée soumise. Si il y a plusieurs LU affectées (disons, à cause d'une réinitialisation de cible) alors une PDU Message asynchrone DOIT être envoyée pour chacune de ces LU sur chaque connexion qui a au moins une tâche affectée soumise. Le champ LUN dans la PDU Message asynchrone DOIT être réglé à correspondre au LUN pour chacune de ces LU.

e) DOIT adresser le fanion réponse close sur la réponse de TMF sur la session productrice comme défini au paragraphe 4.2.2.3.3.

f) DOIT adresser le fanion réponse close sur la première réponse post TMF sur les sessions tierces comme défini au paragraphe 4.2.2.3.3. Si certaines tâches proviennent de lien I_T_L non iSCSI, les moyens par lesquels la cible s'assure que toutes les tâches affectées ont retourné leur état à l'initiateur sont définis par le ou les protocoles de transport non iSCSI spécifiques.

g) DOIT libérer les TTT affectées (et les STags pour iSER, si applicable) et les mémoires tampons correspondantes, si il en est, une fois reçue chaque accusé de réception NOP-Out associé que l'initiateur a généré en réponse à chaque message asynchrone.

Techniquement, le service de TMF s'achève à l'étape e). Les transferts de données correspondants pour terminer les tâches peuvent, cependant, être encore en cours même à la fin de l'étape f). Une réponse de TMF NE DOIT PAS être envoyée par la couche iSCSI de la cible avant la fin de l'étape e) et PEUT être envoyée à la fin de l'étape e) en dépit de ces transferts de données en cours jusqu'à l'étape g). L'étape g) spécifie un événement pour libérer toutes ces ressources qui peuvent avoir été réservées pour la prise en charge des transferts de données en cours.

4.2.3.5 Tâches affectées partagées sur des sessions standard et FastAbort

Si une mise en œuvre de cible iSCSI est capable de prendre en charge la fonctionnalité TaskReporting=FastAbort (paragraphe 13.23) elle peut finir dans une situation où certaines sessions ont TaskReporting=RFC3720 opérationnel (sessions RFC 3720) tandis que d'autres sessions ont TaskReporting=FastAbort opérationnel (sessions FastAbort) même lors de l'accès à un ensemble partagé de tâches affectées (paragraphe 4.2.3.2). Si la session productrice est une session de la RFC 3720, la mise en œuvre de cible iSCSI est à capacité FastAbort, et la session affectée de tiers est une session FastAbort, le comportement suivant DEVRAIT être suivi pas la couche cible iSCSI :

a) Entre les étapes c) et d) du comportement de la cible du paragraphe 4.2.3.3, envoyer une PDU Message asynchrone avec AsyncEvent=5 (paragraphe 11.9) sur chaque connexion de chaque session tierce à laquelle au moins une tâche affectée est soumise. Si il y a plusieurs LU affectées, envoyer alors une PDU Message asynchrone pour chacune de ces LU sur chaque connexion qui a au moins une tâche affectée soumise. Lorsque c'est envoyé, le champ LUN dans la PDU Message asynchrone DOIT être réglé de façon à correspondre au LUN pour chacune de ces LU.

b) Après l'étape e) du comportement de la cible du paragraphe 4.2.3.3, libérer les TTT affectés (et les STags pour iSER, si applicable) et les mémoires tampon correspondantes, si il en est, une fois que chaque accusé de réception NOP-Out associé est reçu, que l'initiateur tiers a généré en réponse à chaque Message asynchrone envoyé dans l'étape a).

Si la session productrice est une session FastAbort, la mise en œuvre de cible iSCSI est à capacité FastAbort, et la session affectée de tiers est une session de la RFC 3720, la couche cible iSCSI NE DOIT PAS envoyer de PDU Message asynchrone sur la session tierce pour inviter au comportement FastAbort.

Si la session affectée de tiers est une session FastAbort et si la session productrice est une session FastAbort, l'initiateur dans le rôle de tiers DOIT répondre à chaque PDU Message asynchrone avec AsyncEvent=5 comme défini au paragraphe 11.9. Noter qu'un initiateur PEUT donc recevoir ces messages asynchrones sur une session affectée de tiers même si la session est une session à une seule connexion.

4.2.3.6 Raisons de la sémantique de FastAbort

Il y a fondamentalement trois objectifs de base derrière la sémantique spécifiée aux paragraphes 4.2.3.3 et 4.2.3.4.

a) Maintenir un lien I_T abstrait de flux de commandes ordonné vers la couche cible SCSI même avec des sessions multi connexions.

- Le traitement de la cible iSCSI d'une demande de TMF doit maintenir l'illusion d'un seul flux. Le comportement de la cible à l'étape b) du paragraphe 4.2.3.3 et le comportement de la cible à l'étape a) du paragraphe 4.2.3.4 correspondent à cet objectif.

b) Maintenir un seul lien I_T abstrait de flux de réponse ordonné vers la couche d'initiateur SCSI même avec des sessions multi connexions quand une réponse (c'est-à-dire, une réponse de TMF) pourrait impliquer l'état d'autres tâches non terminées du point de vue de l'initiateur.

- La cible doit s'assurer que l'initiateur ne voit pas des réponses de "vieilles" tâches (qui ont été mises dans le réseau plus tôt que la réponse de TMF) après avoir vu la réponse de TMF. Le comportement de la cible à l'étape d) du paragraphe 4.2.3.3 et le comportement de la cible à l'étape e) du paragraphe 4.2.3.4 correspondent à cet objectif.
- Chaque fois que le résultat d'une action de TMF est visible sur plusieurs liens I_T_L, [SAM2] exige que l'appareil serveur iSCSI déclenche une UA sur chacun des autres liens I_T_L. Une fois qu'un initiateur est notifié d'une telle UA, l'application client chez l'initiateur qui la reçoit doit libérer son état des tâches (clause 5.5 de [SAM2]) pour les tâches affectées. Il serait donc inapproprié de livrer une réponse SCSI pour une tâche après que l'état de tâche a été libéré chez l'initiateur, c'est-à-dire, après la notification d'UA. La notification d'UA contenue dans la première PDU de réponse SCSI sur chaque lien I_T_L affecté après l'action de TMF NE DOIT donc passer les réponses de tâche affectée sur aucune des sessions iSCSI qui accèdent à la LU. Le comportement de la cible à l'étape e) du paragraphe 4.2.3.3 et le comportement de la cible à l'étape f) du paragraphe 4.2.3.4 correspond à cet objectif.

c) Écouler toutes les TTT actives correspondant aux tâches affectées d'une façon déterministe.
- Les PDU Data-Out avec des TTT périmées qui arrivent après la fin des tâches peuvent créer un problème de gestion de mémoire tampon même pour des mises en œuvre iSCSI traditionnelles et c'est fatal pour la connexion pour les mises en œuvres iSCSI/iSER. Soit la terminaison des tâches affectées devrait être retardée jusqu'à ce que les TTT soient retirées (comme dans l'étape a) du paragraphe 4.2.3.3) soit les TTT et les mémoires tampon devraient rester allouées au delà de la terminaison des tâches pour être libérées ultérieurement de façon déterministe (comme dans les étapes c) et g) du paragraphe 4.2.3.4).

La seule autre optimisation notable est le bouchage. Si toutes les tâches sur un lien I_T vont de toutes façons être interrompues (comme avec la réinitialisation de cible) il n'est pas besoin d'attendre de recevoir toutes les commandes pour boucher les trous de CmdSN. La couche iSCSI cible peut simplement boucher tous les intervalles de numéros de séquence de commande manquants et continuer le traitement de TMF. Le premier objectif (maintenir un seul flux de commandes ordonné) est quand même satisfait avec cette optimisation parce que la couche cible SCSI ne voit que les commandes ordonnées.

4.2.4 Établissement iSCSI

L'objet de l'établissement iSCSI est d'activer une connexion TCP pour l'usage d'iSCSI, pour l'authentification des parties, la négociation des paramètres de la session et le marquage de la connexion comme appartenant à une session iSCSI.

Une session est utilisée pour identifier à une cible toutes les connexions avec un certain initiateur qui appartiennent au même lien I_T. (Pour des précisions sur la façon dont une session se rapporte à un lien I_T, voir au paragraphe 4.4.2).

Les cibles écoutent sur un accès TCP bien connu ou sur un autre accès TCP pour les connexions entrantes. L'initiateur commence le processus d'établissement en se connectant à un de ces accès TCP.

Au titre du processus d'établissement, l'initiateur et la cible DEVRAIENT s'authentifier l'un l'autre et PEUVENT établir un protocole d'association de sécurité pour la session. Cela peut se faire de nombreuses façons différentes et est l'objet de négociations.

Pour protéger la connexion TCP, une association de sécurité IPsec PEUT être établie avant la demande de connexion. Pour des informations sur l'utilisation de la sécurité IPsec pour iSCSI voir la Section 9 et les [RFC3723] et [RFC7146].

La phase d'établissement iSCSI est réalisée par des demandes et réponses d'établissement. Une fois qu'une authentification convenable est effectuée et que les paramètres de fonctionnement ont été établis, la session passe à la phase de pleine caractéristiques et l'initiateur peut commencer à envoyer des commandes SCSI. La politique de sécurité par laquelle, et les moyens par lesquels, une cible choisit d'autoriser un initiateur, sort du domaine d'application du présent document. Pour une description plus détaillée de la phase d'établissement, voir la Section 6.

La PDU d'établissement inclut la partie ISID de l'identifiant de session (SSID). Le groupe de portail cible qui dessert la connexion est impliqué par le choix des points d'extrémité de connexion. Pour une nouvelle session, le TSIH est zéro. Au titre de la réponse, la cible génère un TSIH.

Durant l'établissement de session, la cible identifie l'accès d'initiateur SCSI (le "I" dans le "nexus I_T") par la paire de valeurs (Nom_d'initiateur, ISID). On décrit Nom_d'initiateur plus loin dans cette section. Tout état persistant (par exemple, des réservations persistantes) sur la cible, qui est associé à un accès d'initiateur SCSI, est identifié sur la base de cette paire de valeurs. Tout état associé à l'accès de cible SCSI (le "T" dans le "nexus I_T") est identifié en externe par le Nom_de_cible et l'étiquette de groupe portail (voir au paragraphe 3.4.1 "Modèle d'architecture iSCSI"). ISID est soumis à des restrictions de réutilisation parce qu'il sert à identifier un état persistant (voir au paragraphe 4.4.3).

Avant l'établissement de la phase de pleines caractéristiques, seules les PDU de demande d'établissement et de réponse de connexion sont admises. Les demandes et réponses d'établissement DOIVENT être utilisées exclusivement durant l'établissement. Sur toute connexion, la phase Établissement DOIT immédiatement suivre l'établissement de la connexion TCP et une phase d'établissement suivante NE DOIT PAS survenir avant la suppression de la connexion.

Une cible qui reçoit une PDU sauf de demande d'établissement avant le début de la phase d'établissement DOIT immédiatement terminer la connexion sur laquelle la PDU a été reçue. Une fois la phase Établissement commencée, si la cible reçoit toute PDU autre qu'une demande Établissement, elle DOIT envoyer un Rejet d'établissement (avec l'état "invalide durant l'établissement ") et ensuite déconnecter. Si l'initiateur reçoit toute PDU autre qu'une réponse d'établissement, il DOIT immédiatement terminer la connexion.

4.2.5 Phase pleines caractéristiques iSCSI

Une fois que l'initiateur est autorisé à faire ainsi, la session iSCSI est dans la phase Pleines caractéristiques iSCSI. Une session est dans la phase Pleines caractéristiques après avoir achevé avec succès la phase Connexion sur la première connexion (principale) d'une session. Une connexion est dans la phase Pleines caractéristiques si la session est en phase Pleines caractéristiques et si l'établissement de la connexion s'est réalisé avec succès. Une connexion iSCSI n'est pas en phase Pleines caractéristiques :

a) lorsque elle n'a pas établi une connexion de transport, OU
b) si elle a une connexion de transport valide, mais qu'un établissement réussie n'a pas été effectué ou que la connexion est actuellement désétablie.

Dans une phase Pleines caractéristiques normale, l'initiateur peut envoyer des commandes SCSI et des données aux diverses LU sur la cible en les encapsulant dans les PDU iSCSI qui s'écoulent sur la session iSCSI établie.

4.2.5.1 Allégeance de connexion par commande

Pour toute demande iSCSI produite sur une connexion TCP, la réponse correspondante et/ou la ou les autres PDU en relation avec elle DOIVENT être envoyées sur la même connexion. On appelle cela "allégeance de connexion". Si la connexion d'origine échoue avant l'achèvement de la commande, l'allégeance de connexion de la commande peut être explicitement réallouée à une connexion de transport différente, comme décrit en détail au paragraphe 7.2.

Donc, si un initiateur produit une commande READ, la cible DOIT envoyer les données requises, s'il en est, suivies par l'état à l'initiateur sur la même connexion TCP qu'utilisée pour livrer la commande SCSI. Si un initiateur produit une commande WRITE, l'initiateur DOIT envoyer les données, s'il en est, pour cette commande, sur la même connexion TCP qu'utilisée pour livrer la commande SCSI. La cible DOIT retourner Prêt au transfert (R2T), s'il en est, et l'état sur la même connexion TCP qui a été utilisée pour livrer la commande SCSI. Les demandes de retransmission (les PDU SNACK) ainsi que les données et l'état qu'elles génèrent DOIVENT aussi utiliser la même connexion.

Cependant, des commandes consécutives qui font partie d'une tâche SCSI de commande-chaîne liée (voir [SAM2]) PEUVENT utiliser des connexions différentes. L'allégeance de connexion est strictement par commande et non par tâche. Durant la phase iSCSI Pleines caractéristiques, l'initiateur et la cible PEUVENT entrelacer sur la session des commandes SCSI qui ne sont pas en rapport les unes avec les autres, leurs données SCSI, et leurs réponses.

4.2.5.2 Généralités sur le transfert de données

Les données SCSI sortantes (données d'utilisateur d'initiateur à cible ou paramètres de commande) sont envoyées comme données sollicitées ou non sollicitées. Les données sollicitées sont envoyées en réponse aux PDU R2T. Les données non sollicitées peuvent être envoyées au titre d'une PDU de commande iSCSI ("données immédiates") ou dans des PDU de données iSCSI séparées.

Les données immédiates sont supposées être générées au décalage 0 dans la mémoire tampon d'écriture de l'initiateur SCSI (mémoire tampon de données sortantes). Toutes les autres PDU de données ont le décalage de mémoire tampon réglé explicitement dans l'en-tête de la PDU.

Un initiateur peut envoyer des données non sollicitées jusqu'à FirstBurstLength *(longueur de première salve)* (voir au paragraphe 13.14) comme immédiates (selon la longueur maximum de PDU négociée) dans une séquence de PDU séparée, ou les deux. Toutes les données suivantes DOIVENT être sollicitées. La longueur maximum d'une PDU de données individuelle ou la partie immédiate de la première salve non sollicitée PEUT être négociée à l'établissement.

La quantité maximum de données non sollicitées qui peuvent être envoyées avec une commande est négociée au moment de l'établissement au moyen de la clé FirstBurstLength (voir au paragraphe 13.14). Une cible PEUT activer séparément les

données immédiates (au moyen de la clé ImmediateData) sans activer la forme plus générale (les PDU de données séparées) de données non sollicitées (au moyen de la clé InitialR2T).

Les données non sollicitées en écriture sont destinées à réduire l'effet de la latence sur le débit (aucun R2T n'est nécessaire pour commencer à envoyer des données). De plus, les données immédiates sont destinées à réduire les frais généraux du protocole (à la fois en bande passante et en temps d'exécution).

Un initiateur iSCSI PEUT choisir de ne pas envoyer de données non sollicitées, seulement des données immédiates ou FirstBurstLength octets de données non sollicitées avec une commande. Si des données non sollicitées non immédiates sont envoyées, le total des données non sollicitées DOIT être soit FirstBurstLength, soit toutes les données si la quantité totale est inférieure à FirstBurstLength.

On considère que c'est une erreur pour un initiateur d'envoyer des PDU de données non sollicitées à une cible qui fonctionne en mode R2T (seules des données sollicitées sont permises). C'est aussi une erreur pour un initiateur d'envoyer plus de données non sollicitées, immédiates ou comme PDU séparés, que FirstBurstLength.

Un initiateur DOIT honorer une demande de données R2T pour une commande en instance valide (c'est-à-dire, qui porte une étiquette de tâche d'initiateur valide) et livrer toutes les données demandées pourvu que la commande soit supposée livrer des données sortantes et que le R2T spécifie des données dans les limites de la commande. L'action de l'initiateur est non spécifiée pour recevoir une demande R2T qui spécifie des données qui sont, en tout ou partie, en dehors des limites de la commande.

Une cible NE DEVRAIT PAS éliminer en silence des données et ensuite en demander la retransmission par un R2T. Les initiateurs NE DEVRAIENT PAS garder trace des données transférées de/vers la cible (tableau des résultats). Les cibles SCSI effectuent un calcul de compte résiduel pour vérifier combien de données ont en fait été transférées de/vers l'appareil par une commande. Cela peut différer de la quantité que l'initiateur a envoyé et/ou reçu pour des raisons telles que des retransmissions et des erreurs. Les commandes Lecture ou Bidirectionnel sollicitent implicitement la transmission de la quantité totale de données couverte par la commande. Les paquets de données SCSI sont confrontés à leurs commandes SCSI correspondantes en utilisant des étiquettes spécifiées dans le protocole.

De plus, les initiateurs et cibles iSCSI DOIVENT appliquer certaines règles de rangement. Lorsque des données non sollicitées sont utilisées, l'ordre des données non sollicitées sur chaque connexion DOIT correspondre à l'ordre dans lequel les commandes sont envoyées sur cette connexion. Une commande et des PDU de données non sollicitées peuvent être entrelacées sur une seule connexion pour autant que les exigences de rangement de chacune soient conservées (par exemple, la commande N + 1 PEUT être envoyée avant les PDU Data-Out non sollicitées pour la commande N, mais les PDU Data-Out non sollicitées pour la commande N DOIVENT précéder les PDU non sollicitées Data-Out de la commande N + 1). Une cible qui reçoit des données déclassées PEUT terminer la session.

4.2.5.3 Étiquettes et vérifications d'intégrité

Les étiquettes d'initiateur pour les commandes en cours sont uniques au niveau de l'initiateur pour une session. Les étiquettes de cible ne sont pas strictement spécifiées par le protocole. On suppose que les étiquettes de cible sont utilisées par la cible pour étiqueter (seules ou en combinaison avec le LUN) les données sollicitées. Les étiquettes de cible sont générées par la cible et l'initiateur leur fait "écho". Ces mécanismes sont conçus pour accomplir une livraison efficace des données ainsi qu'assurer un bon contrôle sur le flux des données.

Comme l'étiquette de tâche d'initiateur est utilisée pour identifier une tâche durant son exécution, l'initiateur et la cible iSCSI DOIVENT vérifier que tous les autres champs utilisés dans des PDU en rapport avec les tâches ont des valeurs qui sont cohérentes avec les valeurs utilisées à l'instanciation de la tâche sur la base de l'étiquette de tâche d'initiateur (par exemple, le LUN utilisé dans une PDU R2T DOIT être le même que celui utilisé dans la PDU de commande SCSI utilisée pour instancier la tâche). L'utilisation de valeurs de champ incohérentes est considérée comme une erreur de protocole.

4.2.5.4 Gestion de tâche SCSI durant la phase de pleines caractéristiques iSCSI

La gestion de tâche SCSI suppose que les tâches individuelles et les groupes de tâches peuvent être interrompus sur la seule base des étiquettes de tâche (pour les tâches individuelles) ou la temporisation de la commande de gestion de tâche (pour les groupes de tâches) et que l'action de gestion de tâche est exécutée de façon synchrone - c'est-à-dire, aucun message impliquant une tâche interrompue ne sera vu par l'initiateur SCSI après la réception d'une réponse de gestion de tâche. Dans iSCSI, initiateurs et cibles interagissent de façon asynchrone sur plusieurs connexions. iSCSI spécifie le mécanisme de protocole et les exigences de mise en œuvre nécessaires pour présenter une vue synchrone tout en utilisant une infrastructure asynchrone.

4.2.6 Terminaison de connexion iSCSI

Une connexion iSCSI peut être terminée par l'utilisation d'une fermeture de connexion de transport ou d'une réinitialisation de transport. Une réinitialisation de transport est supposée être un événement exceptionnel.

Les fermetures en douceur de connexion TCP sont faites par l'envoi de FIN TCP. Une fermeture en douceur de connexion de transport DEVRAIT être initialisée par l'une ou l'autre partie seulement lorsque la connexion n'est pas en phase de pleines caractéristiques iSCSI. Une cible PEUT terminer une connexion en phase de pleines caractéristiques sur des événements d'exception internes, mais elle DEVRAIT annoncer le fait par une PDU Message asynchrone. La fin de connexion avec des commandes en instance peut exiger des actions de récupération.

Si une connexion est terminée en phase de pleines caractéristiques, un nettoyage de connexion (voir la section 7) est nécessaire avant la récupération. En faisant le nettoyage de connexion avant de commencer la récupération, l'initiateur et la cible vont éviter de recevoir des PDU périmées après la récupération.

4.2.7 Noms iSCSI

Les cibles et initiateurs ont tous deux besoin de noms pour les besoins de l'identification. De plus, les noms permettent aux ressources de mémorisation iSCSI d'être gérées sans considération de leur localisation (adresse). Un nom de nœud iSCSI est aussi le nom d'appareil SCSI d'un appareil iSCSI. Le nom iSCSI d'un appareil SCSI est le principal objet utilisé dans l'authentification des cibles auprès des initiateurs, et des initiateurs auprès des cibles. Ce nom est aussi utilisé pour identifier et gérer les ressources de mémorisation iSCSI.

Les noms iSCSI doivent être uniques au sein du domaine de fonctionnement de l'utilisateur final. Cependant, comme le domaine opérationnel d'un réseau IP est potentiellement le monde entier, les formats de nom iSCSI sont architecturés comme étant uniques au monde. Pour aider les autorités de désignation à la construction de noms uniques au monde, iSCSI fournit trois formats de noms pour les différents types d'autorités de dénomination.

Les noms iSCSI sont associés à des nœuds iSCSI, et non à des cartes d'adaptateur réseau iSCSI, pour assurer que le remplacement des cartes d'adaptateur réseau n'exige pas de reconfiguration de toutes les informations d'allocation de ressources SCSI et iSCSI.

Certaines commandes SCSI exigent que des identifiants spécifiques du protocole soient communiqués au sein des CDB SCSI. Voir au paragraphe 2.2 la définition du nom/identifiant d'accès SCSI pour les accès iSCSI.

Un initiateur peut découvrir les noms de cibles iSCSI auxquelles il a accès, ainsi que leurs adresses, en utilisant la demande de texte SendTargets, ou d'autres techniques discutées dans la [RFC3721].

Un équipement iSCSI qui a besoin de fonctions de découverte allant au delà de SendTargets DEVRAIT mettre en œuvre iSNS (voir la [RFC4171]) pour des capacités étendues de découverte et d'interopérabilité. Bien que la [RFC3721] implique une exigence de mise en œuvre de SLP ([RFC2608]) SLP n'a pas en pratique été largement mis en œuvre ou déployé pour être utilisé avec iSCSI. Les mises en œuvre de iSCSI NE DEVRAIENT donc PAS compter sur l'interopérabilité avec la découverte fondée sur SLP.

4.2.7.1 Propriétés du nom iSCSI

Chaque nœud iSCSI, qu'il soit initiateur, cible, ou les deux, DOIT avoir un nom iSCSI. Chaque fois qu'un nœud iSCSI contient un nœud initiateur iSCSI et un nœud cible iSCSI, le nom d'initiateur iSCSI DOIT être le même que le nom de cible iSCSI pour les nœuds contenus de telle sorte qu'il y ait seulement un nom de nœud iSCSI pour l'ensemble du nœud iSCSI. Noter les exigences en rapport au paragraphe 9.2.1 sur la façon de transposer les noms CHAP en noms iSCSI dans un tel scénario.

Initiateurs et cibles DOIVENT prendre en charge la réception des noms iSCSI jusqu'à la longueur maximum de 223 octets.

L'initiateur DOIT présenter à la fois son nom d'initiateur iSCSI et le nom de cible iSCSI auquel il souhaite se connecter dans la première demande de connexion d'une nouvelle session ou connexion. La seule exception est si une session de découverte (voir au paragraphe 4.3) va être établie. Dans ce cas, le nom d'initiateur iSCSI est toujours exigé, mais le nom de cible iSCSI PEUT être omis.

Les noms iSCSI ont les propriétés suivantes :
- Les noms iSCSI sont uniques au monde. Deux initiateurs ou cibles ne peuvent pas avoir le même nom.
- Les noms iSCSI sont permanents. Un nœud iSCSI initiateur ou cible a le même nom pour toute la durée de sa vie.

- Les noms iSCSI n'impliquent pas une localisation ou adresse. Un initiateur ou cible iSCSI peut bouger, ou avoir plusieurs adresses. Un changement d'adresse n'implique pas un changement de nom.
- Les noms iSCSI ne s'appuient pas sur un courtier en noms central ; l'autorité de désignation est répartie.
- Les noms iSCSI acceptent l'intégration dans les schémas existants de désignation univoque.
- Les noms iSCSI s'appuient seulement sur les autorités de désignation existantes. ISCSI ne crée aucune nouvelle autorité de désignation.

Le codage d'un nom iSCSI a les propriétés suivantes :
- Les noms iSCSI ont la même méthode de codage sans considération des protocoles sous-jacents.
- Les noms iSCSI sont relativement simples à comparer. L'algorithme de comparaison de l'équivalence de deux noms iSCSI ne s'appuie pas sur un serveur externe.
- Les noms iSCSI sont composés seulement de caractères affichables et de caractères Unicode. Les noms iSCSI permettent l'utilisation des jeux de caractères internationaux mais les caractères majuscules sont interdits. Le profil iSCSI stringprep [RFC3722] transpose les caractères majuscules en caractères minuscules et DEVRAIT être utilisé pour préparer les noms iSCSI à partir d'entrées qui pourraient inclure des caractères majuscules. Aucun caractère d'espace n'est utilisé dans les noms iSCSI ; voir les détails dans la [RFC3722].
- Les noms iSCSI peuvent être transportés en utilisant aussi bien des protocoles binaires que fondés sur ASCII.

Un nom iSCSI désigne réellement une entité logicielle logique, et n'est pas lié à un accès ou autre matériel qui puisse être changé. Par exemple, un nom d'initiateur devrait désigner le nœud initiateur iSCSI, et non un NIC ou HBA particulier. Lorsque plusieurs NIC sont utilisés, ils devraient généralement tous présenter le même nom iSCSI d'initiateur aux cibles, parce que ils sont simplement des chemins sur la même couche SCSI. Dans la plupart des systèmes d'exploitation, l'entité désignée est l'image du système d'exploitation.

De même, un nom de cible ne devrait pas être lié à des interfaces matérielles qui peuvent être changées. Un nom de cible devrait identifier la cible logique et doit être le même pour la cible, sans considération de la portion physique à laquelle on s'adresse. Cela aide les initiateurs iSCSI à déterminer que les deux cibles découvertes sont en fait deux chemins pour la même cible.

Le nom iSCSI est conçu pour satisfaire aux exigences fonctionnelles des noms de ressource uniformes (URN, *Uniform Resource Name*) [RFC1737]. Par exemple, il est exigé que le nom ait une portée mondiale, soit indépendant de l'adresse ou de la localisation, et soit persistant et unique au monde. Les noms doivent être extensibles et adaptables à l'utilisation des autorités de désignation. Le codage de nom devrait être lisible à la fois par l'homme et la machine. Voir les autres exigences dans la [RFC1737].

4.2.7.2 Codage de nom iSCSI

Un nom iSCSI DOIT être un codage UTF-8 (voir la [RFC3629]) d'une chaîne de caractères Unicode ayant les propriétés suivantes :
- Elle est en forme de normalisation C (voir "Formes de normalisation Unicode" [UNICODE]).
- Elle contient seulement des caractères permis par la sortie du gabarit iSCSI stringprep (décrit dans la [RFC3722]).
- Les caractères suivants sont utilisés pour le formatage des noms iSCSI :
  - trait d'union ('-' = U+002d)
  - point ('.' = U+002e)
  - deux-points (':' = U+003a)
- Le codage UTF-8 du nom ne fait pas plus de 223 octets.

Le processus stringprep est décrit dans la [RFC3454] ; l'utilisation par iSCSI du processus stringprep est décrit dans la [RFC3722]. Stringprep est une méthode conçue par le groupe de travail Noms de domaines internationalisés (IDN) pour traduire des chaînes formées par le langage humain dans un format qui puisse être comparé comme des chaînes opaques. Les noms iSCSI sont supposés être utilisés par les administrateurs pour des besoins de configuration de système ; pour cette raison, les caractères qui peuvent conduire l'homme à la confusion entre différents noms iSCSI (par exemple, la ponctuation, les espaces, les marques diacritiques) devraient être évités, même lorsque de tels caractères sont permis comme résultat de traitement stringprep par la [RFC3722]. Le processus stringprep convertit aussi les chaînes en chaînes équivalentes de caractères minuscules.

Le processus stringprep n'a pas besoin d'être mis en œuvre si les noms sont seulement générés en utilisant des caractères permis comme résultat par le traitement stringprep spécifié dans la [RDC3722]. Ces caractères admis incluent tous les caractères minuscules et numériques ASCII, ainsi que les caractères minuscules Unicode comme spécifié dans la [RFC3722]. Une fois que les noms iSCSI codés en UTF-8 sont "normalisés" ils peuvent être comparés octet par octet en toute sécurité.

4.2.7.3 Structure de nom iSCSI

Un nom iSCSI consiste en deux parties – un désignateur de type suivi par une chaîne de nom univoque.

iSCSI utilise trois autorités de désignation existantes pour construire des noms iSCSI uniques au monde. Le désignateur de type dans un nom iSCSI indique l'autorité de désignation sur laquelle est fondé le nom. Les trois formats de nom iSCSI sont les suivants :

a) Nom qualifié iSCSI : fondé sur les noms de domaines pour identifier une autorité de désignation

b) Nom au format NAA : fondé sur un format de désignation défini par [FC-FS3] pour construire des identifiants uniques au monde, appelé l'autorité des adresses réseau (NAA, *Network Address Authority*)

c) Nom au format EUI : fondé sur les noms EUI, où l'autorité d'enregistrement de l'IEEE apporte son aide à la formation de noms uniques au monde (format EUI-64)

Les chaînes de type de désignateur correspondants actuellement définies sont :
a) iqn. : iSCSI Qualified name : nom qualifié iSCSI
b) naa. : Reste de la chaîne dans un identifiant d'autorité d'adresse réseau défini par le comité T11 de l'INCITS, en hexadécimal codé en ASCII
c) eui. : le reste de la chaîne string est un identifiant IEEE EUI-64 , en hexadécimal codé en ASCII.

Ces trois désignateurs d'autorité de désignation étaient considérés comme suffisants au moment de la rédaction du présent document. La création de désignateurs de type de désignation supplémentaires pour iSCSI peut être examinée par l'IETF et précisée dans des RFC distinctes.

Le tableau qui suit résume les protocole de transport SCSI courants et leurs formats de désignation.

Protocole de transport SCSI	**Format de désignation**		
	EUI-64	**NAA**	**IQN**
iSCSI (Internet SCSI)	X	X	X
FCP (Fibre Channel)		X	
SAS (Serial Attached SCSI)		X	

4.2.7.4 Type "iqn." (nom qualifié iSCSI)

Ce type de nom iSCSI peut être utilisé par toute organisation qui possède un nom de domaine. Ce format de désignation est utile lorsque un utilisateur final ou fournisseur de service souhaite allouer les noms iSCSI de cibles et/ou initiateurs.

Pour générer des noms de ce type, la personne ou organisation qui génère le nom doit posséder un nom de domaine enregistré. Ce nom de domaine n'a pas à se résoudre en une adresse ; il a juste besoin d'être réservé pour empêcher d'autres usagers de générer des noms iSCSI utilisant le même nom de domaine.

Comme un nom de domaine peut expirer, être acquis par une autre entité, ou peut être utilisé pour générer des noms iSCSI par les deux possesseurs, le nom de domaine doit être qualifié en plus par une date durant laquelle l'autorité de désignation a possédé le nom de domaine. Pour cette raison, un code de date est fourni au titre du format "iqn.".

La chaîne de nom qualifié iSCSI consiste en :
- La chaîne "iqn.", utilisée pour distinguer ces noms des noms formatés en "eui.".
- Un code de date, en format aaaa-mm. Cette date DOIT être une date durant laquelle l'autorité de désignation a possédé le nom de domaine utilisé dans ce format, et DEVRAIT être le premier mois dans lequel le nom de domaine a été possédé par cette autorité de désignation à 00:01 GMT du premier jour du mois. Ce code de date utilise le calendrier grégorien. Les quatre chiffres de l'année doivent être présents. Les deux chiffres du mois doivent être présents, avec janvier = "01" et décembre = "12". Le trait d'union doit être inclus.
- Un point ".".
- Le nom de domaine inversé de l'autorité de désignation (personne ou organisation) créant ce nom iSCSI.
- Une chaîne facultative, précédée de deux points (:) dans les limites du jeu de caractères et de la longueur que le possesseur du nom de domaine juge appropriées. Cela peut inclure des types de produit, des numéros de série, des identifiants d'hôtes, ou des clés logicielles (par exemple, cela peut inclure des caractères deux-points pour séparer les limites d'organisations). À l'exception du préfixe deux-points, le possesseur de nom de domaine peut tout allouer à son gré après le nom de domaine inversé. Il est de la responsabilité de l'entité qui est l'autorité de désignation de s'assurer que les noms iSCSI qu'elle alloue sont uniques au monde. Par exemple, "Exemple de dispositifs de mémorisation, SA.", peut posséder le nom de domaine "exemple.com".

Voici des exemples de nom qualifié iSCSI qui pourraient être générés par "EXEMPLE de dispositifs de mémorisation, SA."

Type	Date	Autorité de désignation	Chaîne définie par l'autorité de désignation "exemple.com"
iqn.	2001-04.	com.	exemple:storage:diskarrays-sn-a8675309
iqn.	2001-04.	com.	exemple
iqn.	2001-04.	com.	exemple:storage.tape1.sys1.xyz
iqn.	2001-04.	com.	exemple:storage.disk2.sys1.xyz

4.2.7.5 Type "eui." (format IEEE EUI-64)

L'autorité d'enregistrement de l'IEEE fournit un service pour allouer des identifiants uniques au monde [EUI]. Le format EUI-64 est utilisé pour construire un identifiant mondial dans d'autres protocoles réseau. Par exemple, canal Fibre définit une méthode de codage dans un WorldWideName *(nom mondial)*. Plus d'informations sur l'enregistrement des identifiants EUI se trouvent dans [OUI].

Le format est "eui." suivi d'un identifiant EUI-64 (16 chiffres hexadécimaux codés en ASCII).

Exemple de nom iSCSI :

Type	Identifiant EUI-64 (hexadécimal codé en ASCII)
eui.	02004567A425678D

Le format IEEE EUI-64 de nom iSCSI peut être utilisé lorsque un fabricant est déjà enregistré auprès de l'autorité d'enregistrement IEEE et utilise les noms uniques au monde formatés en EUI-64 pour ses produits.

On trouvera plus d'exemples de construction de noms dans la [RFC3721].

4.2.7.6 Type "naa." (Network Address Authority)

La spécification INCITS T11 "Tramage et signalisation [FC-FS3] définit un format appelé format d'autorité d'adresse réseau (NAA, *Network Address Authority*) pour construire des identifiants uniques au monde qui utilisent diverses autorités d'enregistrement d'identifiants. Ce format d'identifiant est utilisé par les protocoles de transport canal fibre et SAS SCSI. Comme FC et SAS constituent une large fraction des accès SCSI de réseautage, le format NAA est largement utilisé pour les transports SCSI. L'objectif de la prise en charge par iSCSI d'une représentation directe d'un nom de format NAA est de faciliter la construction d'un nom d'appareil cible qui se traduise facilement à travers de multiples espaces de noms pour les appareils de mémorisation SCSI contenant des accès désservis par différents transports. Plus précisément, ce format permet des mises en œuvre dans lesquelles un identifiant NAA peut être alloué sur la base du nom de l'appareil SCSI pour une cible SCSI avec des accès SAS et des accès iSCSI.

Le format de désignation NAA iSCSI est "naa.", suivi par un identifiant NAA représenté en chiffres hexadécimaux codés en ASCII.

Un exemple de nom iSCSI avec une valeur de NAA de 64 bits suit :

```
Type  Identifiant NAA (hexadécimal codé en ASCII)
+--++--------------+
|  ||               |
naa.52004567BA64678D
```

Un exemple de nom iSCSI avec une valeur de NAA de 128 bits suit :

```
Type Identifiant NAA (hexadécimal codé en ASCII)
+--++------------------------------+
|  ||                              |
naa.62004567BA64678D0123456789ABCDEF
```

Le format de désignation NAA iSCSI peut être utilisé dans une mise en œuvre quand l'infrastructure pour générer des noms NAA uniques au monde est déjà en place parce que l'appareil contient des accès SAS et des accès iSCSI SCSI.

L'identifiant NAA formaté en représentation ASCII-hexadécimal a une taille maximum de 32 caractères (format NAA de 128 bits). Par suite, il n'y a pas de problème avec ce format de désignation qui excède la taille maximum pour les noms de nœud iSCSI.

4.2.8 État persistent

iSCSI n'exige aucun entretien d'état persistant entre les sessions. Cependant, dans certains cas, SCSI exige une identification persistante du nom d'accès d'initiateur SCSI (voir les paragraphes 4.4.2 et 4.4.3).

Les sessions iSCSI ne persistent pas à travers les opérations de mise sous tension et d'amorçage. Tous les paramètres de session et connexion iSCSI sont réinitialisés à la création de session et de connexion.

Les commandes persistent au delà de la fin de la connexion si la session persiste et si la récupération de commande est prise en charge au sein de la session. Cependant, lorsque une connexion est abandonnée, l'exécution des commandes, telle que perçue par iSCSI (c'est-à-dire, impliquant des échanges de protocole iSCSI pour les tâches affectées) est suspendue jusqu'à ce qu'une nouvelle allégeance soit établie par la fonction "réallocation de 'tâche" de gestion de tâches. Voir le paragraphe 11.5.

4.2.9 Synchronisation et pilotage de message

iSCSI présente une transposition du protocole SCSI dans TCP. Cette encapsulation est réalisée en envoyant des PDU iSCSI de longueurs variées. Malheureusement, TCP n'a pas de mécanisme incorporé pour signaler les limites de message à la couche TCP. iSCSI surmonte cet obstacle en plaçant la longueur du message dans l'en-tête du message iSCSI. Cela sert à délimiter la fin du message en cours ainsi que le début du message suivant.

Dans les situations où les paquets IP sont livrés dans l'ordre par le réseau, le tramage de message iSCSI ne pose pas de problème et les messages sont traités l'un après l'autre. En présence de réarrangement des paquets IP (c'est-à-dire, lorsque des trames sont abandonnées) les mises en œuvre TCP traditionnelles mémorisent les segments TCP "décalés" dans des mémoires tampon temporaires jusqu'à ce qu'arrivent les segments TCP manquants, moment auquel les données doivent être copiées dans les mémoires tampon de l'application. Dans iSCSI, il est souhaitable de piloter les données SCSI au sein de ces segments TCP décalés dans les mémoires tampon pré allouées de SCSI plutôt que de les mémoriser dans des mémoires tampon temporaires. Cela diminue le besoin de mémoires tampon de réassemblage dédiées ainsi que la latence et la bande passante qui se rapportent aux copies supplémentaires.

S'appuyer seulement sur les informations de "longueur de message" provenant de l'en-tête du message iSCSI peut rendre impossible de trouver les limites du message iSCSI dans les segments TCP suivants à cause de la perte d'un segment TCP qui contient la longueur du message iSCSI. Le ou les segments TCP manquants doivent être reçus avant qu'aucun des segments suivants puisse être conduit dans les mémoires tampon SCSI correctes (dû à l'incapacité de déterminer les limites de message iSCSI). Comme ces segments ne peuvent pas être conduits à la localisation correcte, ils doivent être sauvegardés dans des mémoires tampon temporaires qui doivent alors être copiées dans les mémoires tampon SCSI.

Différents schémas peuvent être utilisés pour retrouver la synchronisation. Les détails de ces schémas sortent du domaine d'application de la présente spécification de protocole, mais il suffit de noter que la [RFC4297] donne une vue d'ensemble du problème du placement direct des données sur les réseaux IP, et la [RFC5046] spécifie une extension de protocole pour iSCSI qui facilite cet objectif de placement direct des données. Le reste de ce document se réfère à cet usage de protocole de placement direct des données comme un exemple d'une"couche de synchronisation et de pilotage".

Dans des circonstances normales (pas de perte de PDU ou de réception de données décalées) le pilotage des données iSCSI peut être réalisé en utilisant l'étiquette d'identification et les champs de décalage des données dans l'en-tête iSCSI en plus du numéro de séquence TCP provenant de l'en-tête TCP. L'étiquette d'identification aide à associer la PDU à une adresse de mémoire tampon SCSI alors que le décalage des données et le numéro de séquence TCP sont utilisés pour déterminer le décalage au sein de la mémoire tampon.

4.2.9.1 Synchronisation/pilotage et longueur de PDU iSCSI

Lorsque un grand message iSCSI est envoyé, le ou les segments TCP qui contiennent l'en-tête iSCSI peuvent être perdus. Le ou les segments TCP restants, jusqu'au prochain message iSCSI, doivent être mis en mémoire tampon (dans des mémoires temporaires) parce que l'en-tête iSCSI qui indique vers quelle mémoire tampon SCSI les données sont à diriger a été perdu. Pour minimiser la quantité de mémoire tampon, il est recommandé que la longueur de PDU iSCSI soit limitée à une petite valeur (peut-être quelques segments TCP de long). Durant la connexion, chaque extrémité de la session iSCSI spécifie la longueur maximum de PDU iSCSI qu'elle acceptera.

4.3 Types de session iSCSI

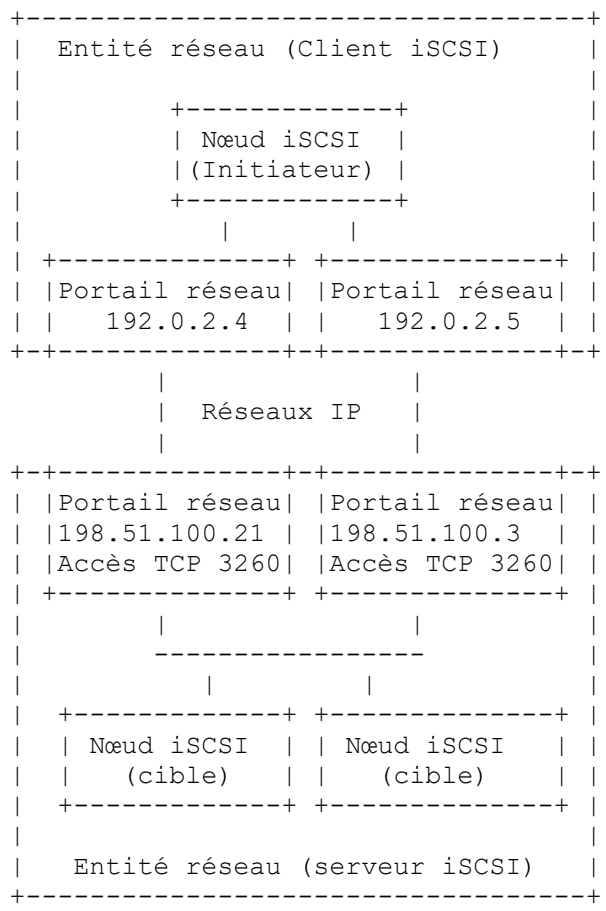
iSCSI définit deux types de sessions :

a) Session de fonctionnement normale – c'est une session sans restriction.
b) Session de découverte – c'est une session qui n'est ouverte que pour la découverte de cible. La cible DOIT seulement accepter les demandes de texte avec la clé SendTargets et une demande Désétablissement *(logout)* avec la raison "clore la session". Toute autre demande DOIT être rejetée.

Le type de session est défini durant l'établissement de connexion avec le paramètre SessionType=valeur dans la commande d'établissement *(login)*.

4.4 Modèles de transposition des concepts de SCSI à iSCSI

Le diagramme qui suit donne un exemple de la façon dont plusieurs nœuds iSCSI (des cibles dans ce cas) peuvent coexister au sein de la même entité réseau et peuvent partager des portails réseau (adresses IP et accès TCP). D'autres configurations plus complexes sont aussi possibles. Pour des descriptions détaillées des composants de ces diagrammes, voir au paragraphe 4.4.1.



4.4.1 Modèle d'architecture iSCSI

Ce paragraphe décrit la partie du modèle d'architecture iSCSI qui supporte la plus grande partie des relations entre iSCSI et le modèle d'architecture SCSI.

Entité réseau : représente un appareil ou passerelle qui est accessible depuis le réseau IP. Une entité réseau doit avoir un ou plusieurs portails réseau (voir l'élément "Portail réseau" ci-dessous) dont chacun peut être utilisé par des nœuds iSCSI (voir l'élément suivant)) contenus dans cette entité réseau pour obtenir l'accès au réseau IP.

Nœud SCSI : représente un seul initiateur ou cible iSCSI, ou une instance de chaque. Il y a un ou plusieurs nœuds iSCSI au sein d'une entité réseau. Le nœud iSCSI est accessible via un ou plusieurs portails réseau (voir ci-dessous). Un nœud iSCSI est identifié par son nom iSCSI (voir au paragraphe 4.2.7 et à la Section 13). La séparation du nom iSCSI des adresses utilisées par et pour le nœud iSCSI permet à plusieurs nœuds iSCSI d'utiliser les mêmes adresses, et au même nœud iSCSI d'utiliser plusieurs adresses.

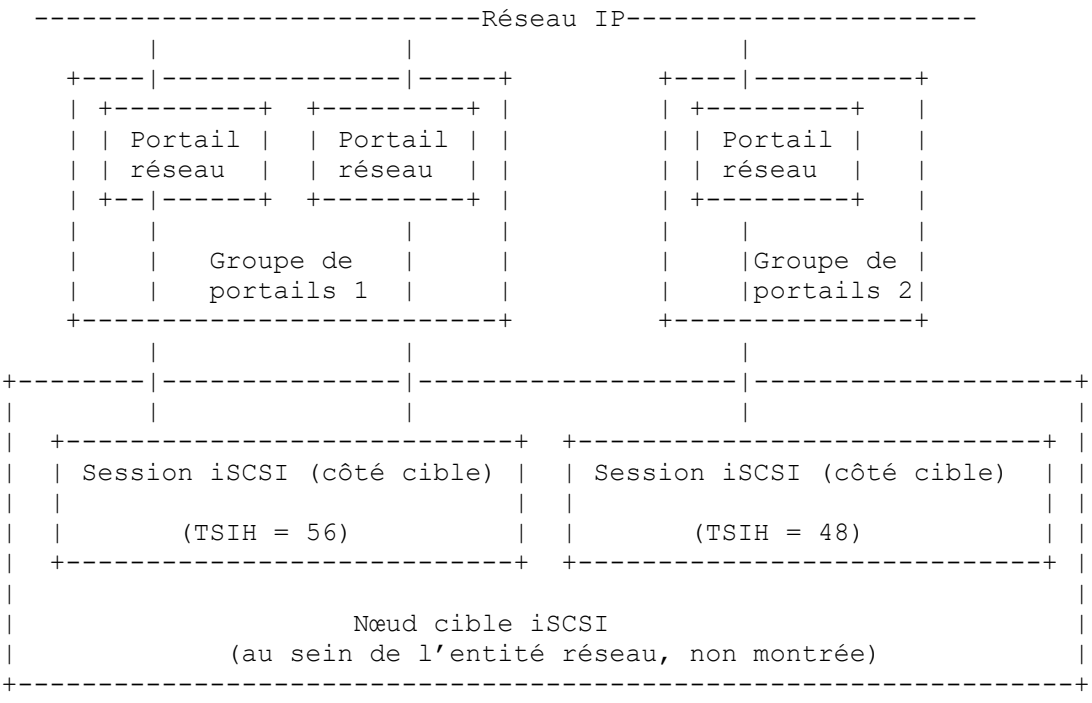
Une chaîne d'alias peut aussi être associée à un nœud iSCSI. L'alias permet à une organisation d'associer une chaîne facile à mémoriser au nom iSCSI. Cependant, la chaîne d'alias n'est pas un substitut du nom iSCSI.

Portail réseau : composant d'une entité réseau qui a une adresse réseau TCP/IP et qui peut être utilisé par un nœud iSCSI au sein de cette entité réseau pour la ou les connexions au sein d'une de ses sessions iSCSI. Chez un initiateur, il est identifié par son adresse IP. Chez une cible, il est identifié par son adresse IP et son accès d'écoute TCP.

Groupes de portails : iSCSI prend en charge plusieurs connexions au sein de la même session ; certaines mises en œuvre vont avoir la capacité de combiner des connexions dans une session à travers plusieurs portails réseau. Un groupe de portails définit un ensemble de portails réseau au sein d'un nœud iSCSI qui prend en charge collectivement la capacité de coordonner une session avec des connexions qui franchissent ces portails. Tous les portails réseau d'un groupe de portails n'ont pas besoin de participer à toute session connectée à ce groupe de portails. Un ou plusieurs groupes de portails peuvent fournir l'accès à un nœud iSCSI. Chaque portail réseau, tel qu'utilisé par un certain nœud iSCSI, appartient à exactement un groupe de portails au sein de ce nœud. Les groupes de portails sont identifiés au sein d'un nœud iSCSI par une étiquette de groupe de portails, un simple entier non signé entre 0 et 65 535 (voir au paragraphe 13.9). Tous les portails réseau qui ont la même étiquette de groupe de portail dans le contexte d'un certain nœud iSCSI sont dans le même groupe de portails. Les initiateurs et cibles iSCSI ont tous deux des groupes de portails, bien que seuls les groupes de portails de cible iSCSI soient utilisés directement dans le protocole iSCSI (par exemple, dans SendTargets). Pour les références aux groupes de portails d'initiateur, voir au paragraphe 10.1.2.

Au sein d'un groupe de portails, les portails devraient prendre en charge des paramètres de session similaires, parce qu'ils peuvent participer à une session commune.

Le diagramme qui suit donne un exemple d'une telle configuration sur une cible et sur la façon dont une session qui partage des portails réseau au sein d'un groupe de portails peut être établie.



4.4.2 Modèle d'architecture SCSI

Ce paragraphe décrit les relations entre le modèle d'architecture SCSI [SAM2] et la construction de l'appareil SCSI, de l'accès SCSI et du nexus I_T, avec la construction iSCSI décrite au paragraphe 4.4.1.

Cette relation implique des exigences de mise en œuvre afin de se conformer au modèle SAM2 et aux autres fonctions opérationnelles SCSI. Ces exigences sont détaillées au paragraphe 4.4.3.

La liste suivante souligne les transpositions des éléments architecturaux SCSI en iSCSI.

a) Appareil SCSI – c'est le terme SAM2 pour une entité qui contient un ou plusieurs accès SCSI qui sont connectés à un sous-système de livraison de service et prennent en charge un protocole d'application SCSI. Par exemple, un appareil initiateur SCSI contient un ou plusieurs accès d'initiateur SCSI et zéro, un ou plusieurs clients d'application. Un appareil cible SCSI contient un ou plusieurs accès de cible SCSI et une ou plusieurs unités logiques. Pour iSCSI, l'appareil SCSI est le composant au sein d'un nœud iSCSI qui fournit la fonctionnalité SCSI. À ce titre, il peut y avoir au plus un appareil SCSI au sein d'un nœud iSCSI. L'accès à l'appareil SCSI peut seulement être réalisé dans une session de fonctionnement iSCSI normale (voir au paragraphe 4.3). Le nom de l'appareil SCSI est défini comme étant le nom iSCSI du nœud et DOIT être utilisé dans le protocole iSCSI.

b) Accès SCSI : c'est le terme SAM2 pour une entité dans un appareil SCSI qui fournit la fonctionnalité SCSI à l'interface avec un sous-système de livraison de service ou de transport. Pour iSCSI, la définition de l'accès d'initiateur SCSI est différente de l'accès de cible SCSI.
Accès d'initiateur SCSI : il se transpose en un point d'extrémité d'une session normale de fonctionnement iSCSI (voir au paragraphe 4.3). Une session de fonctionnement iSCSI normale est négociée par le processus de connexion entre un nœud initiateur iSCSI et un nœud cible iSCSI. À l'achèvement réussi de ce processus, un accès d'initiateur SCSI est créé au sein de l'appareil initiateur SCSI. Le nom d'accès d'initiateur SCSI et l'identifiant d'accès d'initiateur SCSI sont tous deux définis comme étant le nom d'initiateur iSCSI avec (a) une étiquette qui l'identifie comme un nom/identifiant d'accès d'initiateur et (b) la portion ISID de l'identifiant de session.
Accès de cible SCSI : il se transpose en un groupe de portails cibles iSCSI. Le nom d'accès cible SCSI et l'identifiant d'accès cible SCSI sont tous deux définis comme étant le nom de cible iSCSI avec (a) une étiquette qui l'identifie comme un nom/identifiant d'accès de cible et (b) l'étiquette de groupe de portails cibles.
Le nom d'accès SCSI DOIT être utilisé dans iSCSI. Lorsque il est utilisé dans des données de paramètre SCSI, le nom d'accès SCSI DOIT être codé comme :
1) le nom iSCSI en format UTF-8, suivi par
2) une virgule de séparation (1 octet), suivie par
3) le caractère ASCII 'i' (pour accès d'initiateur SCSI) ou le caractère ASCII 't' (pour accès de cible SCSI) (1 octet) suivi par
4) une virgule de séparation (1 octet) suivie par
5) un codage texte comme constante hexadécimale (voir au paragraphe 6.1) de l'ISID (pour l'accès d'initiateur SCSI) ou l'étiquette de groupe de portails cibles (pour l'accès de cible SCSI) incluant le 0X ou 0x initial et le nul de terminaison (15 octets pour l'accès d'initiateur iSCSI, 7 octets pour l'accès cible iSCSI).

Le caractère ASCII 'i' ou 't' est l'étiquette qui identifie cet accès comme accès d'initiateur SCSI ou accès de cible SCSI.

c) Nexus I_T : Il indique une relation entre un accès d'initiateur SCSI et un accès de cible SCSI, conformément à [SAM2]. Pour iSCSI, cette relation est une session, définie comme une relation entre l'extrémité initiateur iSCSI de la session (accès d'initiateur SCSI) et le groupe de portails de la cible iSCSI. Le nexus I_T peut être identifié par la conjonction des noms d'accès SCSI ou par l'identifiant de session iSCSI (SSID). iSCSI définit l'identifiant de nexus I_T comme le tuplet (nom d'initiateur iSCSI + ",i,0x" + ISID en format texte, nom de cible iSCSI + ",t,0x" + étiquette de groupe de portails cible en format texte). Un préfixe hexadécimal en majuscules "0X" peut autrement être utilisé à la place de "0x".

Note : L'identifiant de nexus I_T n'est pas égal à l'identifiant de session (SSID).

4.4.3 Conséquences du modèle

Ce paragraphe décrit les exigences de mise en œuvre et de comportement qui résultent de la transposition des constructions SCSI en constructions iSCSI définies ci-dessus. Entre un certain accès d'initiateur SCSI et un certain accès de cible SCSI, seulement un nexus I_T (session) peut exister. Pas plus d'une relation de nexus (nexus parallèle) n'est admise par [SAM2]. Donc, à un moment donné, seule une session peut exister entre un certain nœud initiateur iSCSI et un nœud cible iSCSI, avec le même identifiant de session (SSID).

Ces hypothèses conduisent aux conclusions et exigences suivantes :

Règle ISID : Entre un certain initiateur iSCSI et un groupe de portails cibles iSCSI (accès de cible SCSI) il peut seulement y avoir une session avec une certaine valeur d'ISID qui identifie l'accès d'initiateur SCSI. Voir le paragraphe 1112.5.

La structure de l'ISID qui contient un composant d'autorité de désignation (voir au paragraphe 1112.5 et la [RFC3721]) donne un mécanisme pour faciliter la conformité avec la règle ISID. (Voir au paragraphe 10.1.1.)

Le nœud initiateur iSCSI devrait gérer l'allocation des ISID avant l'initiation de session. La "règle ISID" n'empêche pas d'utiliser le même ISID provenant du même initiateur iSCSI avec différents groupes de portails cibles sur la même cible iSCSI ou sur d'autres cibles iSCSI (voir au paragraphe 10.1.1). Permettre cela serait analogue à un seul accès d'initiateur SCSI ayant des relations (nexus) avec plusieurs accès de cible SCSI sur le même appareil cible SCSI ou accès de cible SCSI sur d'autres appareils cibles SCSI. Il est aussi possible d'avoir plusieurs sessions avec des ISID différents avec le même groupe de portails cibles. Chacune de ces sessions serait considérée comme étant avec un initiateur différent même lorsque les sessions ont pour origine le même appareil initiateur. Le même ISID peut être utilisé par un initiateur iSCSI différent parce que c'est le nom iSCSI avec l'ISID qui identifie l'accès d'initiateur SCSI.

Note : Une conséquence de la règle ISID et de la spécification de l'identifiant de nexus I_T est que deux nexus avec le même identifiant ne devraient jamais exister en même temps.

Règle TSIH : La cible iSCSI choisit une valeur non zéro pour le TSIH à la création de session (lorsque un initiateur présente une valeur 0 à la connexion). Après avoir été choisie, la même valeur de TSIH DOIT être utilisée chaque fois que l'initiateur ou la cible se réfère à la session et qu'un TSIH est requis.

4.4.3.1 État de nexus I_T

Certaines relations de nexus contiennent un état explicite (par exemple, des pages de mode spécifiques de l'initiateur) qui peut devoir être préservé par le serveur d'appareil [SAM2] dans une unité logique à travers les changements ou défaillances à la couche iSCSI (par exemple, des défaillances de session). Afin que cet état soit restauré, l'initiateur iSCSI devrait rétablir sa session (reconnexion) au même groupe de portails cibles en utilisant l'ISID précédant. C'est-à-dire qu'il devrait réinstaller la session via la réinstallation de session iSCSI (paragraphe 6.3.5) ou continuer via la continuation de session (paragraphe 6.3.6). Cela parce que l'identifiant d'accès d'initiateur SCSI et l'identifiant d'accès de cible SCSI (ou l'accès de cible qui s'y rapporte) forment les données qu'utilise le serveur d'appareil d'unité logique SCSI pour identifier le nexus I_T.

4.4.3.2 Réservations

Il y a deux méthodes de gestion de réservation définies dans les normes SCSI : les réservations reserve/release, sur la base des commandes RESERVE et RELEASE [SPC2], et les réservations persistantes, sur la base des commandes PERSISTENT RESERVE IN et PERSISTENT RESERVE OUT [SPC3]. Les réservations reserve/release sont obsolètes [SPC3] et ne devraient pas être utilisées. Les réservations persistantes sont suggérées comme solution de remplacement ; voir l'Annexe B de [SPC4].

Il est exigé que l'état pour les réservations persistantes persiste à travers les changements et défaillances à la couche iSCSI qui résultent en défaillances de nexus I_T ; voir dans [SPC3] les détails et les exigences spécifiques.

À l'opposé, [SPC2] ne spécifie pas d'exigences de persistance détaillées pour l'état de réservation reserve/release après une défaillance de nexux I_T. Néanmoins, quand les réservations reserve/release sont prises en charge par une cible iSCSI, l'approche de mise en œuvre préférée est de préserver l'état de réservation reserve/release pour la réinstallation de session iSCSI (voir le paragraphe 6.3.5) ou la continuation de session (voir le paragraphe 6.3.6).

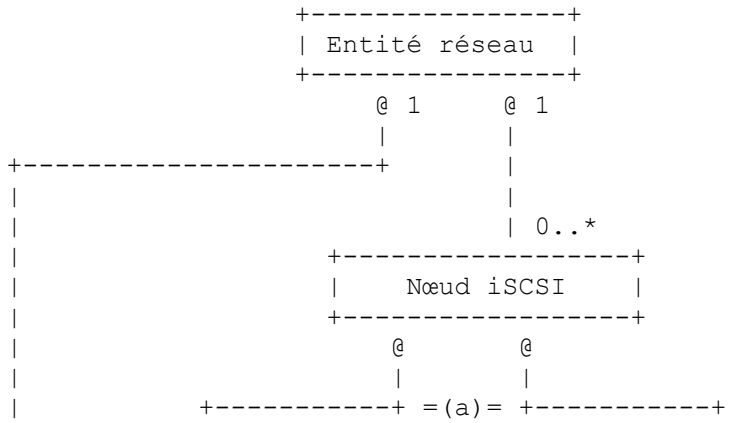
Deux avertissements supplémentaires s'appliquent aux réservations reserve/release :

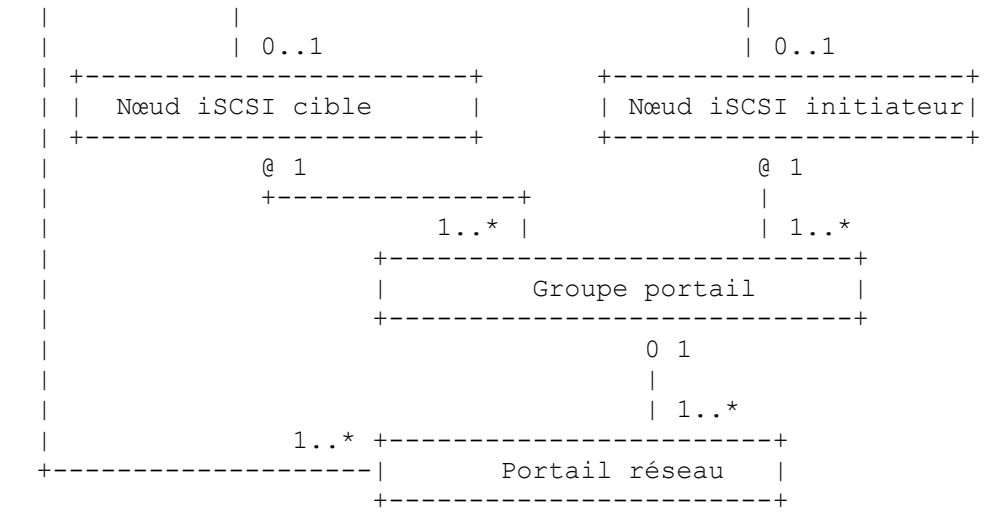
- La rétention de l'état de réservation reserve/release d'une session défaillante par une cible iSCSI, même après qu'une session iSCSI défaillante n'est ni réinstallée ni continuée, pour exiger qu'un initiateur produise une réinitialisation (par exemple, LOGICAL UNIT RESET ; voir le paragraphe 11.5) afin de supprimer cet état de réservation.

- Les réservations reserve/release peuvent ne pas se comporter comme attendu quand des réservations persistantes sont aussi utilisées sur la même LU ; voir la discussion de "Exceptions au comportement SPC-2 RESERVE et RELEASE" dans [SPC4].

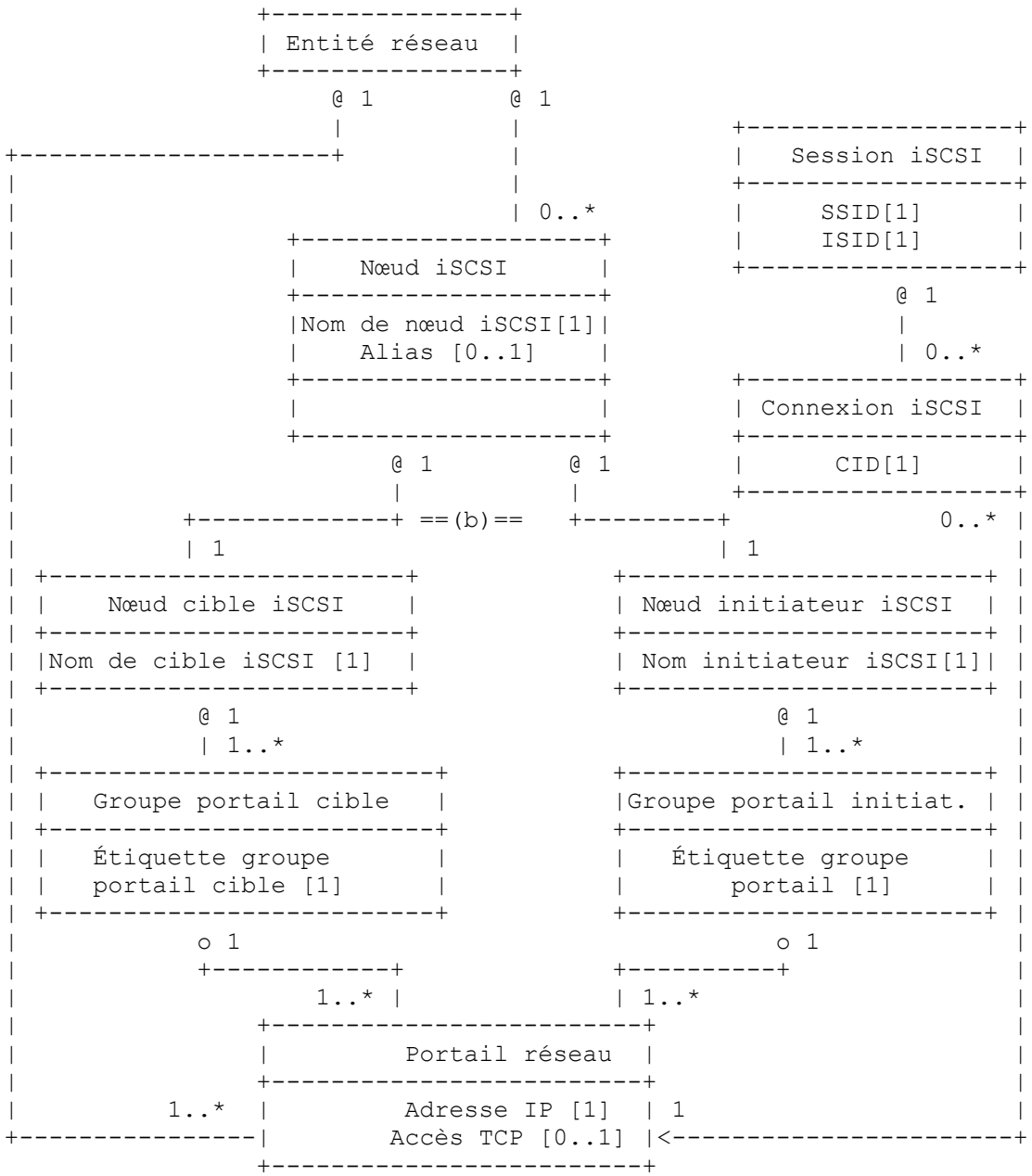
4.5 Modèle UML iSCSI

Ce paragraphe présente l'application des concepts de modélisation UML discutés à la Section 3 du modèle d'architecture iSCSI et SCSI exposé au paragraphe 4.4.





(a) Chaque instance d'une classe de nœud iSCSI DOIT contenir une instance de nœud cible iSCSI, une instance de nœud initiateur iSCSI, ou les deux.



(b) Chaque instance d'une classe de nœud iSCSI DOIT contenir une instance de nœud cible iSCSI, une instance de nœud initiateur iSCSI, ou les deux. Cependant, dans tous les scénarios, on note qu'un nœud iSCSI DOIT avoir un seul nom iSCSI. Noter que l'exigence qui s'y rapporte est au paragraphe 4.2.7.1.

4.6 Résumé des demandes/réponses

Ce paragraphe énumère et décrit brièvement tous les types de PDU iSCSI (demandes et réponses).

Toutes les PDU iSCSI sont construites comme un ensemble d'un ou plusieurs segments d'en-tête (de base et auxiliaires) et zéro ou un segment de données. Le groupe d'en-tête et le segment de données peuvent chacun être suivis d'un CRC (résumé).

Le segment d'en-tête de base a une longueur fixe de 48 octets.

4.6.1 Types de demande/réponse portant une charge utile SCSI

4.6.1.1 Commande SCSI

Cette demande porte le CDB *(bloc descripteur de commande)* SCSI et tous les autres arguments IN d'invocation de procédure de commande d'exécution SCSI (voir [SAM2]) tels que les attributs de tâche, la longueur de transfert de données attendue pour une direction de transfert ou les deux (cette dernière pour les commandes bidirectionnelles) et l'étiquette de tâche (au titre du nexus I_T_L_x). Le nexus I_T_L est déduit par l'initiateur et la cible du champ LUN *(Numéro d'unité logique)* dans la demande et le nexus I_T est implicite dans l'identification de session.

De plus, la PDU Commande SCSI porte les informations requises pour le bon fonctionnement du protocole iSCSI - le numéro de séquence de la commande (CmdSN) pour la session et le nombre d'états attendu (ExpStatSN) pour la connexion.

Tout ou partie des données de sortie SCSI (écriture) associées à la commande SCSI peut être envoyé au titre de la PDU Commande SCSI comme un segment de données.

4.6.1.2 Réponse SCSI

La réponse SCSI porte tous les arguments OUT de l'invocation de commande d'exécution de SCSI (voir [SAM2]) et la valeur de retour de l'invocation de procédure de commande d'exécution de SCSI.

La réponse SCSI contient le compte résiduel de l'opération, s'il en est, une indication de si le compte représente un débordement ou une sous alimentation, et de l'état SCSI si l'état est valide ou un code de réponse (une valeur de retour non zéro pour l'invocation de procédure de la commande d'exécution) si l'état n'est pas valide.

Pour un état valide qui indique que la commande a été traitée, mais a résulté en une exception (par exemple, une CHECK CONDITION SCSI) le segment de données de la PDU contient les données de sens associées. L'utilisation de Autosense ([SAM2]) est EXIGÉ par iSCSI.

Certains contenus de segment de données peuvent aussi être associés (dans le segment de données) avec un code de réponse non zéro.

De plus, la PDU Réponse SCSI porte les informations requises pour le bon fonctionnement du protocole iSCSI :
- ExpDataSN - le nombre de PDU Data-In qu'une cible a envoyées (pour permettre à l'initiateur de vérifier que toutes sont arrivées).
- Numéro de séquence d'état - le numéro de séquence d'état sur cette connexion.
- Numéro de séquence de commande attendu - le prochain numéro de séquence de commande attendu chez la cible.
- MaxCmdSN - le numéro de séquence de commande maximum acceptable chez la cible provenant de cet initiateur.

4.6.1.3 Demande de la fonction de gestion de tâche

La demande de fonction de gestion de tâche fournit à un initiateur un moyen de contrôler explicitement l'exécution d'une ou plusieurs tâches SCSI ou fonctions iSCSI. La PDU porte un identifiant de fonction (qui dit quelle fonction de gestion de tâche effectuer) et assez d'informations pour identifier sans équivoque la tâche ou l'ensemble de tâches sur lesquelles effectuer l'action, même si la ou les tâches sur lesquelles agir ne sont pas encore arrivées ou ont été éliminées suite à une erreur.

L'étiquette référencée identifie une tâche individuelle si la fonction se réfère à une tâche individuelle.

Le nexus I_T_L identifie des ensembles de tâches. Dans iSCSI le nexus I_T_L est identifié par le LUN et l'identification de session (la session identifie un nexus I_T).

Pour les ensembles de tâches, le numéro de séquence de commande de la demande de fonction de gestion de tâche aide à identifier les tâches sur lesquelles agir, à savoir toutes les tâches associées à un LUN et qui ont un numéro de séquence de commande précédant le numéro de séquence de commande de demande de fonction de gestion de tâche.

Pour une fonction de gestion de tâche, la coordination entre les réponses aux tâches affectées et la réponse de fonction de gestion de tâche est faite par la cible.

4.6.1.4 Réponse de fonction de gestion de tâche

La réponse de fonction de gestion de tâche porte une indication d'achèvement de fonction pour une demande de fonction de gestion de tâche incluant comment elle a été achevée (réponse et qualificatif) et des informations supplémentaires pour les réponses d'échec.

Lorsque la réponse de gestion de tâche indique l'achèvement de la fonction de gestion de tâche, l'initiateur ne va recevoir aucune réponse supplémentaire de la part des tâches affectées.

4.6.1.5 SCSI Data-Out et SCSI Data-In

SCSI Data-Out et SCSI Data-In sont les principaux véhicules par lesquels la charge utile de données SCSI est portée entre l'initiateur et la cible. La charge utile de données est associée à une commande SCSI spécifique à travers l'étiquette Tâche d'initiateur. Pour la convenance de la cible, les données sortantes sollicitées portent aussi une étiquette de transfert de cible *(Target Transfer Tag)* (copiée de R2T) et le LUN. Chaque PDU contient la longueur de charge utile et le décalage des données par rapport à l'adresse de mémoire tampon contenue dans l'invocation de procédure de la commande SCSI Execute.

Dans chaque direction, le transfert des données est partagé en "séquences". Une fin-de-séquence est indiquée par le bit F.

Une séquence sortante est soit non sollicitée (seule la première séquence peut être non sollicitée) soit consiste en toutes les PDU Data-Out envoyées en réponse à un R2T.

Les séquences entrantes permettent le changement de direction pour les commandes bidirectionnelles lorsque nécessaire.

Pour l'entrée, la cible peut demander un accusé de réception positif des données d'entrée. Ceci est limité aux sessions qui acceptent la récupération d'erreur et est mis en œuvre grâce au bit A dans l'en-tête de PDU SCSI Data-In.

Les PDU Data-In et Data-Out portent aussi le DataSN *(numéro de séquence de données)* pour permettre à l'initiateur et à la cible de détecter les PDU manquantes (éliminées suite à une erreur).
De plus, Numéro de séquence d'état est porté par les PDU Data-In.

Pour permettre de traiter une commande SCSI tout en impliquant un nombre minimum de messages, la dernière PDU SCSI Data-In passée pour une commande peut aussi contenir l'état si celui-ci indique une terminaison sans exception (pas de sens ou de réponse impliquée).

4.6.1.6 Prêt au transfert (R2T)

R2T *(Ready to Transfer)* est le mécanisme par lequel la cible SCSI "demande" à l'initiateur une sortie de données. R2T spécifie à l'initiateur le décalage des données demandées par rapport à l'adresse en mémoire tampon à partir de l'appel de procédure de commande Execute et la longueur des données sollicitées.

Pour aider la cible SCSI à associer les Data-Out résultantes à un R2T, le R2T porte une étiquette de transfert de cible *(*TTT, *Target Transfer Tag)* qui va être copiée par l'initiateur dans les PDU Data-Out SCSI sollicitées. Il n'y a pas d'exigence spécifique du protocole par rapport à la valeur de ces étiquettes, mais on suppose qu'avec le LUN, elles vont permettre à la cible d'associer les données à un R2T.

R2T porte aussi les informations requises pour le fonctionnement correct du protocole iSCSI, comme :
- R2TSN (pour permettre à un initiateur de détecter un R2T manquant)
- Numéro de séquence d'état
- Numéro de séquence de commande attendu
- MaxCmdSN

4.6.2 Demandes/réponses portant une charge utile SCSI et iSCSI

4.6.2.1 Message asynchrone

Les messages asynchrones sont utilisés pour porter des notifications d'événements SCSI asynchrones (AEN, *asynchronous event notification*) et des messages iSCSI asynchrones.

Lorsque ils portent une AEN, les détails de l'événement sont rapportés comme données de sens dans le segment de données.

4.6.3 Demandes/réponses portant une charge utile seulement iSCSI

4.6.3.1 Demandes et réponses de texte

Les demandes et réponses de texte sont conçues comme un véhicule de négociation de paramètre et comme un véhicule pour de futures extensions.

Dans le segment de données, les demandes et réponses de texte portent des informations textuelles utilisant une simple syntaxe "clé=valeur".

Les demandes/réponses de texte peuvent former des séquences étendues en utilisant la même étiquette de tâche d'initiateur. L'initiateur utilise le bit fanion F (Final) dans l'en-tête de la demande de texte pour indiquer qu'il est prêt à terminer une séquence. La cible utilise le bit fanion F dans l'en-tête de la réponse de texte pour indiquer son consentement à la terminaison de séquence.

Les demandes et réponses de texte utilisent aussi l'étiquette de transfert de cible pour indiquer la continuation d'une opération ou le commencement d'une nouvelle. Une cible qui souhaite continuer une opération va établir l'étiquette de transfert de cible dans une réponse de texte à une valeur différente du 0xffffffff par défaut. Un initiateur qui veut continuer va copier cette valeur dans l'étiquette de transfert de cible de la prochaine demande de texte. Si l'initiateur veut redémarrer la négociation de cible en cours (démarrage frais) il va établir l'étiquette de transfert de cible à 0xffffffff.

Bien qu'un échange complet soit toujours démarré par l'initiateur, des négociations spécifiques de paramètres peuvent être initiées par l'initiateur ou par la cible.

4.6.3.2 Demande et réponse d'établissement

Les demandes et réponses d'établissement *(Login)* sont utilisées exclusivement durant la phase Login de chaque connexion pour établir les paramètres de connexion et de session. (La phase Login consiste en une séquence de demandes et réponses d'établissement portant la même étiquette de tâche d'initiateur.)

Une connexion est identifiée par un identifiant de connexion (CID, *connexion identifier*) choisi de façon arbitraire qui est unique au sein d'une session.

Similaires aux demandes et réponses de texte, les demandes et réponses d'établissement portent des informations de texte clé=valeur avec une syntaxe simple dans le segment de données.

La phase Login se poursuit à travers plusieurs étapes (négociation de sécurité, négociation des paramètres de fonctionnement) qui sont choisies avec deux champs codés en binaire dans l'en-tête : l'étape en cours (CSG, *current stage*) et la prochaine étape (NSG, *next stage*) l'apparition de cette dernière étant signalée par le fanion "transit" (T).

La première phase Login d'une session joue un rôle particulier, appelé établissement de tête *(leading login)* qui détermine certains champs d'en-tête (par exemple, le numéro de version, le nombre maximum de connexions, et l'identification de la session).

La valeur initiale du numéro de séquence de commande est aussi établie par l'établissement de tête.

Pour chaque connexion, le numéro de séquence d'état est initié par l'établissement de tête.

Une demande Login peut indiquer un désétablissement impliqué (nettoyage) de la connexion à établir (un redémarrage de connexion) en utilisant le même identifiant de connexion (CID, *connexion identifier*) que celui d'une connexion existante, ainsi que les mêmes éléments d'identification de session que ceux de la session à laquelle la vieille connexion était associée.

4.6.3.3 Demande et réponse de désétablissement *(Logout)*

Les demandes et réponses Logout sont utilisées pour la fermeture régulière des connexions pour récupération ou maintenance. La demande de désétablissement peut être produite suite à l'invite d'une cible (par un message asynchrone) ou à l'initiative des initiateurs. Lorsque elle est produite sur la connexion à désétablir, aucune autre demande ne la suit.

La réponse Logout indique que le nettoyage de la connexion ou session est achevé et qu'aucune autre réponse n'arrivera sur la connexion (si elle est reçue sur la connexion en cours de désétablissement). De plus, la réponse Logout indique pendant combien de temps la cible va continuer de garder des ressources pour la récupération (par exemple, l'exécution d'une commande qui se continue sur une nouvelle connexion) dans le champ Time2Retain et combien de temps l'initiateur doit attendre avant de procéder à la récupération dans le champ Time2Wait.

4.6.3.4 Demande SNACK

Avec la demande SNACK, l'initiateur demande à la cible la retransmission de réponses ou données numérotées. Une seule demande SNACK couvre un ensemble contigu d'éléments manquants, appelé une séquence, d'un certain type d'éléments. Le type est indiqué dans un champ de type dans l'en-tête de PDU. La séquence se compose d'un élément initial (Numéro de séquence d'état, DataSN, R2TSN) et du nombre de PDU Status, Data, ou R2T manquées. Pour les longues séquences Data-In, la cible peut demander (à des intervalles minimum prédéfinis) un accusé de réception positif des données envoyées. Une demande SNACK avec un champ Type qui indique ACK et le nombre de PDU Data-In acquittées porte cet accusé de réception positif.

4.6.3.5 Rejet

Rejet permet à la cible de faire rapport d'une condition d'erreur iSCSI (par exemple, de protocole, d'une option non prise en charge) qui utilise un champ Cause dans l'en-tête de la PDU et comporte l'en-tête complet de la mauvaise PDU dans le segment de données de la PDU Rejet.

4.6.3.6 Demande NOP-Out et réponse NOP-In

Cette paire de demande/réponse peut être utilisée par un initiateur et une cible comme un mécanisme de "ping" pour vérifier qu'une connexion/session est encore active et que tous ses composants sont opérationnels. Un tel ping peut être déclenché par l'initiateur ou par la cible. Celui qui déclenche indique qu'il veut une réponse en établissant une valeur différente du 0xffffffff par défaut dans l'étiquette de transfert initiateur/cible correspondante.

NOP-In/NOP-Out peut aussi être utilisé de façon "unidirectionnelle" pour porter à la commande initiateur/cible des valeurs d'état ou de compteur de données lorsque il n'y a pas d'autre "porteur" et qu'il y a besoin de mettre à jour l'initiateur/cible.

5. Paramètres de mode SCSI pour iSCSI

Il n'y a pas de paramètre de mode spécifique de iSCSI.

6. Connexion et négociation de la phase de pleines caractéristiques

Les paramètres iSCSI sont négociés à l'établissement de la session ou connexion en utilisant les demandes et réponses Login (voir au paragraphe 4.2.4) et durant la phase Pleines caractéristiques (paragraphe 4.2.5) en utilisant les demandes et réponse Text. Dans les deux cas, le mécanisme utilisé est un échange de iSCSI-texte-clé=paires de valeurs. Pour faire court, les iSCSI-texte-clés sont juste appelées clés dans le reste du document.

Les clés sont soit déclaratives, soit elles exigent une négociation et la description de la clé indique si la clé est déclarative ou requiert une négociation.

Pour les clés déclaratives, la partie déclarante établit la valeur de la clé. La spécification de la clé indique si la clé peut être déclarée par l'initiateur, la cible ou les deux.

Pour les clés qui exigent une négociation, une des parties (la partie qui propose) propose une valeur ou ensemble de valeurs en incluant le clé=valeur dans la partie Données des PDU de demande ou réponse Login ou Text. L'autre partie (la partie qui accepte) fait un choix sur la base de la valeur ou de la liste de valeurs proposées et inclut la valeur choisie dans un

clé=valeur dans la partie Données d'une des PDU de demande ou réponse Login ou Text suivantes. Pour la plupart des clés, l'initiateur et la cible peuvent être la partie qui propose.

Le processus d'établissement se poursuit en deux étapes : l'étape de négociation de sécurité et l'étape de négociation des paramètres de fonctionnement. Les deux étapes sont facultatives mais au moins une d'elles doit être présente pour permettre l'établissement de certains paramètres obligatoires.

Si elle est présente, l'étape de négociation de la sécurité précède l'étape de la négociation de paramètres de fonctionnement.

La progression d'une étape à l'autre est contrôlée par le bit T (Transition) dans l'en-tête de PDU de demande/réponse Login. Lorsque le bit T est réglé à 1, l'initiateur indique qu'il aimerait que se fasse la transition. La cible accepte la transition (et choisit l'étape suivante) lorsque elle est prête. Un champ dans l'en-tête de PDU Login indique l'étape actuelle (CSG, *current stage*) et durant la transition, un autre champ indique la nouvelle étape (NSG, *next stage*) proposée (par l'initiateur) et choisie (par la cible).

Le processus de négociation text est utilisé pour négocier ou déclarer les paramètres de fonctionnement. Le processus de négociation est contrôlé par le bit F (final) dans l'en-tête de PDU. Durant les négociations text, le bit F est utilisé par l'initiateur pour indiquer qu'il est prêt à finir la négociation et par la cible pour acquiescer à la fin de la négociation.

Comme certaines paires de clé=valeurs peuvent ne pas tenir entièrement dans une seule PDU, le bit C (continuation) est utilisé (aussi bien dans Login que dans Text) pour indiquer qu'il y en a "à suivre".

La négociation Text utilise un mécanisme supplémentaire par lequel une cible peut délivrer de plus grandes quantités de données à un initiateur qui le demande. La cible établit une étiquette de tâche de cible *(Target Task Tag)* à utiliser comme marque page qui lorsque il est retourné par l'initiateur, signifie "vas y". Si il est établi à une "valeur neutre", il signifie "oublie tout le reste".

Cette section détaille les types de clés et valeurs utilisés, les règles de syntaxe pour la formation des paramètres, et les schémas de négociation à utiliser avec les différents types de paramètres.

6.1. Format Texte

L'initiateur et la cible envoient un ensemble de paires clé=valeur codées en UTF-8 Unicode. Toutes les clés de texte et les valeurs de texte spécifiées dans ce document sont sensibles à la casse ; elles doivent être présentées et interprétées comme elles apparaissent dans ce document sans changer la casse.

Les symboles de caractères suivants sont utilisés dans ce document pour les éléments text (Les valeurs hexadécimales représentent les codets Unicode) :

(a-z, A-Z) (0x61-0x7a, 0x41-0x5a) - lettres
(0-9) (0x30-0x39) - chiffres
" " (0x20) - espace
"." (0x2e) - point
"-" (0x2d) - moins
"+" (0x2b) - plus
"@" (0x40) – arobase (à commercial)
"_" (0x5f) - souligné
"=" (0x3d) - égal
":" (0x3a) – deux points
"/" (0x2f) – barre oblique
"[" (0x5b) – crochet gauche
"]" (0x5d) - crochet droit
nul (0x00) – séparateur nul
"," (0x2c) - virgule
"~" (0x7e) - tilde

Les paires clé=valeur peuvent s'étendre au delà des limites de PDU. Un initiateur ou cible qui envoie un texte partiel clé=valeur au sein d'une PDU indique que plus de texte suit en établissant le bit C dans la demande Text ou Login ou la réponse Text ou Login à 1. Les segments de données dans une série de PDU qui ont le bit C réglé à 1 et se terminent par une PDU qui a le bit C réglé à 0, ou qui incluent une seule PDU qui a le bit C réglé à 0, doivent être considérés comme formant un seul segment logique de données de texte (LTDS, *logical-text-data-segment*).

Chaque paire clé=valeur, incluant la dernière ou seule paire dans un LTDS, DOIT être suivie par un délimiteur nul (0x00).

Une clé-nom est ce qui précède le premier "=" dans la paire clé=valeur. Le terme "clé" est fréquemment utilisé dans ce document à la place de "clé-nom".

Une valeur est ce qui suit le premier "=" dans la paire clé=valeur jusqu'à la fin de la paire clé=valeur, mais non inclus le délimiteur nul.

Les définitions suivantes seront utilisées dans la suite de ce document:

- étiquette-standard : Chaîne d'un ou plusieurs caractères consistant en lettres, chiffres, point, signes moins, plus, arobase, ou souligné. Une étiquette-standard DOIT commencer par une majuscule et ne doit pas excéder 63 caractères.

- nom-clé : une étiquette-standard.

- valeur-texte : chaîne de zéro, un ou plusieurs caractères consistant en lettres, chiffres, points, signes moins, plus, arobase, souligné, barre oblique, crochet gauche, crochet droit, ou deux-points.

- valeur-nom-iSCSI : chaîne d'un ou plusieurs caractères consistant en caractères moins, point, deux-points ou tout autre, permis par le résultat du gabarit stringprep iSCSI spécifié dans la [RFC3722] (voir aussi paragraphe 4.2.7.2).

- valeur-nom-local-iSCSI : chaîne UTF-8 ; aucun caractère nul n'est admis dans la chaîne. Ce codage est à utiliser pour les alias localisés (internationalisés).

- valeur-booléenne : la chaîne "OUI" ou "NON".

- hex-constante : constante hexadécimale codée comme une chaîne qui commence par "0x" ou "0X" suivie par un ou plusieurs chiffres ou les lettres a, b, c, d, e, f, A, B, C, D, E, ou F. Les hex-constantes sont utilisées pour coder les valeurs numériques ou les chaînes binaires. Lorsque utilisé pour coder des valeurs numériques, l'usage excessif de zéros en tête est déconseillé. La chaîne suivant 0X (ou 0x) représente un nombre en base16 qui commence par le chiffre en base16 de poids fort, suivi par tous les autres chiffres en ordre de poids décroissant et se terminant par le chiffre base16 de moindre poids. Lorsque utilisé pour coder des chaînes binaires, les constantes hexadécimales ont une longueur d'octets implicite qui inclut quatre bits pour chaque chiffre hexadécimal de la constante, incluant les zéros en tête. Par exemple, une hex-constante de n chiffres hexadécimaux a une longueur de (partie entière de) (n + 1)/2 octets.

- décimal-constante : nombre décimal non signé avec le chiffre 0 ou une chaîne d'un ou plusieurs chiffres qui commence par un chiffre différent de zéro. Les décimal-constantes sont utilisées pour coder les valeurs numériques ou les chaînes binaires. Les décimal-constantes ne peuvent être utilisées pour coder des chaînes binaires que si la longueur de la chaîne est explicitement spécifiée. Il n'y a pas de longueur implicite pour les chaînes décimales. Les décimal-constantes NE DOIVENT PAS être utilisées pour des valeurs de paramètre si les valeurs peuvent être égales ou supérieures à 2**64 (numérique) ou pour les chaînes binaires qui peuvent faire plus de 64 bits.

- base64-constante : constante en base64 codée comme une chaîne qui commence par "0b" ou "0B" suivi par un ou plusieurs chiffres, lettres, signe plus, barre oblique, ou signe égal. Le codage est fait selon la [RFC4648].

- valeur-numérique : entier non signé toujours inférieur à 2**64 codé comme décimal-constante ou hex-constante. L'arithmétique d'entier non signé s'appliquee aux valeurs numériques.

- grande-valeur-numérique : entier non signé qui peut être supérieur ou égal à 2**64 codé comme hex-constante ou base64-constante. Les entiers arithmétiques non signés s'appliquent aux grandes-valeurs-numériques.

- gamme-numérique : deux valeurs-numériques séparées par un tilde, où la valeur à droite du tilde ne doit pas être inférieure à la valeur de gauche.

- valeur-binaire-régulière : chaîne binaire de pas plus de 64 bits codée comme décimal-constante, hex-constante, ou base64-constante. La longueur de la chaîne est soit spécifiée par la définition de clé, soit est la longueur en octets implicite de la chaîne codée.

- grande-valeur-binaire : chaîne binaire de plus de 64 bits codée comme hex-constante ou base64-constante. La longueur de la chaîne est soit spécifiée par la définition de clé, soit est la longueur d'octets implicite de la chaîne codée.

- valeur-binaire : valeur-binaire-régulière ou grande-valeur-binaire. Les opérations sur les valeurs binaires sont spécifiques de la clé.

- simple-valeur : valeur-texte, valeur-nom-iSCSI, valeur-booléenne, valeur-numérique, gamme-numérique, ou valeur-binaire.

- liste-de-valeurs : séquence de valeur-texte séparées par une virgule.

Sauf mention contraire, la longueur maximum d'une simple-valeur (pas de représentation codée) est 255 octets, non inclus le délimiteur (virgule ou octet zéro).

Toute cible ou initiateur iSCSI DOIT prendre en charge la réception d'au moins 8192 octets de données clé=valeur dans une séquence de négociation. Quand ils proposent ou acceptent des méthodes d'authentification qui exigent explicitement la prise en charge de très longs éléments d'authentification, initiateur et cible DOIVENT prendre en charge la réception d'au moins 64 kilo octets de données clé=valeur.

6.2 Négociation du mode Texte

Durant la connexion, et à partir de là, certains paramètres de session ou de connexion sont déclarés ou négociés par un échange d'informations textuelles.

L'initiateur commence la négociation et/ou déclaration par une demande Text ou Login et indique quand il est prêt pour l'achever (en réglant le bit F à 1 et en le gardant à 1 dans une demande Text, ou le bit T dans la demande Login). Comme le texte de négociation peut s'étendre au delà des limites de PDU, une PDU de demande ou réponse Text ou Login qui a le bit C réglé à 1 NE DOIT PAS avoir le bit F ou T réglé à 1.

Une cible qui reçoit une demande Text ou Login avec le bit C réglé à 1 DOIT répondre par une réponse Text ou Login sans segment de données (DataSegmentLength 0). Un initiateur qui reçoit une réponse Text ou Login avec le bit C réglé à 1 DOIT répondre par une demande Text ou Login sans segment de données (DataSegmentLength 0).

Une cible ou initiateur NE DEVRAIT PAS utiliser une réponse ou demande Text ou Login sans segment de données (DataSegmentLength 0) sauf explicitement exigé par une règle de négociation générale ou spécifique de clé.

Il NE DOIT PAS y avoir plus d'une PDU de demande ou réponse Text en cours sur une connexion iSCSI. Une PDU en cours dans ce contexte est celle qui n'a pas encore été acquittée par le côté iSCSI distant.

Le format d'une déclaration est :

Déclarant-> <clé>=<valeurx>

Le format général d'une négocation text est :

Proposant-> <clé>=<valeurx>

Acceptant-> <clé>={<valeury>|NonCompris|NonPertinent|Rejet}

Donc, une déclaration est un échange textuel unidirectionnel (sauf si la clé n'est pas comprise par le receveur, tandis qu'une négociation est un échange bidirectionnel.

Le proposant ou déclarant peut être l'initiateur ou la cible, et l'acceptant peut être, respectivement, la cible ou l'initiateur. Les cibles ne sont pas limitées à répondre aux paires clé=valeur comme proposé par l'initiateur. La cible peut proposer d'elle-même des paires clé=valeur.

Toutes les négociations sont explicites (c'est-à-dire, le résultat DOIT seulement se fonder sur les valeurs nouvellement échangées ou déclarées). Il n'y a pas de propositions implicites. Si il n'est pas fait de proposition, on ne peut alors pas attendre de réponse. Une conception prudente exige aussi qu'on ne s'appuie pas sur les valeurs par défaut lorsque l'utilisation d'autres valeurs a des conséquences sérieuses.

La valeur proposée ou déclarée peut être une valeur-numérique, une gamme-numérique définie par les valeurs supérieure et inférieure avec les deux entiers séparés par un tilde, une valeur binaire, une valeur textuelle, une valeur-nom-iSCSI, une valeur-nom-local-iSCSI, une valeur booléenne (Oui ou Non), ou une liste de valeurs de texte séparées par des virgules. Une gamme, une grande-valeur-numérique, une valeur-nom-iSCSI, et une valeur-nom-local-iSCSI NE PEUVENT être utilisées que si c'est explicitement permis. Une valeur acceptée peut être une valeur-numérique, une grande-valeur-numérique, une valeur textuelle, ou une valeur booléenne.

Si une clé spécifique n'est pas pertinente pour la négociation en cours, l'acceptant peut répondre avec la constante "NonPertinent" pour tous les types de négociations. Cependant, la négociation n'est pas considérée avoir échoué si la réponse est "NonPertinent". La réponse "NonPertinent" est destinée aux cas dans lesquels plusieurs clés sont présentées par le proposant mais où le choix fait par l'acceptant d'une des clés rend les autres clés non pertinentes. L'exemple qui suit illustre l'utilisation de "NonPertinent" :

I->T InitialR2T=Non,ImmediateData=Oui,FirstBurstLength=4192
T->I InitialR2T=Oui,ImmediateData=Non,FirstBurstLength=NonPertinent
I->T X-rdname-vkey1=(bla,alb,None), X-rdname-vkey2=(bla,alb)
T->I X-rdname-vkey1=None, X-rdname-vkey2=NonPertinent

Toute clé non comprise par l'acceptant peut être ignorée par l'acceptant sans affecter la fonction de base. Cependant, la réponse pour une clé qui n'est pas comprise DOIT être clé=NonCompris. Noter que NonCompris est une réponse valide pour les clés déclaratives aussi bien que négociées. La philosophie générale de iSCSI est que la compréhension précède le traitement pour toute clé iSCSI. Un proposant de clé iSCSI, négociée ou déclarative, dans un échange de clé de texte DOIT donc être capable de traiter correctement une réponse NonCompris.

La bonne façon de traiter une réponse NonCompris dépend de si la clé est spécifiée et si la clé est déclarative ou négociée. Toute mise en œuvre iSCSI DOIT comprendre toutes les clés de texte définies dans le présent document. Retourner une réponse NonCompris à une de ces clés de texte DOIT donc être considéré comme une erreur de protocole et traité en conséquence. Pour toutes les autres clés "ultérieures", c'est-à-dire, les clés de texte définies dans des spécifications ultérieures, une réponse NonCompris conclut la négociation pour une clé négociée, tandis que pour une clé déclarative, une réponse NonCompris informe simplement le déclarant d'un manque de compréhension de la part du receveur.

Dans l'un et l'autre cas, une réponse NonCompris exige toujours que le comportement de protocole associé à cette clé ne soit pas utilisé dans la portée de la clé (connexion/session) par l'un et l'autre côté.

Les constantes "Aucune", "Rejet", "NonPertinent", et "NonCompris" sont réservées et DOIVENT être seulement utilisées comme décrit ici. La violation de cette règle est une erreur de protocole (en particulier, l'utilisation de "Rejet", "NonPertinent", et "NonCompris" comme valeurs proposées).

"Rejet" ou "NonPertinent" sont des options de négociation légitimes quand elles sont permises, mais leur utilisation excessive est déconseillée. Une négociation est considérée achevée quand l'acceptant a envoyé la paire de valeurs de clé même si la valeur est "Rejet", "NonPertinent", ou "NonCompris". Envoyer à nouveau la clé serait une renégociation et est interdit pour de nombreuses clés.

Si l'acceptant envoie "Rejet" comme réponse, la clé négociée est laissée comme valeur actuelle (ou par défaut si aucune valeur n'était établie). Si la valeur courante n'est pas acceptable au proposant sur la connexion ou la session dans laquelle elle est envoyée, le proposant PEUT choisir de terminer la connexion ou session.

Toutes les clés dans ce document DOIVENT être prises en charge par les initiateurs et les cibles iSCSI quand elles sont utilisées comme spécifié ici. Si elles sont utilisées comme spécifié, ces clés NE DOIVENT PAS avoir une réponse NonCompris.

Les mises en œuvre peuvent introduire de nouvelles clés privées en mettant en préfixe X- suivi par leur nom de domaine (inversé) ou avec une nouvelle clé publique enregistrée auprès de l'IANA. Par exemple, l'entité qui possède le domaine exemple.com peut produire :

X-com.exemple.bar.foo.fait_quelque_chose=3

Chaque nouvelle clé publique sur la voie de la normalisation DOIT définir les réponses acceptables à la clé, incluant NonCompris si approprié. À la différence de la [RFC3720], noter que le présent document interdit le préfixe X# pour les nouvelles clés publiques. Sur la base de l'expérience de la mise en œuvre de iSCSI, on sait qu'il n'est plus besoin d'un préfixe de nom standard pour les clés qui permettent une réponse NonCompris. Noter que NonCompris va généralement devoir être permis pour les nouvelles clés publiques pour la rétro compatibilité, ainsi que pour les clés X- privées. Donc, le préfixe de nom "X#" dans les noms de nouvelles clés publique n'a plus aucune signification. Pour éviter la confusion, les noms de nouvelles clés publique NE DOIVENT PAS commencer par un préfixe "X#".

Les développeurs PEUVENT aussi introduire de nouvelles valeurs, mais SEULEMENT pour de nouvelles clés ou méthodes d'authentification (voir la Section 12) ou résumés (voir le paragraphe 13.1).

Chaque fois que des actions ou l'acceptation de paramètres dépendent d'autres paramètres, les règles de dépendance et la séquence de paramètres doivent être spécifiées avec les paramètres.

Dans la phase Login (voir le paragraphe 6.3) chaque étape est une négociation séparée. Dans la phase de pleines caractéristiques, une demande/réponse Text est une négociation. Les négociations DOIVENT être traitées comme des opérations atomiques. Par exemple, toutes les valeurs négociées entrent en effet après que la négociation se conclut par un accord ou sont ignorées si la négociation échoue.

Certains paramètres peuvent être soumis à des règles d'intégrité (par exemple, le paramètre x ne doit pas excéder le paramètre y, ou le paramètre u non 1 implique que le paramètre v soit Oui). Chaque fois qu'exigé, les règles d'intégrité sont spécifiées avec les clés. La vérification de conformité aux règles d'intégrité doit n'être effectuée qu'après que tous les paramètres sont disponibles (l'existant et celui nouvellement négocié). Une cible iSCSI DOIT effectuer la vérification d'intégrité avant l'entrée en vigueur des nouveaux paramètres. Un initiateur PEUT effectuer des vérifications d'intégrité.

Un initiateur ou cible iSCSI PEUT terminer une négociation qui ne se finit pas dans un délai raisonnable, ou un nombre d'échanges spécifique d'une mise en œuvre, mais DEVRAIT permettre au moins six (6) échanges.

6.2.1 Négociations de liste

Dans une négociation de liste, le générateur envoie une liste de valeurs (qui peut inclure "Aucune") dans l'ordre des préférences.

La partie qui répond DOIT le faire avec la même clé et la première valeur qu'elle prend en charge (et qu'il est permis d'utiliser pour le générateur spécifique) choisie dans la liste du générateur.

La constante "Aucune" DOIT toujours être utilisée pour indiquer une fonction manquante. Cependant, "Aucune" n'est un choix valide que si c'est explicitement proposé. Quand "Aucune" est proposé comme élément de choix dans une négociation de clé, cela indique à celui qui répond que ne prendre en charge aucune fonctionnalité relative à cette clé est légal, et si "Aucune" est le résultat de la négociation pour une telle clé, cela signifie que la sémantique spécifique de la clé n'est pas opérationnelle pour la portée de négociation (connexion ou session) de cette clé.

Si un acceptant ne comprend pas une valeur particulière dans une liste, il DOIT l'ignorer. Si un acceptant ne prend pas en charge, ne comprend pas, ou n'a pas la permission d'utiliser toutes les options proposées avec un générateur spécifique, il peut utiliser la constante "Rejet" ou terminer la négociation. Le choix d'une valeur non proposée DOIT être traité par le générateur comme une erreur de protocole.

6.2.2 Négociations de valeur simple

Pour les négociations de valeurs simples, la partie acceptante DOIT répondre avec la même clé. La valeur qu'elle choisit devient le résultat de la négociation.

Proposer une valeur non admissible (par exemple, pas dans les limites spécifiées) PEUT avoir pour réponse la constante "Rejet" ; autrement, l'acceptant DOIT choisir une valeur admissible.

Le choix par l'acceptant d'une valeur non admissible selon les règles de choix est considéré comme erreur de protocole. Les règles de choix sont spécifiques de la clé.

Pour une gamme numérique, la valeur choisie DOIT être un entier dans la gamme proposée ou "Rejet" (si la gamme n'est pas acceptable).

Pour les négociations booléennes (c'est-à-dire, les clés qui prennent des valeurs "Oui" ou "Non") la partie acceptante DOIT répondre avec la même clé et le résultat de la négociation quand la valeur reçue ne détermine pas ce résultat par elle-même. La dernière valeur transmise devient le résultat de la négociation. Les règles de choix de la valeur avec laquelle répondre sont exprimées comme des fonctions booléennes de la valeur reçue, et la valeur que la partie acceptante aurait choisi si elle avait eu le choix.

Spécifiquement, les deux cas dans lesquels les réponses sont FACULTATIVES sont :
- la fonction booléene est "ET" et la valeur "Non" est reçue. Le résultat de la négociation est "Non" ;
- la fonction booléenne est "OU" et la valeur "Oui" est reçue. Le résultat de la négociation est "Oui".

Des réponses sont EXIGÉES dans tous les autres cas, et la valeur choisie et envoyée par l'acceptant devient le résultat de la négociation.

6.3 Phase Login

La phase Login établit une connexion iSCSI entre un initiateur et une cible ; elle crée aussi une nouvelle session ou associe la connexion à une session existante. La phase Login établit les paramètres de protocole iSCSI et les paramètres de sécurité, et authentifie l'initiateur et la cible l'un à l'autre.

La phase Login n'est mise en œuvre que via des demandes et réponses Login. Toute la phase Login est considérée comme une seule tâche et a une seule étiquette de tâche d'initiateur (similaire aux commandes SCSI reliées).

Il NE DOIT PAS y avoir plus d'une demande ou réponse Login en cours sur une connexion iSCSI. Une PDU en cours dans ce contexte est celle qui n'a pas encore été acquittée par le côté iSCSI distant.

La MaxRecvDataSegmentLength *(longueur maximale de segment de données reçu)* par défaut est utilisée durant l'établissement de connexion *(login)*.

La séquence de demandes et réponses de la phase Login se déroule comme suit :
- demande Login initiale,
- réponse Login partielle (facultatif),
- demandes et réponses Login supplémentaires (facultatif),
- réponse Login finale (obligatoire),

La demande Login initiale de toute connexion DOIT inclure la paire clé=valeur InitiatorName *(nom de l'initiateur)*. La demande Login initiale de la première connexion d'une session PEUT aussi inclure la paire clé=valeur SessionType *(type de session)*. Pour toute connexion au sein d'une session dont le type n'est pas "Découverte", la première demande Login DOIT aussi inclure la paire clé=valeur TargetName *(nom de cible)*.

La réponse Login finale accepte ou rejette la demande Login.

La phase Login PEUT inclure une étape de négociation de sécurité *(SecurityNegotiation)* et une étape de négociation Login opérationnelle *(LoginOperationalNegotiation)* et DOIT inclure au moins une d'elles, mais l'étape incluse PEUT être vide sauf pour les noms obligatoires.

Les demandes et réponses Login contiennent un champ (CSG) qui indique l'étape de négociation actuelle (SecurityNegotiation ou LoginOperationalNegotiation). Si les deux étapes sont utilisées, l'étape SecurityNegotiation DOIT précéder l'étape LoginOperationalNegotiation.

Certains paramètres de fonctionnement peuvent être négociés en dehors du Login par des demandes et réponses Text.

Les clés de sécurité relatives à l'authentification (Section 12) DOIVENT être complètement négociées au sein de la phase Login. L'utilisation de la sécurité IPsec sous-jacente est spécifiée au paragraphe 9.3, dans la [RFC3723], et dans la [RFC7146]. La prise en charge par iSCSI de la sécurité au sein du protocole consiste seulement en l'authentification dans la phase Login.

Dans certains environnements, une cible ou un initiateur n'est pas intéressé à authentifier son partenaire. Il est possible de sauter l'authentification dans la demande et réponse Login.

L'initiateur et la cible PEUVENT vouloir négocier les paramètres d'authentification iSCSI. Une fois cette négociation achevée, le canal est considéré comme sûr.

La plupart des clés de négociation ne sont permises que dans une étape spécifique. Les clés utilisées durant l'étape SecurityNegotiation sont données à la Section 12, et les clés utilisées durant l'étape LoginOperationalNegotiation sont discutées à la Section 13. Seul un ensemble limité de clés (marqué comme Toute-étape à la Section 13) peut être utilisé dans l'une ou l'autre des deux étapes.

Toute demande ou réponse Login appartient à une étape spécifique ; cela détermine les clés de négociation permises avec la demande ou réponse. L'envoi d'une clé qui n'est pas permise dans l'étape en cours est une erreur de protocole.

La transition entre étapes est effectuée par un échange de commandes (demande/réponse) qui porte le bit T et le même code de CSG. Durant cet échange, la prochaine étape est choisie par la cible via le code de prochaine étape (NSG, *Next Stage code*). Le NSG choisi NE DOIT PAS excéder la valeur déclarée par l'initiateur. L'initiateur peut demander une transition chaque fois qu'il est prêt, mais une cible ne peut répondre avec une transition qu'après qu'une est proposée par l'initiateur.

Dans une séquence de négociation, les réglages du bit T dans une paire de demande réponse Login n'ont pas d'impact sur les réglages du bit T dans la paire suivante. Un initiateur qui a le bit T réglé à 1 dans une paire et a une réponse avec un réglage du bit T à 0 peut produire la demande suivante avec le bit T réglé à 0.

Quand une transition est demandée par l'initiateur et acquittée par la cible, l'initiateur et la cible passent tous deux à l'étape choisie.

Les cibles NE DOIVENT PAS soumettre des paramètres qui exigent une demande Login supplémentaire à l'initiateur avec une réponse dont le bit T est réglé à 1.

Les transitions d'étape durant l'établissement de connexion (incluant l'entrée et la sortie) ne sont possibles que comme montré dans le tableau suivant :

De Vers → | \V	Sécurité	Opérationnel	Pleines caractéristiques
(début)	oui	oui	non
Sécurité	non	oui	oui|
Opérationnel	non	non	oui

La réponse finale Login qui accepte une demande Login peut seulement venir comme une réponse à une demande Login avec le bit T réglé à 1, et la demande et la réponse DOIVENT toutes deux indiquer FullFeaturePhase comme prochaine phase via le champ NSG.

Ni l'initiateur ni la cible ne devraient tenter de déclarer ou négocier un paramètre plus d'une fois durant l'établissement de connexion, excepté pour des réponses pour des clés spécifiques qui permettent explicitement des déclarations répétées de clés (par exemple, TargetAddress). Une tentative de renégocier/redéclarer des paramètres non spécifiquement permis DOIT être détectée par l'initiateur et la cible. Si une telle tentative est détectée par la cible, la cible DOIT répondre par un rejet de connexion (erreur de l'initiateur) ; si elle est détectée par l'initiateur, l'initiateur DOIT abandonner la connexion.
|

6.3.1 Début de la phase Login

La phase commence par une demande Login de l'initiateur à la cible. La demande initiale Login inclut :
- la version de protocole prise en charge par l'initiateur,
- le nom de l'initiateur iSCSI et le nom de cible iSCSI,
- l'ISID, le TSIH, et les identifiants de connexion,
- l'étape de négociation dans laquelle l'initiateur est prêt à entrer.

Un Login peut créer une nouvelle session, ou il peut ajouter une connexion à une session existante. Entre un nœud initiateur iSCSI (choisi seulement par un InitiatorName) et une certaine cible iSCSI définie par un TargetName iSCSI et une étiquette de groupe de portail cible, les résultats du Login sont définis par le tableau suivant :

ISID	**TSIH**	**CID**	**Action de la cible**
nouveau	non zéro	tout	Échec de connexion ("la session n'existe pas")
nouveau	zéro	tout	Instancie une nouvelle session
existant	zéro	tout	Fait une réinstallation de session (paragraphe 6.3.5)
existant	non zéro existant	nouveau existant	Ajoute une nouvelle connexion à la session
existant	non zéro existant	existant	Fait une réinstallation de session (paragraphe 7.1.4.3)
existant	non zéro nouveau	tout	Échec de connexion ("la session n'existe pas")

La détermination de "existant" ou "nouveau" est faite par la cible.

Facultativement, la demande Login peut inclure :
- des paramètres de sécurité OU
- des paramètres de fonctionnement iSCSI ET/OU
- la prochaine étape de négociation dans laquelle l'initiateur est prêt à entrer.

La cible peut répondre au Login d'une des façons suivantes :

- réponse Login avec rejet de Login. C'est un rejet immédiat de la cible qui cause la terminaison de la connexion et la session se termine si c'est la première (ou la seule) connexion d'une nouvelle session. Le bit T, le champ CSG, et le champ NSG sont réservés.

- réponse Login avec connexion acceptée comme réponse finale (bit T réglé à 1 et le NSG dans la demande et la réponse est réglé à FullFeaturePhase). La réponse inclut la version de protocole prise en charge par la cible et l'identifiant de session, et peut inclure des paramètres iSCSI de fonctionnement ou de sécurité (cela dépend de l'étape en cours).

- réponse Login avec connexion acceptée comme réponse partielle (NSG non réglé à FullFeaturePhase dans la demande et la réponse) qui indique le début d'une séquence de négociation. La réponse inclut la version de protocole acceptée par la cible et les paramètres de sécurité ou iSCSI (quand aucun mécanisme de sécurité n'est choisi) supportés par la cible.

Si l'initiateur décide de sauter l'étape SecurityNegotiation, il produit le Login avec le CSG réglé à LoginOperationalNegotiation, et la cible peut répondre avec une réponse Login qui indique qu'elle ne veut pas accepter la connexion (paragraphe 11.13) sans négociation de sécurité et va terminer la connexion avec une réponse de “échec d'authentification” (voir le paragraphe 11.13.5).

Si l'initiateur veut négocier la sécurité iSCSI, mais ne veut pas faire la proposition initiale de paramètres et peut accepter une connexion sans sécurité iSCSI, il produit le Login avec le bit T réglé à 1, le CSG réglé à SecurityNegotiation, et le NSG réglé à LoginOperationalNegotiation. Si la cible est aussi prête à sauter la sécurité, la réponse Login contient seulement la clé TargetPortalGroupTag (voir le paragraphe 13.9) le bit T réglé à 1, le CSG réglé à SecurityNegotiation, et le NSG réglé à LoginOperationalNegotiation.

Un initiateur qui choisit de fonctionner sans sécurité iSCSI et avec tous les paramètres de fonctionnement qui prennent les valeurs par défaut, produit le Login avec le bit T réglé à 1, le CSG réglé à LoginOperationalNegotiation, et le NSG réglé à FullFeaturePhase. Si la cible est aussi prête à sauter la sécurité et peut finir sa LoginOperationalNegotiation, la réponse de Login a le bit T réglé à 1, le CSG réglé à LoginOperationalNegotiation, et le NSG réglé à FullFeaturePhase dans l'étape suivante.

Durant la phase Login, la cible iSCSI DOIT retourner la clé TargetPortalGroupTag avec la première PDU de réponse de Login avec laquelle elle est autorisée à le faire (c'est-à-dire, la première réponse de Login produite après la première demande Login avec le bit C réglé à 0) pour tous les types de session. La valeur de clé TargetPortalGroupTag indique le groupe de portail iSCSI qui dessert la PDU de demande Login. Si la reconfiguration des groupes de portails iSCSI est un problème dans un certain environnement, l'initiateur iSCSI devrait utiliser cette clé pour s'assurer qu'il a bien initié la phase Login avec le groupe portail cible voulu.

6.3.2 Négociation de la sécurité iSCSI

L'échange de sécurité établit le mécanisme de sécurité et authentifie l'initiateur et la cible l'un à l'autre. L'échange se fait selon la méthode d'authentification choisie dans la phase de négociation et est conduite en utilisant les paramètres clé=valeur portés dans les demandes et réponses Login.

Une négociation dirigée par l'initiateur procède comme suit :

- L'initiateur envoie une demande Login avec une liste ordonnée des options qu'il prend en charge (algorithme d'authentification). La liste des options est dans l'ordre de préférence de l'initiateur. L'initiateur PEUT aussi envoyer des options d'extension privées ou publiques.

- La cible DOIT répondre avec la première option de la liste qu'il prend en charge et qu'il lui est permis d'utiliser pour cet initiateur, sauf si il n'en prend aucune en charge, auquel cas il DOIT répondre par "Rejet" (paragraphe 6.2). Les paramètres sont codés en UTF-8 comme clé=valeur. Pour les paramètres de sécurité, voir la Section 12.

- Quand l'initiateur se considère prêt à conclure l'étape SecurityNegotiation, il règle le bit T à 1 et le NSG auquel il voudrait que soit la prochaine étape. La cible va alors régler le bit T à 1 et régler le NSG à la prochaine étape dans la réponse Login quand il finit d'envoyer ses clés de sécurité. La prochaine étape choisie sera celle que la cible a choisie. Si la prochaine étape est FullFeaturePhase, la cible DOIT répondre avec une réponse Login de valeur TSIH.

Si la négociation de sécurité échoue à la cible, celle-ci DOIT envoyer la PDU de réponse Login appropriée. Si la négociation de sécurité échoue chez l'initiateur, l'initiateur DEVRAIT clore la connexion.

On devrait noter que la négociation peut aussi être dirigée par la cible si l'initiateur ne prend pas en charge la sécurité mais n'est pas prêt à diriger la négociation (proposer les options) ; voir un exemple à l'Appendice B.

6.3.3 Négociation des paramètres de fonctionnement durant la phase Login

La négociation des paramètres de fonctionnement durant la phase Login PEUT être faite :
- en commençant par la première demande Login si l'initiateur ne propose pas d'option de sécurité/intégrité.
- en commençant immédiatement après la négociation de sécurité si l'initiateur et la cible effectuent une telle négociation.

La négociation des paramètres de fonctionnement PEUT impliquer plusieurs échanges de demande/réponse Login commencés et terminés par l'initiateur. L'initiateur DOIT indiquer son intention de terminer la négociation en réglant le bit T à 1 ; la cible règle le bit T à 1 sur la dernière réponse.

Même quand l'initiateur indique son intention de changer d'étape en réglant le bit T à 1 dans une demande Login, la cible PEUT répondre avec une réponse Login où le bit T est réglé à 0. Dans ce cas, l'initiateur DEVRAIT continuer de régler le bit T à 1 dans les demandes Login suivantes (même les demandes vides) qu'il envoie, jusqu'à ce que la cible envoie une réponse Login avec le bit T réglé à 1 ou envoie une clé qui exige que l'initiateur règle le bit T à 0.

Certains paramètres spécifiques de session ne peuvent être spécifiés que durant la phase Login de la première connexion d'une session (c'est-à-dire, commencée par une demande Login qui contient un TSIH de valeur zéro) – la phase Login du début (par exemple, le nombre maximum de connexions qui peuvent être utilisées pour cette session).

Une session est opérationnelle une fois qu'elle a au moins une connexion dans la phase Pleines caractéristiques *(Full Feature)*. Les connexions nouvelles ou de remplacement ne peuvent être ajoutées à une session qu'après que la session est opérationnelle.

Pour les paramètres de fonctionnement, voir la Section 13.

6.3.4 Réinstallation de connexion

La réinstallation de connexion est le processus d'un initiateur qui établit sa connexion avec une combinaison ISID-TSIH-CID qui est éventuellement active du point de vue de la cible, ce qui cause la fermeture implicite de la connexion correspondant au CID et à la réinstallation d'une nouvelle connexion iSCSI de phase Pleines caractéristiques à sa place (avec le même CID). Donc, le TSIH dans la PDU de demande Login DOIT être non zéro, et le CID ne change pas durant une réinstallation de connexion. La demande Login effectue la fonction de fermeture de la vieille connexion si une fermeture explicite n'a pas été effectuée antérieurement. Dans les sessions avec une seule connexion, cela peut impliquer l'ouverture d'une seconde connexion dans le seul but de nettoyer la première. Les cibles DOIVENT prendre en charge l'ouverture d'une seconde connexion même quand elles ne prennent pas en charge les connexions multiples dans la phase Pleines caractéristiques si Niveau de récupération d'erreur *(ErrorRecoveryLevel)* est 2 et DEVRAIENT prendre en charge l'ouverture d'une seconde connexion si Niveau de récupération d'erreur est inférieur à 2.

Si le niveau de récupération d'erreur en cours est 2, la réinstallation de connexion permet une future réallocation des tâches. Si le niveau de récupération d'erreur en cours est inférieur à 2, la réinstallation de connexion est le remplacement du vieux CID sans permettre de réallocation des tâches. Dans ce cas, toutes les tâches qui étaient actives sur le vieux CID doivent être immédiatement terminées sans autre mention à l'initiateur.

L'état de la connexion de l'initiateur DOIT être CLEANUP_WAIT (paragraphe 8.1.3) quand l'initiateur tente une réinstallation de connexion.

En pratique, en plus de la fermeture implicite de l'ancienne connexion, la réinstallation est équivalente à l'établissement d'une nouvelle connexion.

6.3.5 Réinstallation, clôture, et fin de temporisation de session

La réinstallation de session est le processus d'un initiateur qui établit la connexion avec un ISID qui est éventuellement actif du point de vue de la cible pour cet initiateur, fermant donc implicitement la session qui correspond à l'ISID et réinstallant une nouvelle session iSCSI à sa place (avec le même ISID). Donc, le TSIH dans la PDU Login DOIT être zéro pour signaler la réinstallation de session. La réinstallation de session cause la fermeture immédiate par la cible de toutes les tâches qui étaient actives sur la vieille session sans autre mention à l'initiateur.

Létat de session de l''initiateur DOIT être FAILED (paragraphe 8.3) quand l'initiateur tente une réinstallation de session.

La clôture de session est un événement défini comme un des suivants :
- une fermeture de session réussie,
- une fermeture de connexion réussie pour la dernière connexion en phase Pleines caractéristiques quand aucune autre connexion dans la session n'est en attente de nettoyage (paragraphe 8.2) et aucune tâche de la session n'attend de réallocation.

La fin de temporisation de session est un événement défini comme survenant lorsque la temporisation de l'état de la dernière connexion arrive à expiration et qu'aucune tâche n'attend de réallocation. Cela amène la session à l'état FREE *(libre)* (voir les diagrammes d'état de session au paragraphe 8.3).

6.3.5.1 Perte de notifications de nexus

La couche iSCSI fournit à la couche SCSI la notification "perte de nexus I_T" lorsque un des événements suivants se produit :
- achèvement réussi d'une réinstallation de session,
- événement de cloture de session,
- événement de fin de temporisation de session.

Certaines actions d'élimination d'objet SCSI peuvent résulter du fait de la notification dans les nœuds d'extrémité SCSI, comme documenté à l'Appendice E.

6.3.6 Continuation et échec de session

La continuation de session est le procès par lequel l'état d'une session préexistante continue d'être utilisé par la réinstallation de connexion (paragraphe 6.3.4) ou par l'ajout d'une connexion avec un nouveau CID. L'une et l'autre de ces actions associe la nouvelle connexion de transport à l'état de session.

L'échec de session est un événement où la dernière connexion en phase de pleines caractéristiques atteint l'état CLEANUP_WAIT (paragraphe 8.2) ou achève avec succès une fermeture de connexion de récupération, causant donc le début d'attente par toutes les tâches actives (qui obéissaient antérieurement à la connexion) d'une réallocation de tâche.

6.4 Négociation des paramètres de fonctionnement en dehors de la phase Login

Certains paramètres de fonctionnement PEUVENT être négociés en dehors de (après) la phase Login.

La négociation de paramètres dans la phase de pleines caractéristiques est faite par des demandes et réponses Text. La négociation des paramètres de fonctionnement PEUT impliquer plusieurs échanges de demandes-réponses Text, dont chacun va utiliser la même étiquette de tâche d'initiateur ; l'initiateur commence et termine toujours chacun de ces échanges. L'initiateur DOIT indiquer son intention de finir la négociation en réglant le bit F à 1 ; la cible règle le bit F à 1 sur la dernière réponse.

Si la cible répond à une demande Text avec le bit F réglé à 1 avec une réponse Text dont le bit F est réglé à 0, l'initiateur devraient continuer d'envoyer la demande Text (même des demandes vides) avec le bit F réglé à 1 bien qu'il veuille toujours finir la négociation, jusqu'à ce qu'il reçoive la réponse Text avec le bit F réglé à 1. Répondre à une demande Text avec le bit F réglé à 1 avec une réponse vide (pas de paire clé=valeur) avec le bit F réglé à 0 est déconseillé.

Même quand l'initiateur indique son intention de finir la négociation en réglant le bit F à 1 dans une demande Text, la cible PEUT répondre avec une réponse Text dont le bit F est réglé à 0. Dans ce cas, l'initiateur DEVRAIT continuer de régler le bit F à 1 dans les demandes Text suivantes (même des demandes vides) qu'il envoie, jusqu'à ce que la cible envoie la réponse Text finale avec le bit F réglé à 1. Noter que dans le même cas d'une demande Text avec le bit F réglé à 1, la cible NE DEVRAIT PAS répondre avec une réponse Text vide (pas de paire clé=valeur) avec le bit F réglé à 0, parce qu'une telle réponse peut causer l'abandon de la négociation par l'initiateur.

Les cibles NE DOIVENT PAS soumettre des paramètres qui exigent une demande Text supplémentaire de l'initiateur dans une réponse Text avec le bit F réglé à 1.

Dans une séquence de négociation, les réglages du bit F dans une paire de demande-réponse Text n'ont pas de conséquence sur les réglages du bit F dans la paire suivante. Un initiateur qui a le bit F réglé à 1 dans une demande à qui il est répondu par un réglage du bit F de 0 peut produire la prochaine demande avec le bit F réglé à 0.

Chaque fois que la cible répond avec le bit F réglé à 0, elle DOIT régler l'étiquette de transfert de cible à une valeur autre que la valeur par défaut de 0xffffffff.

Un initiateur PEUT réinitialiser la négociation d'un paramètre de fonctionnement en produisant une demande Text avec l'étiquette de tranfert de cible réglée à la valeur de 0xffffffff après réception d'une réponse dont l'étiquette de tranfert de cible est réglée à une valeur autre que 0xffffffff. Une cible peut réinitiliser une négociation de paramètre de fonctionnement en répondant à une demande Text par une PDU Rejet.

Ni l'initiateur ni la cible ne devraient tenter de déclarer ou négocier un paramètre plus d'une fois durant une séquence de négociation, sauf pour des réponses à des clés spécifiques qui permettent explicitement des déclarations de clés répétées (par exemple, TargetAddress). Si une telle tentative est détectée par la cible, la cible DOIT répondre avec une PDU Rejet avec une cause de "Erreur de protocole". L'initiateur DOIT réinitialiser la négociation comme expliqué ci-dessus.

Les paramètres négociés par une séquence de négociation d'échange de texte ne deviennent effectifs qu'après la fin de la séquence de négociation.

7. Traitement et récupération d'erreur iSCSI

7.1 Vue d'ensemble

7.1.1 Fondements

Les deux considérations suivantes ont guidé la conception d'une grande partie de la fonction de récupération d'erreur dans iSCSI :

- Une PDU iSCSI peut échouer à la vérification de résumé et être abandonnée, en dépit de sa réception par la couche TCP. La couche iSCSI doit facultativement être admise à récupérer de telles PDU abandonnées.
- Une connexion TCP peut échouer à tout moment durant le transfert de données. Toutes les tâches actives doivent facultativement être admises à continuer sur une connexion TCP différente au sein de la même session.

Les mises en œuvre ont une souplesse considérable pour décider quel degré de récupération d'erreur prendre en charge, quand l'utiliser, et par quels mécanismes réaliser le comportement requis. Seules les actions visibles en externe des mécanismes de récupération d'erreur doivent être normalisées pour assurer l'interopérabilité.

Cette section décrit un modèle général pour la récupération en vue de l'interopérabilité. Voir à l'Appendice D les détails de la façon dont le modèle décrit peut être mis en œuvre. Les mises en œuvre conformes n'ont pas à suivre les détails de mise en œuvre de ce modèle tel que présenté, mais le comportement externe de telles mises en œuvre doit correspondre aux caractèristiques externes observables du modèle présenté.

7.1.2 Objectifs

Les objectifs majeurs de la conception du schéma de récupération d'erreur de iSCSI sont les suivants :

- Permettre aux mises en œuvre iSCSI de satisfaire les différentes exigences en définissant une collection de mécanismes de récupération d'erreur parmi lesquels les mises en œuvre peuvent choisir.
- Assurer l'interopérabilité entre deux mises en œuvre qui prennent en charge des ensembles différents de capacités de récupération d'erreur.
- Définir les mécanismes de récupération d'erreur pour assurer l'ordre des commandes même en présence d'erreurs, pour les initiateurs qui demandent cet ordre.
- Ne pas faire d'ajouts dans le chemin rapide, mais permettre une complexité modérée dans le chemin de récupération d'erreur.
- Empêcher l'initiateur et la cible de tenter de récupérer le même ensemble de PDU au même moment. Par exemple, il doit y avoir une claire "distribution de fonctions de récupération d'erreur" entre l'initiateur et la cible.

7.1.3 Caractéristiques de protocole et attentes d'état

Les mécanismes d'initiateur définis dans une connexion avec récupération d'erreur sont :

a) NOP-Out pour vérifier les numéros de séquence de la cible (paragraphe 11.18)
b) Réessai de commande (paragraphe 7.2.1)
c) Prise en charge de la récupération R2T (paragraphe 7.8)
d) Demande de retransmission d'état/données/R2T en utilisant la facilité SNACK (paragraphe 11.16)
e) Accusé de réception des données (paragraphe 11.16)
f) Réallocation de l'allégeance de connexion d'une tâche à une connexion TCP différente (paragraphe 7.2.2)

g) Terminaison de la session iSCSI entière pour en commencer une nouvelle (paragraphe 7.1.4.4)

Les mécanismes de cible définis dans une connexion avec récupération d'erreur sont :
a) NOP-In pour vérifier les numéros de séquence de l'initiateur (paragraphe 11.19)
b) Demande de retransmission des données en utilisant la caractéristique R2T de récupération (paragraphe 7.8)
c) Prise en charge de SNACK (paragraphe 11.16)
d) Demander que des parties des données lues soient acquittées (paragraphe 11.7.2)
e) Prise en charge de la réallocation d'allégeance (paragraphe 7.2.2)
f) Terminaison de la session iSCSI entière pour forcer l'initiateur à recommencer (paragraphe 7.1.4.4)

Pour toute commande SCSI en cours, on suppose que iSCSI, en conjonction avec le SCSI chez l'initiateur, est capable de conserver assez d'informations pour être capable de reconstruire la PDU de commande et que les données sortantes sont disponibles (dans la mémoire de l'hôte) pour la retransmission pendant que la commande est en cours. On suppose aussi qu'à la cible, les données entrantes (données lues) PEUVENT être conservées en vue de la récupération, ou qu'elle peuvent être lues à nouveau à partir d'un appareil serveur.

On suppose de plus qu'une cible va conserver "l'état et le sens" pour une commande qu'elle a exécutée si elle supporte la retransmission d'état.

Une cible qui est d'accord pour prendre en charge la retransmission des données est supposée être prête à retransmettre les données sortantes (c'est-à-dire, Data-In) sur demande jusqu'à ce que l'état pour la commande achevée soit acquitté ou que les données en question aient été acquittées séparément.

7.1.4 Classes de récupération

iSCSI permet les classes de récupération suivantes (en ordre de portée croissante des tâches iSCSI affectéés) :
- au sein d'une commande (c'est-à-dire, sans exiger de redémarrage de la commande)
- au sein d'une connexion (c'est-à-dire, sans exiger que la connexion soit reconstruite, mais peut-être en exigeant le redémarage de la commande)
- récupération de connexion (c'est-à-dire, peut-être en exigeant que les connexions soient reconstruites et les commandes répétées)
- récupération de session

Les scénarios de récupération décrits dans la suite de cette section sont représentatives plutôt qu'exclusives. Dans chaque cas, ils détaillent la plus basse classe de récupération qui PEUT être tentée. Il appartient à la mise en œuvre de décider dans quelles circonstances passer à la prochaine classe de récupération et/ou quelles classes de récupération mettre en œuvre. La cible et l'initiateur iSCSI PEUVENT tous deux rehausser le traitement d'erreur à une classe de récupération d'erreur qui impacte un plus grand nombre de tâches iSCSI dans tous les cas identifiés dans la discussion qui suit.

Dans toutes les classes, la mise en œuvre a le choix de renvoyer les erreurs à l'initiateur SCSI (avec un code de réponse approprié) et dans ce cas, la tâche, si il en est, doit être retirée de la cible et tous les effets colatéraux, comme ACA, doivent être pris en compte.

L'utilisation des classes de récupération au sein de la connexion et au sein de la commande NE DOIT PAS être tentée avant que la connexion soit dans la phase de pleines caractéristiques.

Dans la description détaillée des classes de récupération, les termes d'obligation (DOIT, DEVRAIT, PEUT, etc.) indiquent des actions normatives à exécuter si la classe de récupération est prise en charge (voir le paragraphe 7.1.5 pour la sémantique de négociation qui s'y rapporte) et utilisée.

7.1.4.1 Récupération au sein de la commande

À la cible, les cas suivants se prêtent à la récupération au sein de la commande :

PDU de données perdues – réalisée par une des actions suivantes :
a) Erreur de résumé de données – traité comme spécifié au paragraphe 7.8, en utilisant l'option d'un R2T de récupération
b) Fin de temporisation de réception de séquence (pas de donnée ou données partielles et pas de bit F) – considéré comme une erreur de séquence implicite et traité comme spécifié au paragraphe 7.9, en utilisant l'option d'un R2T de récupération
c) Erreur de résumé d'en-tête, qui se manifeste comme une fin de temporisation de réception de séquence ou une erreur de séquence – traitée comme spécifié au paragraphe 7.9, en utilisant l'option d'un R2T de récupération.

Chez l'initiateur, les cas suivants se prêtent à la récupération au sein de la commande :

PDU de données perdue ou R2T perdu – réalisé par une des actions suivantes :
a) Erreur de résumé de données - traitée comme spécifié au paragraphe 7.8, en utilisant l'option d'un SNACK,
b) Fin de temporisation de réception de séquence (pas d'état) ou fin de temporisation de réception de réponse – traitée comme spécifié au paragraphe 7.9, en utilisant l'option d'un SNACK,
c) Erreur de résumé d'en-tête, qui se manifeste comme une fin de temporisation de réception de séquence ou d'une erreur de séquence – traitée comme spécifié au paragraphe 7.9, en utilisant l'option d'un SNACK.

Pour éviter une compétition avec la cible, qui peut déjà avoir un R2T de récupération ou une réponse de terminaison en cours, un initiateur NE DEVRAIT PAS générer un SNACK pour un R2T sur la base de ses fins de temporisations internes (s'il en est). La récupération dans ce cas est plutôt laissée à la cible.

Les valeurs de temporisation utilisées par l'initiateur et la cible sortent du domaine d'application du présent document. Une fin de temporisation de réception de séquence est généralement une valeur assez grande pour permettre d'achever le transfert de la séquence de données.

7.1.4.2 Récupération au sein de la connexion

Chez l'initiateur, les cas suivants se prêtent à la récupération au sein de la connexion :

a) Demandes non acquittées pendant longtemps. Les demandes ont un accusé de réception explicite avec le numéro de séquence de commande attendu ou implicite en recevant des données et/ou un état. L'initiateur PEUT réessayer des commandes non acquittées comme spécifié au paragraphe 7.2.

b) Réponse iSCSI numérotée perdue. C'est reconnu soit en identifiant une erreur de résumé de données sur une PDU de réponse ou une PDU Data-In portant l'état, soit en recevant une PDU de réponse avec un numéro de séquence d'état supérieur à celui attendu. Dans le premier cas, le traitement d'erreur de résumé est fait comme spécifié au paragraphe 7.8, en utilisant l'option d'un SNACK. Dans le second cas, le traitement d'erreur de séquence est fait comme spécifié au paragraphe 7.9, en utilisant l'option d'un SNACK.

Chez la cible, le cas suivant se prête à la récupération au sein de la connexion :
- État/réponse non acquitté pendant longtemps. La cible PEUT produire un NOP-In (avec une étiquette de transfert de cible valide ou autrement) qui porte le numéro de séquence du prochain état qu'elle va utiliser dans le champ Numéro de séquence d'état. Cela aide l'initiateur à détecter tous les numéros de séquence d'état manquants et produit un SNACK pour l'état.

Les valeurs de fin de temporisation utilisées par l'initiateur et la cible sortent du domaine d'application de ce document.

7.1.4.3 Récupération de connexion

Chez un initiateur iSCSI, les cas suivants se prêtent à la récupération de connexion :

a) Défaillance de connexion TCP : l'initiateur DOIT clore la connexion. Il DOIT ensuite désétablir implicitement ou explicitement la connexion défaillante avec le code de cause "supprimer la connexion pour récupération" et réallouer l'allégeance de connexion pour toutes les commandes encore en cours associées à la connexion défaillante sur une ou plusieurs connexions (dont certaines ou toutes PEUVENT être des connexions nouvellement établies) en utilisant la fonction de gestion des tâches "TASK REASSIGN" (voir le paragraphe 11.5.1). Pour un initiateur, une commande est en cours tant qu'il n'a pas reçu de réponse ou une PDU Data-In incluant l'état.

Note : La fonction logout est obligatoire. Cependant, un nouvel établissement de connexion n'est obligatoire que si la connexion défaillante était la dernière ou la seule connexion dans la session.

b) Réception d'un message asynchrone qui indique qu'une connexion ou toutes celles d'une session ont été abandonnées. L'initiateur DOIT traiter cela comme une défaillance de connexion TCP pour la ou les connexions visées dans le message.

Chez une cible iSCSI, le cas suivant se prête à une récupération de connexion :
- Défaillance de connexion TCP : la cible DOIT clore la connexion et, si plus d'une connexion est disponible, la cible DEVRAIT envoyer un message asynchrone qui indique qu'il a abandonné la connexion. Ensuite, la cible va attendre que l'initiateur continue la récupération.

7.1.4.4 Récupération de session

La récupération de session devrait être effectuée lorsque toutes les autres tentatives de récupération ont échoué. Des initiateurs et cibles très simples PEUVENT effectuer la récupération de session sur toutes les erreurs iSCSI et s'appuyer sur la récupération à la couche SCSI et au dessus.

La récupération de session implique de clore toutes les connexions TCP, interrompant en interne l'exécution et la mise en file d'attente de toutes les tâches pour l'initiateur donné à la cible, terminant les commandes SCSI en cours avec une réponse de service SCSI appropriée chez l'initiateur, et redémarrant une session sur un nouvel ensemble de connexions (établissement de connexion TCP et login sur toutes les nouvelles connexions).

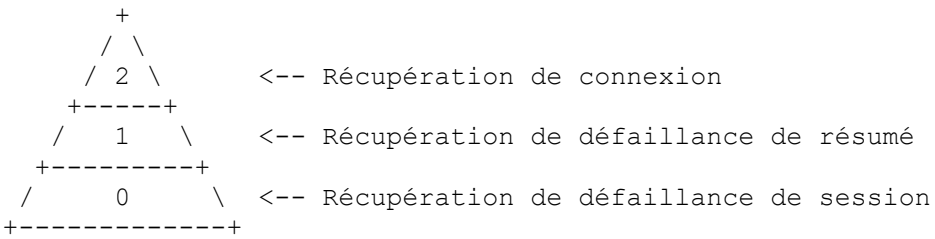
Pour les effets possibles de l'apurement de la récupération de session sur les objets SCSI et iSCSI, se référer à l'Appendice E.

7.1.5 Hiérarchie de récupération des erreurs

Les classes de récupération d'erreur décrites jusqu'à présent sont organisées dans une hiérarchie pour faciliter la compréhension et limiter la complexité de la mise en œuvre. Avec quelques niveaux de récupération bien définis, l'interopérabilité est plus facile à réaliser. Les attributs de cette hiérarchie sont les suivants :

a) chaque niveau est un sur-ensemble des capacités du niveau précédent. Par exemple, la prise en charge du niveau 1 implique la prise en charge de toutes les capacités du niveau 0 et plus.
b) Comme corollaire, la prise en charge d'un niveau de récupération d'erreur supérieur signifie une sophistication accrue et éventuellement une augmentation des exigences de ressources.
c) La prise en charge du niveau "n" de récupération d'erreur est annoncée et négociée par chaque entité iSCSI par l'échange de la clé text "Niveau de récupération d'erreur=n". La plus basse des deux valeurs échangées est le niveau de récupération d'erreur opérationnel pour la session.

Le diagramme suivant représente la hiérarchie de récupération d'erreur .



Le tableau suivant fait la liste des capacités de récupération d'erreur (ER, *error recovery*) attendues des mises en œuvre qui prennent en charge chaque niveau de récupération d'erreur.

Niveau de récupération d'erreur	Capacités de récupération d'erreur associées
0	Classe de récupération de session (récupération de session)
1	Récupération de défaillance de résumé (voir la note) plus les capacités du niveau d'ER 0
2	Classe de récupération de connexion (récupération de connexion) plus les capacitées de ER niveau 1

Note : La récupération de défaillance de résumé se compose de deux classes de récupération : la classe de récupération au sein de la connexion, et la classe de récupération au sein de la commande.

Quand une valeur définie de niveau de récupération d'erreur est proposée par un générateur dans une négociation text, le générateur DOIT prendre en charge la fonctionnalité définie pour la valeur proposée et, de plus, la fonctionnalité correspondant à toute valeur définie numériquement inférieure à la valeur proposée. Quand une valeur définie de niveau de récupération d'erreur est retournée par un répondant d'une négociation text, le répondant DOIT prendre en charge la fonctionnalité correspondant au niveau de récupération d'erreur qu'il accepte.

Quand l'une ou l'autre partie tente d'utiliser une fonctionnalité de récupération d'erreur au delà de ce qui est négocié, les tentatives de récupération PEUVENT échouer, sauf si il existe un accord a priori, qui sort du domaine d'application de ce document, entre les deux parties pour fournir une telle prise en charge.

Les mises en œuvre DOIVENT prendre en charge le niveau "0" de récupération d'erreur, alors que le reste est de mise en œuvre FACULTATIVE. En termes de mise en œuvre, le découpage ci-dessus signifie que l'augmentation de sophistication à chaque niveau suivante est exigée :

Transition de niveau	Exigence incrémentielle
0->1	retransmissions de PDU sur la même connexion
1->2	retransmission à travers les connexions et réallocation d'allégeance

7.2 Réessai et réallocation dans la récupération

Cette section résume deux caractéristiques importantes et d'une certaine façon apparentées, du protocole iSCSI qui sont utilisées dans la récupération d'erreurs.

7.2.1 Usage de Retry

Par le nouvel envoi de la même PDU de commande iSCSI ("retry") en l'absence d'un accusé de réception de commande (au moyen d'une mise à jour de Numéro de séquence de commande attendu) ou d'une réponse, un initiateur tente de "boucher" (ce qu'il pense être) les discontinuités de l'ordre de numéros de séquence de commande du côté de la cible. Les PDU de commandes éliminées, dues à des erreurs de résumé, peuvent avoir créé ces discontinuités.

Retry NE DOIT PAS être utilisé pour des raisons autres que des trous à boucher dans la séquence des commandes et, en particulier, ne peut pas être utilisé pour demander des retransmissions de PDU à une cible. De telles demandes de retransmission de PDU pour une commande en cours actuellement allégeante peuvent être faites en utilisant le mécanisme SNACK décrit au paragraphe 11.16, bien que l'usage de SNACK soit FACULTATIF.

Chez Ies initiateurs, au titre du bouchage des trous de la séquence de commandes comme décrit ci-dessus, les réessais produits par inadvertence pour des commandes allégeantes déjà en cours (c'est-à-dire, les cibles n'ont pas vu les discontinuités dans l'ordre de numéros de séquence de commande) les commandes dupliquées sont ignorées en silence par les cibles comme spécifié au paragraphe 4.2.2.1.

Quand une commande iSCSI est réessayée, la PDU de commande DOIT porter l'étiquette de tâche d'initiateur originale et les attributs de fonctionnement originaux (par exemple, fanions, noms de fonction, LUN, CDB, etc.) ainsi que le numéro de séquence de commande original. La commande réessayée DOIT être envoyée sur la même connexion que la commande originale, sauf si la connexion originale a déjà été retirée avec succès.

7.2.2 Réallocation d'allégeance

En produisant une demande de gestion de tâche "TASK REASSIGN" (paragraphe 11.5.1) l'initiateur signale son intention de continuer une commande déjà active (mais sans allégeance actuelle de connexion) au titre de la récupération de connexion. Cela signifie qu'une nouvelle allégeance de connexion est demandée pour la commande, qui cherche à l'associer à la connexion sur laquelle la demande de gestion de tâche est produite. Avant que la réallocation d'allégeance soit tentée pour une tâche, un Logout implicite ou explicite avec le code de cause "supprimer la connexion pour récupération" (voir le paragraphe 11.14.1) DOIT être achevé avec succès pour la connexion précédente à laquelle la tâche obeïssait.

En réallouant l'allégeance de connexion pour une commande, la cible DEVRAIT continuer la commande à partir de son état en cours. Par exemple, quand elle réalloue des commandes de lecture, la cible DEVRAIT tirer parti du champ ExpDataSN fourni par la demande de fonction de gestion de tâche (qui doit être réglée à 0 si il n'y a pas de transfert de données) et amener la commande de lecture à son achèvement en envoyant les données restantes et en envoyant (ou en renvoyant) l'état. Le ExpDataSN accuse réception de toutes les données envoyées jusque là, mais sans inclure la PDU Data-In et/ou R2T avec le DataSN (ou R2TSN) égal au ExpDataSN. Cependant, la cible peut choisir d'envoyer/recevoir toutes les données non acquittées ou toutes les données sur une réallocation d'allégeance de connexion si elle est incapable de récupérer ou maintenir un état précis. Les initiateurs NE DOIVENT PAS demander ensuite la retransmission des données par un Data SNACK pour les PDU d'un numéro inférieur à ExpDataSN (c'est-à-dire, avant le numéro de séquence acquitté). Pour tous les types de commandes, une demande de réallocation implique que la tâche est toujours considérée comme en cours par l'initiateur, et la cible doit conclure la tâche de la façon appropriée si la cible retourne la réponse "Fonction achevée" à la demande de réallocation. Cela peut éventuellement impliquer la retransmission de PDU de données/R2T/état si nécessaire mais DOIT impliquer la (re)transmission des PDU d'état.

Il est FACULTATIF pour les cibles de prendre en charge la réallocation d'allegéance. Cette capacité est négociée via la clé de texte “Niveau de récupération d'erreur” durant l'établissement de connexion. Quand une cible ne prend pas en charge la réallocation d'allégeance, elle DOIT répondre avec un code de réponse de gestion de tâche de "Réallocation d'allégeance de tâche non prise en charge". Si la réallocation d'allégeance est prise en charge par la cible mais si la tâche est encore allégeante à une connexion différente, ou si un Logout réussi de récupération de la connexion précédemment allégeante n'a pas été effectué, la cible DOIT répondre avec un code de réponse de gestion de tâche de "Tâche toujours allégeante".

Si la réallocation d'allégeance est prise en charge par la cible, la réponse de gestion de tâche à la demande de réallocation DOIT être produite avant que la réallocation devienne effective.

Si une commande SCSI qui implique une entrée de données est réallouée, toute étiquette SNACK qui tient pour une réponse finale de la connexion d'origine est supprimée, et la valeur par défaut de 0 DOIT être utilisée à la place.

7.3 Usage de la PDU Rejet en récupération

Les cibles NE DOIVENT PAS terminer implicitement une tâche active en envoyant une PDU Rejet pour toute PDU échangée durant la vie de la tâche. Si la cible décide de terminer la tâche, une PDU de réponse (SCSI, Text, Task, etc.) doit être retournée par la cible pour conclure la tâche. Si la tâche n'a jamais été active avant le rejet (c'est-à-dire, si le rejet est sur la PDU de commande) les cibles ne devraient pas envoyer d'autre réponse parce que la commande elle-même va être éliminée.

Cette règle signifie que l'initiateur peut éventuellement attendre une réponse à réception des rejets, si le rejet reçu est pour une PDU autre que la PDU de commande elle-même. Les rejets non de commande ont seulement une valeur de diagnostic pour enregistrer les erreurs, et ils peuvent être utilisés pour les décisions de retransmission des initiateurs.

Le numéro de séquence de commande de la PDU de commande rejetée (si elle est une commande non immédiate) NE DOIT PAS être considéré comme reçu par la cible (c'est-à-dire, un trou de la séquence de commandes doit être supposé pour le Numéro de séquence de commande) quand bien même le Numéro de séquence de commande de la PDU de commande rejetée pourrait être tenu pour certain. À réception du rejet, l'initiateur DOIT boucher le trou de numéro de séquence de commande afin de continuer d'utiliser la session. Le trou peut être bouché soit en transmettant une PDU de commande avec le même numéro de séquence de commande, soit en interrompant la tâche (voir au paragraphe 7.11 des informations concernant la façon dont une interruption peut boucher un trou de numéro de séquence de commande).

Quand une PDU de données est rejetée et que son DataSN peut être tenu pour certain, une cible DOIT avancer le ExpDataSN pour la salve de données en cours si un R2T de récupération est généré. La cible PEUT avancer son ExpDataSN si elle ne tente pas de récupérer la PDU de données perdue.

7.4 Considérations de récupération d'erreur pour les sessions de découverte

7.4.1 Niveau de récupération d'erreur pour les sessions de découverte

La négociation de la clé Niveau de récupération d'erreur n'est pas exigée pour les sessions de découverte -- c'est-à-dire, pour les sessions qui ont négocié "SessionType=Discovery" – parce que la valeur par défaut de 0 est nécessaire et suffisante pour les sessions de découverte. Il est cependant possible que certaines mises en œuvre iSCSI traditionnelles puissent tenter de négocier la clé Niveau de récupération d'erreur sur des sessions de découverte. Quand une telle tentative de négociation est faite par le côté distant, une mise en œuvre iSCSI conforme DOIT proposer une valeur de 0 (zéro) en réponse. Le niveau de récupération d'erreur opérationnel pour les sessions de découverte DOIT donc être 0. Cela découle naturellement des contraintes de fonctionnalités que le paragraphe 4.3 impose aux sessions de découverte.

7.4.2 Sémantique de réinstallation pour les sessions de découverte

Les sessions de découverte sont destinées à être relativement courtes. Les initiateurs ne sont pas supposés établir plusieurs sessions de découverte au même portail réseau iSCSI. Un initiateur peut utiliser le même nom d'initiateur iSCSI et ISID quand il établit différentes sessions uniques avec des cibles différentes et/ou différents groupes de portails. Ce comportement est discuté au paragraphe 10.1.1 et est, en fait, encouragé comme réutilisation prudente des ISID.

La règle ISID au paragraphe 4.4.3 déclare qu'il ne doit pas y avoir plus d'une session avec un quadruplet <Nom d'initiateur, ISID, Nom de cible, Étiquette de groupe de portail cible> correspondant. Bien que l'esprit de la règle ISID s'applique aux sessions de découverte de la même façon que pour les sessions normales, noter que certaines sessions de découverte diffèrent des sessions normales sous deux aspects importants :

a) Parce que l'Appendice C permet qu'une session de découverte soit établie sans spécifier une clé TargetName dans la PDU de demande Login (appelons une telle session une session de découverte "innommée") il n'y a pas de contexte de nœud cible pour appliquer la règle ISID.

b) Les groupes portails ne sont définis que dans le contexte d'un nœud cible. Quand la clé TargetName est de valeur NUL - (c'est-à-dire, non spécifiée) il ne peut donc pas être certifié que la TargetPortalGroupTag appliquera la règle ISID.

Les deux paragraphes suivants décrivent respectivement les sessions de découverte innommées et les sessions de découverte nommées.

7.4.2.1 Sessions de découverte innommées

Pour les sessions de découverte innommées, ni le TargetName ni la TargetPortalGroupTag ne sont disponibles aux cibles afin d'appliquer la règle ISID. La règle suivante s'applique donc.

Règle ISID innommée : les cibles DOIVENT appliquer l'unicité du quadruplet suivant pour les sessions de découverte innommées : <Nom d'initiator, ISID, NUL, Adresse de cible>. La sémantique suivante est impliquée par cette exigence d'unicité.

Les cibles DEVRAIENT permettre l'établissement concurrent d'une session de découverte avec chacun de ses portails réseau par le même accès d'initiateur avec un certain nom de nœud iSCSI et un ISID. Chacune des sessions de découverte concurrentes, si établies par le même accès d'initiateur pour d'autres portails réseau, DOIT être traitée comme une session indépendante -- c'est-à-dire, une session NE DOIT PAS réinitialiser l'autre.

Une nouvelle session de découverte innommée qui a un quadruplet <Nom d'initiateur, ISID, NUL, Adresse de cible> correspondant à une session de découverte existante DOIT réinitialiser la session de découverte innommée existante. Noter donc que seule une session de découverte innommée peut réinitialiser une autre session de découverte innommée.

7.4.2.2 Sessions de découverte nommées

Pour les sessions de découverte nommées, la clé TargetName est spécifiée par l'initiateur, et donc la cible peut sans ambiguïté certifier aussi la TargetPortalGroupTag. Comme les quatre éléments du quadruplet sont connus, la règle ISID DOIT être appliquée sans changement par les cibles par rapport à la sémantique du paragraphe 4.4.3. Une nouvelle session avec un quadruplet <Nom d'initiateur, ISID, Nom de cible, Étiquette de groupe de portails cible> correspondant va donc réinitialiser une session existante. Noter que dans ce cas, toute nouvelle session iSCSI (de découverte ou normale) avec le quadruplet correspondant, peut réinitialiser une session iSCSI de découverte nommée existante.

7.4.3 PDU cibles durant la découverte

Les cibles NE DEVRAIENT PAS envoyer de réponses autres qu'une réponse Text et une réponse Logout sur une session de découverte, une fois dans la phase Pleines caractéristiques.

Note de mise en œuvre : une cible peut simplement abandonner la connexion dans une session de découverte quand elle aurait demandé un Logout via un message asynchrone sur des sessions normales.

7.5 Gestion des fins de temporisation de connexion

iSCSI définit deux valeurs de temporisation globales par session (en secondes) -- Time2Wait et Time2Retain – qui sont applicables quand une connexion iSCSI en phase Pleines caractéristiques est mise hors service soit intentionnellement, soit par exception. Time2Wait est le "temps de répit" initial avant de tenter un désétablissement explicite/implicite pour le CID en question ou une réallocation de tâche pour les tâches affectées (si il y en a). Time2Retain est le temps maximum après l'intervalle de répit initial pendant lequel la tâche et/ou le ou les états de connexion ont la garantie d'être maintenus sur la cible pour assurer une possible tentative de récupération. Les tentatives de récupération pour la connexion et/ou tâches NE DEVRAIENT PAS être faites avant Time2Wait secondes mais DOIVENT être achevées dans les Time2Retain secondes après cette période d'attente initiale de Time2Wait.

7.5.1 Temporisations sur les événements exceptionnels de transport

Une fermeture de connexion de transport ou une réinitialisation de transport sans aucune interaction de protocole iSCSI précédente informant les points d'extrémité du fait cause la terminaison abrupte d'une connexion iSCSI en phase de pleines caractéristiques. Les valeurs de temporisation à utiliser dans ce cas sont les valeurs négociées des clé textuelles DefaultTime2Wait (paragraphe 13.15) et DefaultTime2Retain (paragraphe 13.16) pour la session.

7.5.2 Temporisations sur des décommissionnements planifiés

Tout décommissionnement planifié d'une connexion iSCSI en phase de pleines caractéristiques est précédé par une PDU de réponse Logout ou d'une PDU de message asynchrone. Les valeurs des champs Time2Wait et Time2Retain (paragraphe 11.15) dans une PDU de réponse Logout, et les champs Parameter2 et Parameter3 d'un message asynchrone

(les types AsyncEvent "abandonner la connexion" ou "abandonner toutes les connexions"; voir paragraphe 11.9.1) spécifient les valeurs de temporisation à utiliser dans chacun de ces cas.

Ces valeurs de temporisation ne sont applicables qu'à la connexion affectée et aux tâches actives sur cette connexion. Ces valeurs de temporisation n'ont pas d'incidence sur les temporisateurs de l'initiateur (si il y en a) qui courent déjà sur les connexions ou tâches associées à cette session.

7.6 Terminaison implicite de tâches

Une cible termine implicitement les tâches actives du fait de la dynamique du protocole iSCSI dans les cas suivants :

a) Quand une connexion est fermée implicitement ou explicitement avec le code de cause "fermer la connexion" et qu'il y a des tâches actives allégeantes à cette connexion.

b) Quand une connexion est défaillante et que finalement l'état de la connexion arrive en fin de temporisation (transition d'état M1 au paragraphe 8.2.2) et qu'il y a des tâches actives allégeantes à cette connexion.

c) Quand un Logout réussi avec le code de cause "supprimer la connexion pour récupération" est effectué alors qu'il y a des tâches actives allégeantes à cette connexion, et que ces tâches arrivent finalement en fin de temporisation après les périodes Time2Wait et Time2Retain sans réallocation d'allégeance.

d) Quand une connexion est implicitement ou explicitement cloturée avec le code de cause "fermer la session" et qu'il y a des tâches actives dans cette session.

Si les tâches terminées dans les cas a), b), c), et d) ci-dessus sont des tâches SCSI, elles doivent être terminées en interne comme avec l'état Vérifier la condition. Cet état n'a de signification que pour le traitement approprié de l'état interne SCSI et les efffets SCSI colatéraux par rapport à l'ordre, parce que cet état n'est jamais communiqué comme état de terminaison à l'initiateur. Cependant, des actions supplémentaires peuvent devoir être effectuées au niveau SCSI, selon le contexte SCSI comme défini par les normes SCSI (par exemple, les commandes en file d'attente et ACA; UA pour la prochaine commande sur le nexus I_T dans les cas a) b) et c); etc. -- voir [SAM2] et [SPC3]).

7.7 Erreurs de format

Les deux violations explicites suivantes des règles de présentation de PDU sont des erreurs de format :
a) contenu illégal de tout champ d'en-tête de PDU sauf le Opcode (les valeurs légales sont spécifiées à la Section 11).
b) contenu de champ incohérent (les contenus de champ cohérents sont spécifiés à la Section 11).

Les erreurs de format indiquent une faute majeure de mise en œuvre dans une des parties.

Quand une cible ou un initiateur reçoit une PDU iSCSI avec une erreur de format, il DOIT immédiatement terminer toutes les connexions de transport dans la session avec un “fermer la connexion” ou un “réinitialisation de connexion”, et transformer l'erreur de format en récupération de session (voir le paragraphe 7.1.4.4).

Toute erreur de construction de PDU détectée par l'initiateur DOIT être considérée comme une erreur de format. Des exemples de telles erreurs sont :
- NOP-In avec une TTT valide mais un LUN invalide
- NOP-In avec une ITT valide (c'est-à-dire, une réponse NOP-In) et aussi une TTT valide
- PDU de réponse SCSI avec Status=CHECK CONDITION, mais DataSegmentLength = 0

7.8 Erreurs de résumé

La discussion ci-dessous concernant les choix légaux du traitement des erreurs de résumé exclut la récupération de session comme option explicite, mais l'une ou l'autre partie qui détecte une erreur de résumé peut choisir de transformer l'erreur en récupération de session.

Quand une cible ou un initiateur reçoit des PDU iSCSI avec une ereur de résumé d'en-tête, il DOIT soit éliminer l'en-tête et toutes les données jusqu'au début de la PDU suivante, soit clore la connexion. Parce que l'erreur de résumé indique que le champ Longueur de l'en-tête peut avoir été corrompu, la localisation du début de la PDU suivante doit être identifiée de façon fiable par d'autres moyens, comme le fonctionnement d'une couche de synchronisation et de pilotage.

Quand une cible reçoit une PDU iSCSI avec une erreur de résumé de charge utile, elle DOIT répondre avec une PDU Rejet avec un code de cause de Data-Digest-Error et éliminer la PDU.

- Si la PDU éliminée est une PDU de données iSCSI sollicitée ou non sollicitée (pour des données immédiates dans une PDU de commande, la règle de PDU non de données ci-dessous s'applique) la cible DOIT faire une des choses suivantes :
  a) Demander la retransmission avec un R2T de récupération,
  b) Terminer la tâche avec une PDU de réponse SCSI avec un état Vérifier la condition et une condition iSCSI de "Erreur de CRC de service de protocole" (paragraphe 11.4.7.2). Si la cible choisit de mettre en œuvre cette option, elle DOIT attendre de recevoir toutes les données (signalée par une PDU de données avec le bit Final établi pour tous les R2T en instance) avant d'envoyer la PDU de réponse SCSI. Une commande de gestion de tâche (comme un ABORT TASK) provenant de l'initiateur durant cette attente peut aussi conclure la tâche.

- Aucune autre action n'est nécessaire pour les cibles si la PDU éliminée n'est pas une PDU de données. Dans le cas de données immédiates présentes sur une commande éliminée, les données immédiates sont implicitement récupérées quand la tâche est réessayée (voir le paragraphe 7.2.1) suivies par le transfert entier de données pour la tâche.

Quand un initiateur reçoit une PDU iSCSI avec une erreur de résumé de charge utile, il DOIT éliminer la PDU.

- Si la PDU éliminée est une PDU iSCSI de données, l'initiateur DOIT faire une des choses suivantes :

a) Demander la PDU de données désirée avec SNACK. En réponse au SNACK, la cible DOIT soit rencoyer la PDU de données, soit rejeter le SNACK avec une PDU Rejet avec un code de cause de "Rejet de SNACK", et dans ce cas :
   a.1) Si l'état n'a pas déjà été envoyé pour la commande, la cible DOIT terminer la commande avec un état Vérifier la condition et une condition iSCSI de "SNACK rejeté" (paragraphe 11.4.7.2).
   a.2) Si l'état a déjà été envoyé, aucune autre action n'est nécessaire pour la cible. L'initiateur DOIT dans ce cas attendre que l'état soit reçu et ensuite l'éliminer, afin de signaler en interne l'achèvement avec un état Vérifier la condition et une condition iSCSI de "Erreur de CRC de service de protocole" (paragraphe 11.4.7.2).
b) Interrompre la tâche et terminer la commande avec une erreur.

- Si la PDU éliminée est une PDU de réponse ou une PDU non sollicitée (par exemple, Async, Reject), l'initiateur DOIT faire une des choses suivantes :
  a) Demander la retransmission de la PDU avec un état de SNACK.
  b) Fermer la connexion pour récupération, et continuer les tâches sur une instance de connexion différente comme décrit au paragraphe 7.2.
  c) Désenregistrer pour clore la connexion (interrompre toutes les commandes associées à la connexion).

Noter qu'une PDU non sollicitée porte la prochaine valeur de Numéro de séquence d'état sur une connexion iSCSI, avançant par là le numéro de séquence d'état. Quand un initiateur élimine une de ces PDU à cause d'une erreur de résumé de charge utile, la PDU entière, y compris l'en-tête, DOIT être éliminée. Par conséquent, l'initiateur DOIT traiter l'exception comme une perte de toute autre PDU de réponse sollicitée.

7.9 Erreurs de séquence

Quand un initiateur reçoit une PDU iSCSI R2T/données avec un R2TSN/DataSN déclassé ou une PDU de réponse SCSI avec un ExpDataSN qui implique des PDU de données manquantes, cela signifie que l'initiateur doit avoir détecté une erreur de résumé d'en-tête ou de charge utile sur une ou plusieurs PDU R2T/données antérieures.

L'initiateur DOIT traiter ces erreurs de résumé implicites comme décrit au paragraphe 7.8. Quand une cible reçoit une PDU de données avec un DataSN déclassé, cela signifie que la cible doit avoir détecté une erreur de résumé d'en-tête ou de charge utile sur au moins une des PDU de données antérieures. La cible DOIT traiter ces erreurs de résumé implicites comme décrit au paragraphe 7.8.

Quand un initiateur reçoit une PDU d'état SCSI avec un numéro de séquence d'état déclassé qui implique des réponses manquantes, il DOIT traiter la ou les PDU d'état manquantes comme décrit au paragraphe 7.8. Comme effet colatéral de la réception des réponses manquantes, l'initiateur peut découvrir des PDU de données manquantes. Si l'initiateur veut récupérer les données manquantes pour une commande, il NE DOIT PAS accuser réception des réponses reçues qui commencent au numéro de séquence d'état de la commande pertinente jusqu'à ce qu'il ait fini de recevoir toutes les PDU de données de la commande.

Quand un initiateur reçoit des R2TSN dupliqués (dus à une retransmission proactive des R2T par la cible) ou des numéros de séquence de données dupliqués (dus à des SNACK proactifs par l'initiateur) il DOIT éliminer les dupliqués.

7.10 Vérification d'erreur de message

Dans les mises en œuvre iSCSI d'aujourd'hui, il y a eu des incertitudes concernant la mesure dans laquelle les messages entrants doivent être vérifiés quant aux erreurs de protocole, au delà de ce qui est strictement exigé pour le traitement des messages entrants. Ce paragraphe traite de cette question.

Sauf si le présent document l'exige, une mise en œuvre iSCSI n'est pas obligée de faire une vérification exhaustive de la conformité au protocole sur une PDU iSCSI entrante. En particulier, une mise en œuvre iSCSI n'est pas obligée de faire une double vérification de la conformité de la mise en œuvre iSCSIO distante aux exigences du protocole.

7.11 Temporisations SCSI

Un initiateur iSCSI PEUT tenter de boucher un trou de numéro de séquence de commande sur l'extrémité cible (en l'absence d'accusé de réception de la commande au moyen de Numéro de séquence de commande attendu) avant la fin de temporisation de l'ULP en réessayant la commande non acquittée, comme décrit au paragraphe 7.2.

Sur une fin de temporisation d'ULP pour une commande (qui portait un numéro de séquence de commande de n) si l'initiateur iSCSI a l'intention de continuer la session, il DOIT interrompre la commande en utilisant soit une demande de fonction de gestion de tâche appropriée pour la commande spécifique, soit un Logout "clore la connexion".

Quand on utilise un ABORT TASK, si le numéro de séquence de commande attendu est encore inférieur à (n + 1), la cible peut voir la demande d'interruption tout en manquant la commande d'origine elle-même, pour une des raisons suivantes :
- La commande d'origine a été abandonnée à cause d'une erreur de résumé,
- La connexion sur laquelle la commande d'origine a été envoyée a été supprimée avec succès. Sur une sortie de connexion, les commandes non acquittées produites sur la connexion en cours de fermeture sont éliminnées.

Si la demande d'interruption est reçue et si la commande d'origine manque, les cibles DOIVENT considérer que la commande d'origine avec ce RefCmdSN a été reçue et produire une réponse de gestion de tâche avec le code de réponse "Fonction achevée". Cette réponse conclut la tâche aux deux extrémités. Si la demande d'interruption est reçue et si la cible peut déterminer (sur la base de l'étiquette de tâche référencée) que la commande a été reçue et exécutée, et aussi que la réponse a été envoyée avant l'interruption, alors la cible DOIT répondre avec le code de réponse "La tâche n'existe pas".

7.12 Échecs de négociation

Les séquences de demande et réponse Text, quand utilisées pour régler/négocier les paramètres de fonctionnement, constituent le réglage de négociation/paramètres. Un échec de négociation est considéré comme étant un ou plusieurs de ce qui suit :
- pour une clé négociée, aucun des choix n'est acceptable à un des côtés de la négociation ;
- pour une clé déclarative, la valeur déclarée n'est pas acceptable à l'autre côté de la négociation ;
- la demande Text est arrivée en fin de temporisation et est éventuellement terminée ;
- la demande Text a eu une réponse de PDU Rejet.

Les deux règles suivantes devraient être utilisées pour traiter les échecs de négociation :
a) Durant l'établissement de connexion, tout échec de la négociation DOIT être considéré comme défaillance du processus d'établissement ; la phase Login, ainsi que la connexion, DOIT être terminée. Si la cible détecte la défaillance, elle doit terminer l'établissement de connexion avec le code de réponse de Login approprié.
b) Une défaillance de négociation durant la phase de pleines caractéristiques va terminer la séquence entière de négociation, qui peut consister en une série de demandes Text qui utilise la même étiquette de tâche d'initiateur. Les paramètres de fonctionnement de la session ou de la connexion DOIVENT continuer d'être les valeurs objet d'accord durant une négociation antérieure réussie (c'est-à-dire, tous les résultats partiels de cette négociation non réussie NE DOIVENT PAS entrer en effet et DOIVENT être éliminés).

7.13 Erreurs de protocole

La transposition de messages tramés sur une connexion "à flux" comme TCP rend les mécanismes proposés vulnérables aux simples erreurs de tramage logicielles. D'un autre côté, l'introduction de mécanismes de tramage pour limiter les effets de ces erreurs peut être onéreuse en termes de performances pour des mises en œuvre simples. Les numéros de séquence de commandes et les mécanismes pour abandonner et rétablir les connexions (discutés au début de la Section 7) aident à traiter ce type d'erreurs de transposition.

Toutes les violations de séquence d'échange de PDU iSCSI spécifiées dans le présent document sont aussi des erreurs de protocole. Cette catégorie d'erreurs ne peut être traitée qu'en corrigeant les mises en œuvres ; iSCSI définit Rejet et les codes de réponse pour permettre cela.

7.14 Échecs de connexion

iSCSI peut garder une session en fonctionnement si il est capable de garder/établir en temps utile au moins une connexion TCP entre l'initiateur et la cible. Les cibles et/ou les initiateurs peuvent reconnaître une connexion défaillante avec des moyens de niveau transport (TCP), un trou dans les numéros de séquence de commande, un flux de réponses qui n'est pas rempli pendant longtemps, ou un NOP iSCSI défaillant (agissant comme un ping). Ce dernier PEUT être utilisé périodiquement pour augmenter la vitesse et la probabilité de détecter des défaillances de connexion. Comme exemple de moyens au niveau transport, les initiateurs et les cibles PEUVENT aussi utiliser l'option Garder en vie (voir la [RFC1122]) sur la connexion TCP pour permettre une détection précoce de défaillance de liaison sur des liaisons par ailleurs inactives.

Sur une défaillance de connexion, initiateur et cible DOIVENT faire une des choses suivantes :
a) tenter la récupération de connexion au sein de la session (récupération de connexion) ;
b) déconnecter la connexion avec le code de cause "clore la connexion" (paragraphe 11.14.5), produire à nouveau les commandes manquantes, et terminer implicitement toutes les commandes actives. Cette option exige la prise en charge de la classe de récupération au sein de la connexion (récupération au sein de la connexion) ;
c) effectuer la récupération de session (récupération de session).

L'un ou l'autre côté peut choisir de monter à la récupération de session (via l'abandon par l'initiateur de toutes les connexions ou via un message asynchrone qui annonce l'intention similaire de la part d'une cible) et l'autre côté DOIT céder sa priorité. Sur une défaillance de connexion, une cible DOIT terminer et/ou éliminer toute commande immédiate active, sans considération de l'option utilisée (c'est-à-dire, les commandes immédiates ne sont pas récupérables à travers les défaillances de connexion).

7.15 Erreurs de session

Si toutes les connexions d'une session sont défaillantes et ne peuvent pas être rétablies dans un court délai, ou si les initiateurs détectent des erreurs répétées de protocole, un initiateur peut choisir de terminer une session et en établir une nouvelle.

Dans ce cas, l'initiateur effectue les actions suivantes :
- Réinitialise ou clôture toutes les connexions de transport.
- Termine toutes les demandes en instance avec une réponse appropriée avant d'initier une nouvelle session. Si il est prévu de rétablir le même nexus I_T, l'initiateur DOIT employer la réinstallation de session (voir le paragraphe 6.3.5).

Quand la temporisation de la session se termine (l'état de connexion arrive en fin de temporisation pour la dernière connexion défaillante) sur la cible, elle effectue les actions suivantes :
- Réinitialise ou cloture les connexions TCP (clôt la session).
- Termine toutes les tâches actives qui étaient allégeantes à la ou les connexions qui constituaient la session.

Une cible DOIT aussi être prête à traiter une demande de réinitialisation de session de l'initiateur qui peut s'adresser à des erreurs de session.

8. Transitions d’état

Les connexions et sessions iSCSI passent par plusieurs états bien définis entre le moment de leur création et celui où elles sont supprimées.

Les transitions d'état de connexion sont décrites dans deux ensembles de diagrammes d'état séparés mais dépendants pour faciliter la compréhension. Le premier ensemble de diagrammes, "diagrammes d'état de connexion standard", décrit les transitions d'état de connexion quand la connexion iSCSI n'attend pas, ou ne subit pas, un nettoyage au moyen d'une sortie de connexion explicite ou implicite. Le second ensemble, "diagramme d'état de nettoyage de connexion", décrit les transitions d'état de connexion lorsque on effectue le nettoyage de connexion iSCSI. Alors que le premier ensemble a deux diagrammes – un pour l'initiateur et un pour la cible -- le second ensemble a un seul diagramme applicable à la fois aux initiateurs et aux cibles.

Le "diagramme d'état de session" décrit les transitions d'état par lesquelles une session iSCSI va passer durant sa vie, et il dépend des états de connexions iSCSI éventuellement multiples qui participent à la session.

Les états et les transitions sont décrits en texte, tableaux, et diagrammes. Les diagrammes sont utilisés pour l'illustration. Le texte et les tableaux sont la spécification elle-même.

8.1 Diagrammes d'état de connexion standard

8.1.1 Descriptions d'état pour les initiateurs et les cibles

Les descriptions d'état pour le diagramme standard d'état de connexion sont comme suit :

S1 : FREE *(libre)*
- initiateur : état en instanciation, ou après clôture réussie de connexion.
- cible : état en instanciation, ou après clôture réussie de connexion.

S2 : XPT_WAIT
- initiateur : attend une réponse à sa demande d'établissement de connexion de transport.
- cible : illégal.

S3 : XPT_UP
- initiateur : illégal.
- cible : attend que le processus d'établissement commence.

S4 : IN_LOGIN
- initiateur : attend que le processus d'établissement se conclut, impliquant éventuellement plusieurs échanges de PDU.
- cible : attend que le processus d'établissement se conclut, impliquant éventuellement plusieurs échanges de PDU.

S5 : LOGGED_IN
- initiateur : dans la phase de pleines caractéristiques, attente de tous événements internes, iSCSI, et de transport.
- cible : dans la phase de pleines caractéristiques, attente de tous événements internes, iSCSI, et de transport.

S6 : IN_LOGOUT
- initiateur : attend une réponse de désétablissement.
- cible : attend un événement interne signalant l'achèvement du traitement de désétablissement.

S7 : LOGOUT_REQUESTED
- initiateur : attend un événement interne signalant qu'il est prêt à procéder au désétablissement.
- cible : attend que le processus de désétablissement commence après avoir demandé un désétablissement via un message asynchrone.

S8 : CLEANUP_WAIT
- initiateur : attend le contexte et/ou les ressources pour initier le traitement de nettoyage pour ce CSM.
- cible : attend que le processus de nettoyage commence pour ce CSM.

8.1.2 Descriptions des transitions d'état pour les initiateurs et les cibles

T1 :
- initiateur : la demande de connexion de transport a été faite (par exemple, TCP SYN envoyé).
- cible : illégal.

T2 :
- initiateur : la demande de connexion de transport est arrivée en fin de temporisation, une réinitialisation de transport a été reçue, ou un événement interne de réception d'une réponse de désétablissement (succès) sur une autre connexion pour une demande de désétablissement "clore la session" a été reçue.
- cible : illégal.

T3 :
- initiateur : illégal.
- cible : a reçu une demande valide de connexion de transport qui établit la connexion de transport.

T4 :

- initiateur : la connexion de transport est établie, invitant donc l'initiateur à commencer l'établissement iSCSI.
- cible : la demande d'établissement iSCSI a été reçue.

T5 :
- initiateur : la réponse d'établissement iSCSI finale a été reçue avec une classe d'état de zéro.
- cible : la demande d'établissement iSCSI finale de conclure la phase d'établissement a été reçue, invitant donc la cible à envoyer la réponse d'établissement iSCSI finale avec une classe d'état de zéro.

T6:
- initiateur : illégal.
- cible : fin de temporisation de l'attente d'un établissement iSCSI Login, l'indication de déconnexion du transport a été reçue, la réinitialisation du transport a été reçue, ou un événement interne indiquant qu'une fin de temporisation de transport a été reçue. Dans tous ces cas, la connexion va être fermée.

T7 :
- initiateur : Un des événements suivants à causé la transition :
  a) La réponse finale iSCSI de désétablissement a été reçue avec une classe d'état non zéro.
  b) Fin de temporisation du désétablissement.
  c) Une indication de déconnexion de transport a été reçue.
  d) Une réinitialisation de transport a été reçue.
  e) Un événement interne indiquant une fin de temporisation de transport a été reçu.
  f) Un événement interne de réception d'une réponse de désétablissement (succès) sur une autre connexion pour une demande de désétablissement "clore la session" a été reçu.

Dans tous ces cas, la connexion de transport est close.

- cible : Un des événements suivants a causé la transition :
  a) La demande finale iSCSI de désétablissement pour conclure la phase de désétablissement a été reçue, invitant la cible à envoyer la réponse finale iSCSI de désétablissement avec une classe d'état non zéro.
  b) Fin de temporisation de désétablissement.
  c) Une indication de déconnexion de transport a été reçue.
  d) Une réinitialisation de transport a été reçue.
  e) Un événement interne indiquant une fin de temporisation de transport a été reçu.
  f) Sur une autre connexion, une demande de désétablissement"clore la session" a été reçue.

Dans tous ces cas, la connexion va être close.

T8 :
- initiateur : Un événement interne de réception d'une demande de désétablissement (succès) sur une autre connexion pour une demande de désétablissement "clore la session" a été reçu, fermant donc cette connexion et n'exigeant pas d'autre nettoyage,
- cible : Un événement interne d'envoi d'une réponse de désétablissement (succès) sur une autre connexion pour une demande de désétablissement "clore la session" a été reçu, ou un événement interne d'une réinstallation réussie de connexion/session a été reçu, invitant donc la cible à fermer proprement cette connexion.

T9, T10 :
- initiateur : Un événement interne qui indique qu'il est prêt à commencer le processus de désétablissement a été reçu, invitant donc l'initiateur à envoyer un désétablissement iSCSI.
- cible : Une demande de désétablissement iSCSI a été reçue.

T11, T12 :
- initiateur : Une PDU asynchrone avec un AsyncEvent "Demande de désétablissement" a été reçue.
- cible : Un événement interne qui exige le décommissionnement de la connexion a été reçu, causant donc l'envoi d'une PDU Asynchrone avec un AsyncEvent "Demande de désétablissement".

T13 :
- initiateur : Une réponse de désétablissement iSCSI (succès) a été reçue, ou un événement interne de réception d'une réponse de désétablissement (succès) sur une autre connexion pour une demande de désétablissement "clore la session" a été reçu.
- cible : Un événement interne a été reçu qui indique le traitement réussi du désétablissement, qui invite à l'envoi d'une réponse de désétablissement iSCSI (succès) ; un événement interne d'envoi d'une réponse de désétablissement (succès) sur une autre connexion pour une demande de désétablissement "clore la session" a été reçu ; ou un événement interne de réinitialisation réussie de connexion/session a été reçu. Dans tous ces cas, la connexion de transport est close.

T14 :

- initiateur : Une PDU Asynchrone avec un AsyncEvent "Demande de désétablissement" a encore été reçue.
- cible : illégal.

T15, T16 :
- initiateur : Un ou plusieurs des événements suivants ont causé cette transition :
  a) Un événement interne qui indique une fin de temporisation de connexion de transport a été reçu, invitant donc à une réinitialisation de transport ou une fermeture de connexion de transport.
  b) Une réinitialisation de transport a été reçue.
  c) Une indication de déconnexion de transport a été reçue.
  d) Une PDU Asynchrone avec un AsyncEvent "abandon de connexion" (pour ce CID) a été reçue.
  e) Une PDU Asynchrone avec un AsyncEvent "abandon de toutes les connexions" a été reçue.

- cible : Un ou plusieurs des événements suivants ont causé cette transition :
  a) Événement interne qui indique qu'une fin de temporisation de connexion de transport a été reçue, invitant done à une réinitialisation de transport ou fermeture de connexion de transport.
  b) Un événement interne d'échec de réinitialisation de connexion/session a été reçu.
  c) Une réinitialisation de transport a été reçue.
  d) Une indication de déconnexion de transport a été reçue.
  e) Un événement interne de nettoyage d'urgence a été reçu, qui invite à une PDU Asynchrone avec un AsyncEvent "abandon de connexion" (pour ce CID), ou l'événement "abandon de toutes les connexions".

T17 :
- initiateur : Un ou plusieurs des événements suivants ont causé cette transition :
  a) Une réponse de désétablissement (échec, c'est-à-dire, un état non zéro) a été reçu, ou le désétablissement est arrivé en fin de temporisation.
  b) Un des événements spécifiés pour T15 et T16 s'est produit.

- cible : Un ou plusieurs des événements suivants ont causé cette transition :
  a) Un événement interne qui indique un échec du traitement de désétablissement a été reçu, qui invite à l'envoi d'une réponse de désétablissement (échec, c'est-à-dire, un état non zéro).
  b) Un des événements spécifiés pour T15 et T16 s'est produit.

T18 :
- initiateur : Un événement interne de réception d'une réponse de désétablissement (succès) sur une autre connexion pour une demande de désétablissement "clore la session" a été reçu.
- cible : Un événement interne d'envoi d'une réponse de désétablissement (succès) sur une autre connexion pour une demande de désétablissement "clore la session" a été reçu, ou un événement interne de réinitialisation réussie de connexion/session a été reçu. Dans ces deux cas, la connexion est close.

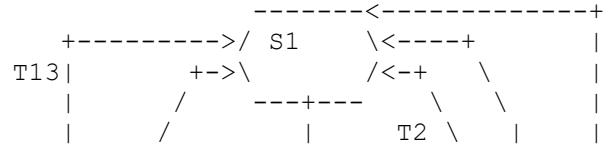
L'état CLEANUP_WAIT (S8) implique qu'il y a de possibles tâches iSCSI qui ne sont pas arrivées à leur terme et sont encore considérées comme actives.

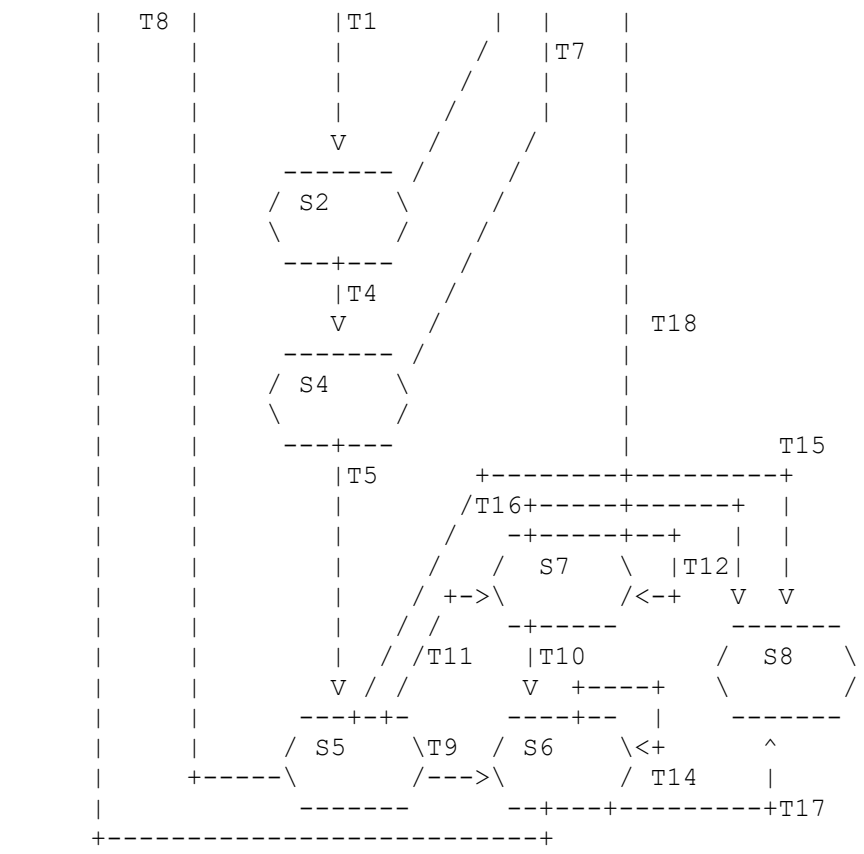
8.1.3 Diagramme d'état de connexion standard pour un initiateur

Noms symboliques des états :
S1 : FREE *(libre)*
S2 : XPT_WAIT *(attente d'établissement)*
S4 : IN_LOGIN *(en cours d'établissement)*
S5 : LOGGED_IN *(connexion établie)*
S6 : IN_LOGOUT *(en cours de désétablissement)*
S7 : LOGOUT_REQUESTED *(désétablissement demamdée)*
S8 : CLEANUP_WAIT *(attente de nettoyage)*

Les états S5, S6, et S7 constituent l'opération Phase de pleines caractéristiques de la connexion.

Le diagramme d'états est le suivant :





Le tableau des transitions d'état suivant représente le diagramme ci-dessus. Chaque rangée représente l'état de départ pour une transition données, qui, après la transition marquée dans une cellule du tableau, va finir dans l'état représenté par la colonne de la cellule. Par exemple, à partir de l'état S1, la connexion prend la transition T1 pour arriver à l'état S2. Les champs marqués "-" correspondent à des transitions indéfinies.

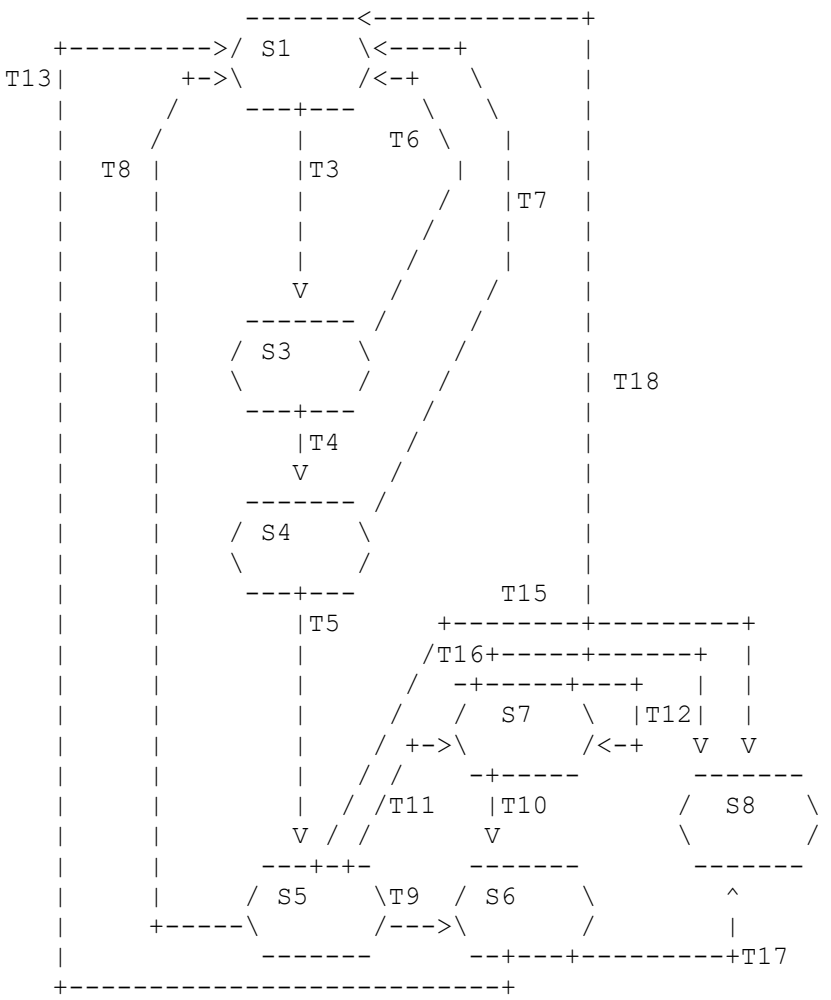
	S1	S2	S4	S5	S6	S7	S8
S1	-	T1	-	-	-	-	-
S2	T2	-	T4	-	-	-	-
S4	T7	-	-	T5	-	-	-
S5	T8	-	-	-	T9	T11	T15
S6	T13	-	-	-	T14	-	T17
S7	T18	-	-	-	T10	T12	T16
S8	-	-	-	-	-	-	-

8.1.4 Diagramme d'état de connexion standard pour une cible

Noms symboliques des états :
S1 : FREE
S3 : XPT_UP
S4 : IN_LOGIN
S5 : LOGGED_IN
S6 : IN_LOGOUT
S7 : LOGOUT_REQUESTED
S8 : CLEANUP_WAIT

Les états S5, S6, et S7 constituent l'opération Phase de pleines caractéristiques de la connexion.

Le diagramme d'état est le suivant:



Le tableau des transitions d'état suivant représente le diagramme ci-dessus et suit les conventions décrites pour le diagramme d'initiateur.

	S1	S3	S4	S5	S6	S7	S8
S1	-	T3	-	-	-	-	-
S3	T6	-	T4	-	-	-	-
S4	T7	-	-	T5	-	-	-
S5	T8	-	-	-	T9	T11	T15
S6	T13	-	-	-	-	-	T17
S7	T18	-	-	-	T10	T12	T16
S8	-	-	-	-	-	-	-

8.2 Diagramme d'état de nettoyage de connexion pour initiateurs et cibles

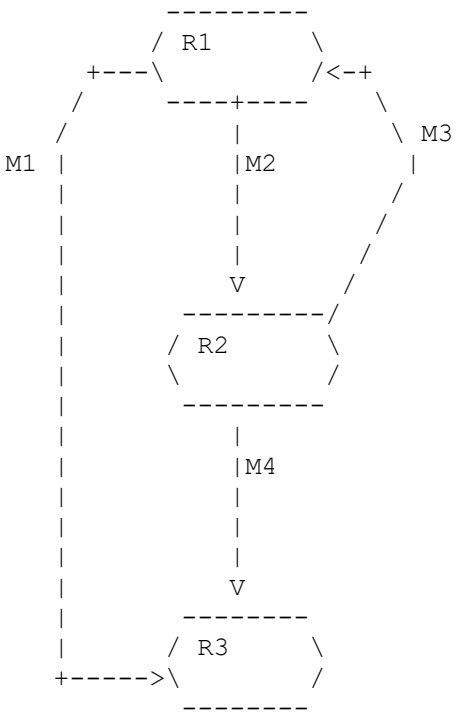
Noms symboliques des états :
R1 : CLEANUP_WAIT (le même que S8)

R2 : IN_CLEANUP
R3 : FREE (le même que S1)

Chaque fois qu'un automate à états de connexion en nettoyage (appelons le CSM-C) entre dans l'état CLEANUP_WAIT (S8), il doit passer par les transitions d'état décrites dans le diagramme d'état de nettoyage de connexion, en utilisant soit a) une connexion séparée en phase de pleines caratéristiques (appelons la CSM-E, pour “explicite”) dans l'état LOGGED_IN dans la même session ou b) une nouvelle connexion de transport (appelons la CSM-I, pour “implicite”) dans l'état FREE qui est à ajouter à la même session. Dans le cas CSM-E, un désétablissement explicite pour le CID qui correspond à CSM-C (comme désétablissement de connexion ou de session) doit être effectué pour achever le nettoyage. Dans le cas de CSM-I, un désétablissement implicite pour le CID qui correspond au CSM-C doit être effectué au moyen d'une réinitialisation de connexion (paragraphe 6.3.4) pour ce CID. Dans l'un et l'autre cas, les échanges de protocole sur CSM-E ou CSM-I déterminent les transitions d'état pour CSM-C. Donc, ce diagramme d'état de nettoyage n'est applicable qu'à l'instance de la connexion en nettoyage (c'est-à-dire, CSM-C). Dans le cas d'un désétablissement implicite, par exemple, CSM-C atteint FREE (R3) au moment où CSM-I atteint LOGGED_IN. Dans le cas d'un désétablissement explicite, CSM-C atteint FREE (R3) quand CSM-E reçoit une réponse de désétablissement réussi tout en continuant d'être dans l'état LOGGED_IN.

Un initiateur doit initier un désétablissement de connexion explicite ou implicite pour une connexion dans l'état CLEANUP_WAIT, si l'initiateur a l'intention de continuer d'utiliser la session iSCSI associée.

Le diagramme d'état suivant s'applique aux initiateurs et aux cibles. (M1, M2, M3, et M4 sont définis au paragraphe 8.2.2.)



Le tableau de transitions d'état suivant représente le diagramme ci-dessus et suit les mêmes conventions que dans les paragraphes précédents.

	R1	R2	R3
R1	-	M2	M1
R2	M3	-	M4
R3	-	-	-

8.2.1 Descriptions d'état pour initiateurs et cibles

R1 : CLEANUP_WAIT (le même que S8)
- initiateur : Attente de l'événement interne pour initier le processus de nettoyage pour CSM-C.

- cible : Attente du début du processus de nettoyage pour CSM-C.

R2 : IN_CLEANUP
- initiateur : Attente de la fin du processus de nettoyage de connexion pour CSM-C.
- cible : Attente de la fin du processus de nettoyage de connexion pour CSM-C.

R3 : FREE (le même que S1)
- initiateur : état final pour CSM-C.
- cible : état final pour CSM-C.

8.2.2 Descriptions des transitions d'état pour initiateurs et cibles

M1 : Un ou plusieurs des événements suivants ont été reçus :
- initiateur :
  * Un événement interne qui indique la fin de la temporisation de l'état de la connexion.
  * Un événement interne de réception d'une réponse de désétablissement réussi sur une connexion différente pour un désétablissement "clore la session".
- cible :
  * Un événement interne qui indique la fin de la temporisation de l'état de la connexion.
  * Un événement interne d'envoi d'une réponse de désétablissement (succès) sur une connexion différente pour un désétablissement "clore la session".

M2 : Un processus implicite/explicite de désétablissement a été initié par l'initiateur.
- Dans l'usage CSM-I :
  * initiateur : un événement interne demandant la réinitialisation de la connexion (ou session) a été reçu, invitant donc à l'envoi d'un établissement de réinitialisation de connexion (ou session), transitant CSM-I à l'état IN_LOGIN.
  * cible : un établissement de réinitialisation de connexion/session a été reçu dans l'état XPT_UP.
- Dans l'usage CSM-E :
  * initiateur : un événement interne a été reçu qui indique qu'un désétablissement explicite a été envoyé pour ce CID dans l'état LOGGED_IN.
  * cible : un désétablissement explicite a été reçu pour ce CID dans l'état LOGGED_IN.

M3 : Une défaillance de désétablissement a été détectée.
- Dans l'usage CSM-I :
  * initiateur : CSM-I a échoué à atteindre LOGGED_IN et est arrivé en FREE à la place.
  * cible : CSM-I a échoué à atteindre LOGGED_IN et est arrivé en FREE à la place.
- Dans l'usage CSM-E :
  * initiateur : soit CSM-E est sorti de LOGGED_IN, soit le désétablissement est arrivé en fin de temporisation et/ou s'est interrompu, soit une réponse de désétablissement (échec) a été reçue.
  * cible : soit CSM-E est sorti de LOGGED_IN, le désétablissement est arrivé en fin de temporisation et/ou s'est interrompu, soit un événement interne qui indique qu'un traitement de désétablissement défaillant a été reçu. Une réponse de désétablissement (échec) a été envoyée dans ce dernier cas.

M4 : Un désétablissement implicite/explicite réussi a été effectué.
- Dans l'usage CSM-I :
  * initiateur : CSM-I atteint l'état LOGGED_IN, ou un événement interne de réception d'une réponse de désétablissement (succès) sur une autre connexion pour une demande de désétablissement de "clore la session" a été reçu.
  * cible : CSM-I a atteint l'état LOGGED_IN, ou un événement interne d'envoi d'une réponse de désétablissement (succès) sur une autre connexion pour une demande de fin d'établissement de "clore la session" a été reçu.
- Dans l'usage CSM-E :
  * initiateur : CSM-E est resté en LOGGED_IN et a reçu une réponse de désétablissement (succès), ou un événement interne de réception d'une réponse de désétablissement (succès) sur une autre connexion pour une demande de fin d'établissement de "clore la session" a été reçu.
  * cible : CSM-E est resté en LOGGED_IN et un événement interne indiquant un traitement réussi de désétablissement a été reçu, ou un événement interne d'envoi d'une réponse de désétablissement (succès) sur une autre connexion pour une demande de désétablissement de "clore la session" a été reçu.

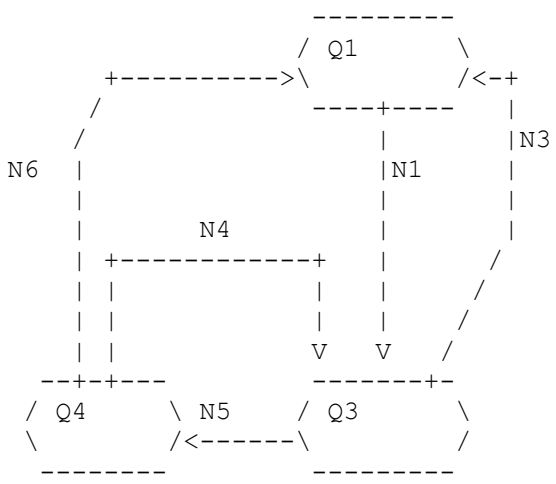
8.3 Diagrammes d'état de session

8.3.1 Diagramme d'état de session pour un initiateur

Noms symboliques des états :
Q1 : FREE
Q3 : LOGGED_IN
Q4 : FAILED

L'état Q3 représente le fonctionnement de la phase Pleines caractéristiques de la session.

Le diagramme d'état est le suivant. (N1, N3, N4, N5, et N6 sont définis au paragraphe 8.3.4.)



Le tableau de transition d'état est le suivant :

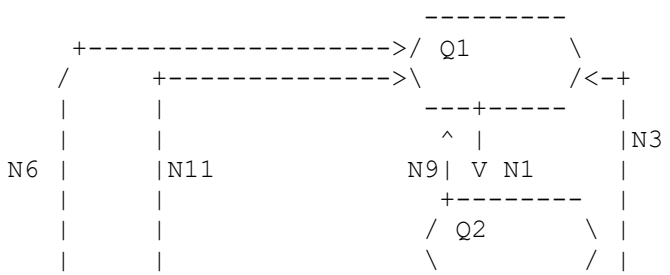
	Q1	Q3	Q4
Q1	-	N1	-
Q3	N3	-	N5
Q4	N6	N4	-

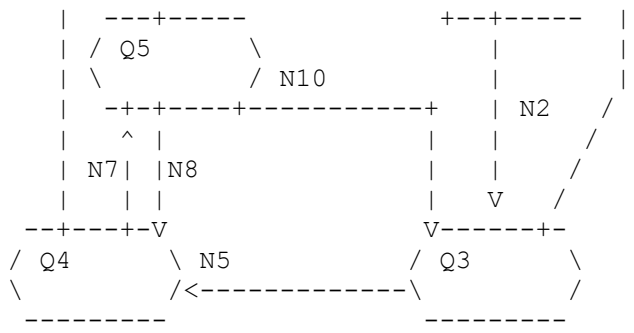
8.3.2 Diagramme d'état de session pour une cible

Noms symboliques des états :
Q1 : FREE
Q2 : ACTIVE
Q3 : LOGGED_IN
Q4 : FAILED
Q5 : IN_CONTINUE

L'état Q3 représente le fonctionnement de la phase de pleines caractéristiques de la session.

Le diagramme d'état est le suivant:





Le tableau de transition d'état est le suivant :

	Q1	Q2	Q3	Q4	Q5
Q1	-	N1	-	-	-
Q2	N9	-	N2	-	-
Q3	N3	-	-	N5	-
Q4	N6	-	-	-	N7
Q5	N11	-	N10	N8	-

8.3.3 Descriptions d'état pour initiateurs et cibles

Q1 : FREE
- initiateur : état à l'instanciation ou après nettoyage.
- cible : état à l'instanciation ou après nettoyage.

Q2 : ACTIVE
- initiateur : illégal.
- cible : La première connexion iSCSI dans la session est passée à IN_LOGIN, attendant qu'elle achève le processus d'établissement.

Q3 : LOGGED_IN
- initiateur : attente de tous événements de session.
- cible : attente de tous événements de session.

Q4 : FAILED
- initiateur : attente de récupération de session ou de continuation de session.
- cible : attente de récupération de session ou de continuation de session.

Q5 : IN_CONTINUE
- initiateur : illégal.
- cible : attente d'une tentative de continuation de session pour tirer une conclusion.

8.3.4 Descriptions des transitions d'état pour initiateurs et cibles

N1 :
- initiateur : Au moins une connexion de transport a atteint l'état LOGGED_IN.
- cible : la première connexion iSCSI dans la session a atteint l'état IN_LOGIN.

N2 :
- initiateur : illégal.
- cible : Au moins une connexion iSCSI a atteint l'état LOGGED_IN.

N3 :

- initiateur : Clôture en douceur de la session via la clôture de session (paragraphe 6.3.6).
- cible : Clôture en douceur de la session via la clôture de session (paragraphe 6.3.6) ou une réinitialisation de session réussie a cloturé proprement la session.

N4 :
- initiateur : Une tentative de continuation de session a réussi.
- cible : illégal.

N5 :
- initiateur : Une défaillance de session (paragraphe 6.3.6) s'est produite,
- cible : Une défaillance de session (paragraphe 6.3.6) s'est produite,

N6 :
- initiateur : la fin de temporisation d'état de session s'est produite, ou une réinitialisation de session a purgé cette instance de session. Il en résulte la libération de toutes les ressources associées, et l'état de session est éliminé.
- cible : la fin de temporisation d'état de session s'est produite, ou une réinitialisation de session a purgé cette instance de session. Il en résulte la libération de toutes les ressources associées, et l'état de session est éliminé.

N7 :
- initiateur : illégal.
- cible : Une tentative de continuation de session a été initiée.

N8 :
- initiateur : illégal.
- cible : la dernière tentative de continuation de session a échoué.

N9 :
- initiateur : illégal.
- cible : la tentative d'établissement sur la connexion de tête a échoué.

N10 :
- initiateur : illégal.
- cible : une tentative de continuation de session a réussi.

N11:
- initiateur : illégal.
- cible : une réinitialisation de session réussie a fermé proprement la session.

9. Considérations sur la sécurité

Historiquement, les systèmes natifs de mémorisation n'avaient pas à tenir compte de la sécurité, parce que leurs environnements présentaient des risques minimaux pour la sécurité. C'est-à-dire que ces environnements consistaient en appareils de mémorisation directement rattachés aux hôtes ou connectés via un réseau de zone de mémorisation (SAN, *Storage Area Network*) distinctement séparé du réseau de communications. L'utilisation de protocoles de mémorisation, comme SCSI, sur les réseaux IP exige que les questions de sécurité soient prises en compte. Les mises en œuvre iSCSI doivent fournir des moyens de protection contre les attaques actives (par exemple, prétendre être une autre identité, insertion, suppression, modification, et répétition de message) et les attaques passives (par exemple, espionnage, analyse des données envoyées sur le réseau).

Bien que techniquement possible, iSCSI NE DEVRAIT PAS être configuré sans sécurité, en particulier authentification dans la bande; voir le paragraphe 9.2. iSCSI configuré sans sécurité devrait être confiné aux environnements clos qui ont des risques de sécurité très limités et bien contrôlés. La [RFC3723] spécifie les mécanismes qui doivent être utilisés afin d'atténuer les risques qui sont décrits dans ce document.

Les paragraphes qui suivent décrivent les mécanismes de sécurité fournis par une mise en œuvre iSCSI.

9.1 Méchanismes de sécurité iSCSI

Les entités impliquées dans la sécurité iSCSI sont l'initiateur, la cible, et les points d'extrémité de communication IP. On prévoit des scénarios iSCSI dans lesquels plusieurs initiateurs ou cibles partagent un seul point d'extrémité de communication. Pour s'accommoder de tels scénarios, iSCSI prend en charge deux mécanismes de sécurité séparés :

l'authentification dans la bande entre l'initiateur et la cible au niveau de la connexion iSCSI (réalisé par l'échange des PDU iSCSI d'établissement de connexion) et la protection de paquet (intégrité, authentification, et confidentialité) par IPsec au niveau IP. Les deux mécanismes de sécurité se complètent l'un l'autre. L'authentification dans la bande assure la confiance de bout en bout (au moment de la connexion) entre l'initiateur iSCSI et la cible, tandis que IPsec fournit un canal sécurisé entre les points d'extrémité de la communication IP. iSCSI peut être utilisé pour accéder à des informations sensibles pour lesquelles une protection de sécurité significative est appropriée. Comme spécifié dans le reste de cette section de considérations sur la sécurité, la mise en œuvre de ces deux mécanismes de sécurité iSCSI est obligatoire (DOIT). L'utilisation de l'authentification dans la bande est fortement recommandée (DEVRAIT). À l'opposé, l'utilisation de IPsec est facultative (PEUT), car les risques pour la sécurité qu'il couvre peuvent n'être présents que sur un sous ensemble des réseaux utilisés par une connexion ou session iSCSI ; un exemple spécifique est quand une session iSCSI s'étend sur des centres de données, des passerelles de VPN IPsec aux frontières du centre de données pour protéger la connexité de WAN entre les centres de données peuvent être appropriées en combinaison avec l'authentification iSCSI dans la bande.

On trouvera plus de détails sur les scénarios typiques de iSCSI et les relations entre les initiateurs, les cibles, et les points d'extrémité de communication dans la [RFC3723].

9.2 Authentification dans la bande initiateur-cible

Durant l'établissement de connexion, la cible PEUT authentifier l'initiateur et l'initiateur PEUT authentifier la cible. L'authentification est effectuée sur chaque nouvelle connexion iSCSI par un échange de PDU d'établissement de connexion iSCSI en utilisant une méthode d'authentification négociée.

La méthode d'authentification ne peut pas supposer une protection IPsec sous-jacente parce que IPsec est d'utilisation facultative. Un attaquant devraient tirer aussi peu d'avantages que possible de l'inspection des PDU de phase d'authentification. Donc, une méthode qui utilise des mots de passe en clair (ou équivalents) NE DOIT PAS être utilisée ; d'un autre côté, la protection de l'identité n'est pas une exigence stricte.

Le mécanisme d'authentification protège contre la mémorisation de ressources par un établissement de connexion non autorisé en utilisant une fausse identité (usurpation d'identité). Une fois la phase d'authentification achevée, si le IPsec sous-jacent n'est pas utilisé, toutes les PDU sont envoyées et reçues en clair. Le mécanisme d'authentification smul (sans Ipsec sous-jacent) ne devrait être utilisé que lorsque il n'y a pas de risque d'espionnage ou d'insertion, suppression, modification, et répétition de message.

La Section 12 définit plusieurs méthodes d'authentification et les étapes exactes qui doivent être suivies dans chacune d'elles, incluant les clés de texte iSCSI et les valeurs permises à chaque étape. Chaque fois qu'un initiateur iSCSI obtient une réponse dont les clés, ou leurs valeurs, ne s'accordent pas à la définition d'étape, il DOIT interrompre la connexion.

Chaque fois qu'une cible iSCSI obtient une demande ou réponse dont les clés, ou leurs valeurs, ne s'accordent pas à la définition d'étape, elle DOIT répondre avec un rejet d'établissement avec l'état "Erreur d'initiateur" ou "Paramètre manquant". Ces états ne sont pas destinés à des valeurs cryptographiquement incorrectes comme la réponse CHAP, pour laquelle l'état "Échec d'authentification" DOIT être spécifié. L'importance de cette règle peut être illustrée dans CHAP avec l'authentification de cible (voir le paragraphe 12.1.3) où l'initiateur devra être capable de conduire une attaque en réflexion en omettant sa clé de réponse (CHAP_R), utilisant le même défi CHAP que la cible et reflétant la réponse de la cible à la cible. Dans CHAP, ceci est empêché parce que la cible doit répondre à la clé CHAP_R manquante par un rejet d'établissement avec l'état "Paramètre manquant".

Pour certaines des méthodes d'authentification, une clé spécifie l'identité de l'initiateur ou cible iSCSI pour les besoins de l'authentification. La valeur associée à cette clé PEUT être différente du nom iSCSI et DEVRAIT être configurable (CHAP_N : voir le paragraphe 12.1.3 ; SRP_U : voir le paragraphe 12.1.2). Pour cette raison, les mises en œuvre iSCSI DEVRAIENT gérer l'authentification d'une façon qui rende impossible l'usurpation d'identité à travers les noms iSCSI via ces identités d'authentification. Précisément, les mises en œuvre DEVRAIENT permettre la configuration d'une identité d'authentification pour un nom s'il est différent, et des accréditifs d'authentification pour cette identité. Durant le temps d'établissement, les mises en œuvre DEVRAIENT vérifier la relation entre le nom et l'identité en plus de l'authentification de l'identité par la méthode d'authentification négociée.

Quand une session iSCSI a plusieurs connexions TCP, concurremment ou à la suite, la méthode d'authentification et les identités ne devraient pas varier entre les connexions. Donc, toutes les connexions d'une session iSCSI DEVRAIENT utiliser la même méthode d'authentification, nom iSCSI, et identité d'authentification (pour les méthodes d'authentification qui utilisent une identité d'authentification). Les mises en œuvre DEVRAIENT vérifier cela et causer un échec d'authentification sur une nouvelle connexion qui utilise une méthode d'authentification, un nom iSCSI, ou une identité d'authentification, différents de ceux déjà utilisés dans la session. De plus, les mises en œuvre NE DEVRAIENT PAS

prendre en charge des connexions TCP à la fois authentifiées et non authentifiées dans la même session iSCSI, ajoutées concurremment ou à la suite à la session.

9.2.1 Considérations sur CHAP

Les initiateurs et les cibles iSCSI conformes DOIVENT mettre en œuvre la méthode d'authentification CHAP [RFC1994] (conformément au paragraphe 12.1.3, incluant l'option d'authentification de la cible).

Quand CHAP est effectué sur un canal non chiffré, il est vulnérable à une attaque de dictionnaire hors ligne. Les mises en œuvre DOIVENT prendre en charge l'utilisation de secrets CHAP ayant jusqu'à 128 bits aléatoires, incluant les moyens de générer de tels secrets et de les accepter d'une source de génération externe. Les mises en œuvre NE DOIVENT PAS fournir de moyens de génération (ou expansion) de secret autres qu'aléatoires.

Une entité administrative d'un environnement dans lequel CHAP est utilisé avec un secret qui a moins de 96 bits d'aléa DOIT appliquer le chiffrement IPsec (conformément aux exigences de mise en œuvre du paragraphe 9.3.2) pour protéger la connexion. De plus, dans ce cas, l'authentification IKE avec des clés de chiffrement de groupe pré-partagées NE DEVRAIT PAS être utilisée sauf si il n'est pas essentiel de protéger les membres du groupe contre des attaques de dictionnaire hors ligne par les autres membres.

Les secrets CHAP DOIVENT faire un nombre entier d'octets. Une mise en œuvre conforme NE DEVRAIT PAS continuer une étape d'établissement dans laquelle elle devrait envoyer une réponse CHAP (CHAP_R; voir le paragraphe 12.1.3) sauf si elle peut vérifier que le secret CHAP fait au moins 96 bits ou que le chiffrement IPsec est utilisé pour protéger la connexion.

Un secret CHAP utilisé pour l'authentification de l'initiateur NE DOIT PAS être configuré pour l'authentification d'une cible, et un secret CHAP utilisé pour l'authentification de la cible NE DOIT PAS être configuré pour l'authentification d'un initiateur. Si la réponse CHAP reçue par une extrémité d'une connexion iSCSI est la même que la réponse CHAP que le point d'extrémité receveur aurait généré pour le même défi CHAP, la réponse DOIT être traitée comme un échec d'authentification et causer la fermeture de la connexion (cela assure que le même secret CHAP n'est pas utilisé pour l'authentification dans les deux directions). Aussi, si une mise en œuvre iSCSI peut fonctionner à la fois comme initiateur et comme cible, des secrets et identités CHAP différents DOIVENT être configurés pour ces deux rôles. Ce qui suit est un exemple des attaques empêchées par ces exigences :

a) "Rogue" veut se faire passer pour "Storage" auprès d'Alice et sait qu'un seul secret est utilisé pour les deux directions de l'authentification de Storage-Alice.
b) Rogue convainc Alice d'ouvrir deux connexions avec lui et s'identifie comme Storage sur les deux connexions.
c) Rogue produit un défi CHAP sur la connexion 1, attend qu'Alice réponde, et ensuite reflète le défi d'Alice comme défi initial à Alice sur la connexion 2.
d) Si Alice ne vérifie pas la reflexion à travers les connexions, la réponse d'Alice sur la connexion 2 permet à Rogue de se faire passer pour Storage sur la connexion 1, même si Rogue ne connaît pas le secret CHAP de Alice-Storage.

Les générateurs NE DOIVENT PAS réutiliser le défi CHAP envoyé par le répondant pour l'autre direction d'une authentification bidirectionnelle. Les répondants DOIVENT vérifier cette condition et clore la connexion TCP iSCSI si elle se produit.

Le même secret CHAP NE DEVRAIT PAS être configuré pour l'authentification de plusieurs initiateurs ou cibles, car cela leur permet de se faire passer pour n'importe lequel d'entre eux, et compromettre l'un d'eux permet à l'attaquant de se faire passer pour n'importe lequel d'entre eux. Il est recommandé que les mises en œuvre iSCSI vérifient l'utilisation de secrets CHAP identiques par des homologues différents quand cette vérifiation est faisable et prennent des mesures appropriées pour avertir les utilisateurs et/ou administrateurs quand ceci est détecté.

Quand un initiateur ou cible iSCSI s'authentifie à ses contreparties dans plusieurs domaines administratifs, il DEVRAIT utiliser un secret CHAP différent pour chaque domaine administratif pour éviter de propager une sécurité compromise à travers les domaines.

Au sein d'un seul domaine administratif :

- Un seul secret CHAP PEUT être utilisé pour l'authentification d'un initiateur à plusieurs cibles.
- Un seul secret CHAP PEUT être utilisé pour l'authentification d'une cible à plusieurs initiateurs quand l'initiateur utilise un serveur externe (par exemple, RADIUS [RFC2865]) pour vérifier les réponses CHAP de la cible et ne connaît pas le secret CHAP de la cible.

Si on utilise pas un serveur externe de vérification des réponses (par exemple, RADIUS) employer un seul secret CHAP pour l'authentification d'une cible auprès de plusieurs initiateurs exige que de tels initiateurs connaissent le secret de cette

cible. Un de ces initiateurs peut se faire passer pour la cible auprès de tous les autres de ces initiateurs et compromettre un tel initiateur permet à un attaquant de se faire passer pour la cible auprès de tous ces initiateurs. Les cibles DEVRAIENT utiliser des secrets CHAP séparés pour l'authentification de chaque initiateur quand de tels risques sont un problème ; dans cette situation, il peut être utile de configurer une cible iSCSI logique séparée avec son propre nom de nœud iSCSI pour chaque initiateur ou groupe d'initiateurs parmi lesquels une telle séparation est désirée.

Les exigences ci-dessus renforcent les propriétés de sécurité de l'authentification CHAP pour iSCSI par comparaison avec le mécanisme de base d'authentification CHAP [RFC1994]. Il est très important de respecter ces exigences, en particulier les exigences de forts secrets CHAP (générés au hasard) car les mises en œuvre et déploiements iSCSI qui manquent à utiliser de forts secrets CHAP sont très probablements vulnérables aux attaques de dictionnaire hors ligne sur les secrets CHAP.

Le remplacement de CHAP par un meilleur mécanisme d'authentification est prévu dans une future version de iSCSI. La norme FC-SP-2 [FC-SP-2] a spécifié le protocole extensible d'authentification – mécanisme d'authentification par clé pré partagée généralisée (EAP-GPSK) [RFC5433] comme solution de remplacement à (et possible remplacement futur de) un usage similaire du canal fibre d'un CHAP renforcé. Un autre remplacement possible pour CHAP est un mécanisme de mot de passe sécurisé, par exemple, une version mise à jour du mécanisme actuel d'authentification SRP de iSCSI.

9.2.2 Considérations sur SRP

La force de la méthode d'authentification SRP (spécifiée dans la [RFC2945]) dépend des caractèristiques du groupe utilisé (c'est-à-dire, le nombre premier modulo N et le générateur g). Comme décrit dans la [RFC2945], N doit être un nombre premier Sophie Germain (de la forme N = 2q + 1, où q est aussi premier) et le générateur g est une racine primitive de GF(N). Dans l'authentification iSCSI, le nombre premier modulo N DOIT être d'au moins 768 bits.

La liste des groupes SRP admis est fournie dans la [RFC3723].

9.2.3 Considérations sur Kerberos

iSCSI utilise Kerberos V5 [RFC4120] brut pour authentifier un client (initiateur iSCSI) principal à un service (cible iSCSI) principal. Noter que iSCSI n'utilise pas l'interface de programme d'application de service de sécurité générique (GSS-API, *Generic Security Service Application Program Interface*) [RFC2743] ni le mécanisme de sécurité Kerberos V5 GSS-API [RFC4121]. Cela signifie que les mises en œuvre iSCSI qui prennent en charge la KRB5 AuthMethod (paragraphe 12.1) sont directement impliquées dans le protocole Kerberos. Quand Kerberos V5 est utilisé pour l'authentification, les actions suivantes DOIVENT être effectuées comme spécifié dans la [RFC4120]:

- La cible DOIT valider KRB_AP_REQ pour s'assurer que l'initiateur est de confiance.
- Quand l'authentification mutuelle est choisie, l'initiateur DOIT valider KRB_AP_REP pour déterminer le résultat de l'authentification mutuelle.

Comme Kerberos V5 est capable de fournir l'authentification mutuelle, les mises en œuvre DEVRAIENT prendre en charge l'authentification mutuelle par défaut pour l'authentification à l'établissement de connexion.

Noter, cependant, que l'authentification Kerberos assure seulement que le serveur (cible iSCSI) est de confiance pour le client Kerberos (initiateur) et vice versa ; un initiateur devrait employer des techniques appropriées de découverte de service sécurisées (par exemple, iSNS ; voir le paragraphe 4.2.7) pour s'assurer qu'il parle à la cible principale prévue.

iSCSI n'utilise pas Kerberos v5 pour la protection de l'intégrité ou de la confidentialité du protocole iSCSI. iSCSI utilise IPsec à cette fin comme spécifié au paragraphe 9.3.

9.3 IPsec

iSCSI utilise le mécanisme IPsec pour la protection des paquets (intégrité, authentification, et confidentialité cryptographique) au niveau IP entre les points d'extrémité de communication iSCSI. Les paragraphes qui suivent décrivent les protocoles IPsec qui doivent être mis en œuvre pour l'authentification et l'intégrité des données; la confidentialité; et la gestion des clés de chiffrement.

Un initiateur ou cible iSCSI peut fournir le support IPsec requis pleinement intégré ou en conjonction avec un appareil frontal IPsec. Dans ce dernier cas, les exigences de conformité à l'égard de la prise en charge de IPsec s'appliquent à "l'appareil combiné". Seul cet "appareil combiné" est à considérer comme un appareil iSCSI.

Les considérations et recommandations détaillées pour l'utilisation d'IPsec pour iSCSI sont fournies dans la [RFC3723] mise à jour par la [RFC7146]. Les exigences IPsec sont reproduites ici par facilité et sont destinées à correspondre à celles de la [RFC7146] ; en cas de désaccord, les exigences de la [RFC7146] s'appliquent.

9.3.1 Authentification et intégrité des données

L'authentification et la protection de l'intégrité des données sont fournies par un code d'authentification de message cryptographique dans chaque paquet envoyé. Ce code protège contre l'insertion, la suppression, et la modification du message. La protection contre la répétition de message est réalisée en utilisant un compteur de séquence.

Un initiateur ou cible conforme à iSCSI DOIT fournir l'authentification et la protection de l'intégrité des données en mettant en œuvre IPsec v2 [RFC2401] avec ESPv2 [RFC2406] en mode tunnel, DEVRAIT fournir l'authentification et la protection de l'intégrité des données en mettant en œuvre IPsec v3 [RFC4301] avec ESPv3 [RFC4303] en mode tunnel, et PEUT fournir l'authentification et la protection de l'intégrité des données en mettant en œuvre soit IPsec v2, soit v3 avec la version appropriée de ESP en mode transport mode. La mise en œuvre IPsec DOIT satisfaire les exigences spécifiques de iSCSI suivantes :

- HMAC-SHA1 DOIT être mis en œuvre sous la forme spécifique de HMAC-SHA-1-96 [RFC2404].
- AES CBC MAC avec extensions XCBC utilisant des clés de 128 bits DEVRAIT être mis en œuvre [RFC3566].
- Les mises en œuvre qui prennent en charge IKEv2 [RFC5996] DEVRAIENT aussi mettre en œuvre AES à code d'authentification de message de Galois (GMAC, *Galois Message Authentication Code*) [RFC4543] en utilisant des clés de 128 bits.

Le service anti répétition ESP DOIT aussi être mis en œuvre.

Aux grandes vitesses auxquelles on s'attend que iSCSI opère, une seule SA IPsec pourrait rapidement épuiser l'espace de numéros de séquence à 32 bits de ESP, exigeant de fréquents changements de clés de la SA, car le retour à zéro du numéro de séquence ESP au sein d'une seule SA est interdit aussi bien pour ESPv2 [RFC2406] que pour ESPv3 [RFC4303]. Afin de fournir les moyens d'éviter ces fréquents changements de clés potentiellement indésirables, les mises en œuvre qui sont capables de fonctionner à des vitesses de 1 gigabit/s ou plus DOIVENT mettre en œuvre les numéros de séquence étendus (à 64 bits) pour ESPv2 (et ESPv3, si c'est pris en charge) et DEVRAIENT utiliser les numéros de séquence étendus pour tout le trafic iSCSI. La négociation de numéros de séquence étendus au titre de l'établissement d'association de sécurié est spécifiée dans la [RFC4304] pour IKEv1 et dans la [RFC5996] pour IKEv2.

9.3.2 Confidentialité

La confidentialité est assurée en chiffrant les données dans chaque paquet. Quand la confidentialité est utilisée, elle DOIT être accompagnée par l'authentification et la protection de l'intégrité des données pour fournir une protection complète contre l'espionnage et l'insertion, la suppression, la modification, et la répétition de message.

Un initiateur ou cible conforme à iSCSI DOIT assurer la confidentialité en mettant en œuvre IPsec v2 [RFC2401] avec ESPv2 [RFC2406] en mode tunnel, DEVRAIT assurer la confidentialité en mettant en œuvre IPsec v3 [RFC4301] avec ESPv3 [RFC4303] en mode tunnel, et PEUT assurer la confidentialité en mettant en œuvre IPsec v2 ou v3 avec la version appropriée de ESP en mode transport, avec les exigences spécifiques de iSCSI suivantes qui s'appliquent à IPsec v2 et IPsec v3 :

- 3DES en mode CBC PEUT être mis en œuvre [RFC2451].
- AES en mode CBC avec des clés de 128 bits DOIT être mis en œuvre [RFC3602] ; d'autres tailles de clé PEUVENT être prises en charge.
- AES en mode compteur PEUT être mis en œuvre [RFC3686].
- Les mises en œuvre qui prennent en charge IKEv2 [RFC5996] DEVRAIENT aussi mettre en œuvre le mode AES Galois/Compteur (GCM) avec des clés de 128 bits [RFC4106] ; d'autres tailles de clé PEUVENT être prises en charge.

Due à sa faiblesse inhérente, DES en mode CBC NE DOIT PAS être utilisé.

L'algorithme de chiffrement NULL DOIT aussi être mis en œuvre.

9.3.3 Politique, associations de décurité, et gestion de clés de chiffrement

Une mise en œuvre iSCSI conforme DOIT satisfaire les exigences de gestion de clé de chiffrement de la suite de protocoles IPsec. L'authentification, la négociation d'association de sécurité, et la gestion de clé de chiffrement DOIVENT être fournies en mettant en œuvre IKE [RFC2409] utilisant le DOI IPsec [RFC2407] et DEVRAIENT être fournies par la mise en œuvre de IKEv2 [RFC5996], avec les exigences spécifiques de iSCSI suivantes :

a) L'authentification de l'homologue utilisant une clé de chiffrement pré partagée DOIT être prise en charge. L'authentification de l'homologue à l'aide de signatures numériques PEUT être prise en charge. Pour IKEv1 ([RFC2409]), l'authentification de l'homologue à l'aide de méthodes de chiffrement à clé publique précisées dans les paragraphes 5.2 et 5.3 de la [RFC2409] NE DEVRAIT PAS être utilisée.

b) Quand des signatures numériques sont utilisées pour réaliser l'authentification, un négociateur IKE DEVRAIT utiliser la charge utile de demande de certificat IKE pour spécifier l'autorité de certification. Les négotiateurs IKE DEVRAIENT vérifier la validité du certificat via la liste de révocation de certificat pertinente ou via l'utilisation du protocole d'état de certificat en ligne (OCSP, *Online Certificate Status Protocol*) [RFC6960] avant d'accepter un certificat PKI à utiliser dans les procédures d'authentification IKE. La prise en charge de OCSP au sein du protocole IKEv2 est spécifiée dans la [RFC4806]. Ces vérifications peuvent n'être pas nécessaires dans des environnements où un petit nombre de certificats sont configurés statiquement comme ancres de confiance.

c) Les mises en œuvre conformes iSCSI de IKEv1 DOIVENT prendre en charge le mode pricipal et DEVRAIENT prendre en charge le mode aggressif. Le mode principal NE DEVRAIT PAS être utilisé avec une méthode d'authentification à clé pré partagée quand l'initiateur ou la cible utilise des adresses allouées de façon dynamique. Bien que dans certains cas des clés pré partagées offrent une bonne sécurité, les situations dans lesquelles les adresses allouées de façon dynamique sont utilisées obligent à faire appel à une clé pré partagée de groupe, ce qui crée une vulnérabilité aux attaques par interposition.

d) Dans le mode rapide de IKEv1 phase 2, dans les échanges pour créer la SA de phase 2, la charge utile d'identification DOIT être présentz.

e) Les exigences de type d'identification suivantes s'appliquent à IKEv1 : les types d'identification ID_IPV4_ADDR, ID_IPV6_ADDR (si la pile de protocoles prend en charge IPv6) et ID_FQDN DOIVENT être pris en charge ; ID_USER_FQDN DEVRAIT être pris en charge. Les types d'identification de sous réseau IP, de gamme d'adresses IP, ID_DER_ASN1_DN, et ID_DER_ASN1_GN NE DEVRAIENT PAS être utilisés. Le type d'identification ID_KEY_ID NE DOIT PAS être utilisé.

f) Si IKEv2 est pris en charge, les exigences d'identification suivantes s'appliquent : les types d'identification ID_IPV4_ADDR, ID_IPV6_ADDR (si la pile de protocoles prend en charge IPv6), et ID_FQDN DOIVENT être pris en charge ; ID_RFC822_ADDR DEVRAIT être pris en charge. Les types d'identification ID_DER_ASN1_DN et ID_DER_ASN1_GN NE DEVRAIENT PAS être utilisés. Le type d'identification ID_KEY_ID NE DOIT PAS être utilisé.

Les raisons du "NE DOIT PAS" et du "NE DEVRAIT PAS" pour les exigences de type d'identification dans les alinéas e) et f) sont :
- sous réseau IP et gamme d'adresse IP sont trop larges pour identifier utilement un point d'extrémité iSCSI,
- les types DN et GN sont des identités X.500 ; il est généralement préférable d'utiliser une identité provenant du subjectAltName d'un certificat PKI,
- ID_KEY_ID n'est pas interopérable tel que spécifié.

La gestion des clés cryptographiques manuelle NE DOIT PAS être utilisée, parce qu'elle n'assure pas kes changements de clés nécessaires.

Lorsque des groupes Diffie-Hellman (DH) sont utilisés, un groupe DH d'au moins 2048 bits DEVRAIT être offert au titre de toute proposition de créer des associations de sécurité IPsec pour protéger le trafic iSCSI, avec IKEv1 et IKEv2.

Quand IPsec est utilisé, la réception d'un message Supprimer IKEv1 phase 2 ou d'un message IKEv2 Échange d'information qui supprime la SA NE DEVRAIT PAS être interprétée comme une raison pour supprimer la connexion TCP iSCSI. Si du trafic supplémentaire est envoyée sur elle, une nouvelle SA IKE sera créée pour la protéger.

La méthode utilisée par l'initiateur pour déterminer si la cible devrait être connectée en utilisant IPsec est considérée comme un problème d'administration de politique IPsec et donc non défini dans la norme iSCSI.

La méthode utilisée par un initiateur qui prend en charge aussi bien IPsec v2 que v3 pour déterminer quelles versions de IPsec sont supportées par la cible est aussi considérée comme un problème d'administration de politique IPsec et donc non définie dans la norme iSCSI. Si IPsec v2 et v3 sont tous deux supportés par l'initiateur et la cible, l'utilisation de IPsec v3 est recommandée.

Si une cible iSCSI est découverte via une demande SendTargets dans une session Découverte qui n'utilise pas IPsec, l'initiateur devraient supposer qu'il n'a pas besoin de IPsec pour établir une session avec cette cible. Si une cible iSCSI est

découverte en utilisant une session Découverte qui utilise IPsec, l'initiateur DEVRAIT utiliser IPsec quand il établit une session avec cette cible.

9.4 Considérations de sécurité sur la clé X#NodeArchitecture

Les considérations de sécurité de ce paragraphe sont spécifiques de X#NodeArchitecture discuté au paragraphe 13.26.

Cette clé d'extension transmet des détails spécifiques de la mise en œuvre sur le nœud qui l'envoie ; de tels détails peuvent être considérés comme sensibles dans certains environnements. Par exemple, si une certaine version d'un logiciel est connue pour contenir des faiblesses de sécurité, annoncer la présence de cette version via cette clé peut n'être pas désirable. Les contre mesures pour ce problème de sécurité sont :

a) envoyer des informations moins détaillées dans les valeurs de clés,
b) ne pas envoyer la clé d'extension,
c) utiliser IPsec ([RFC4303]) pour assurer la confidentialité pour la connexion iSCSI sur laquelle la clé est envoyée.

Pour prendre en charge la première et la seconde des contre mesures, toutes les mises en œuvre de cette clé d'extension DOIVENT fournir un mécanisme administratif pour désactiver l'envoi de la clé. De plus, toutes les mises en œuvre DEVRAIENT fournir un mécanisme administratif pour configurer un niveau de verbosité de la valeur de la clé, contrôlant par là la quantité d'informations envoyées.

Par exemple, un niveau de verbosité inférieur peut activer la seule transmission des noms des composants de l'architecture de nœuds, mais pas les numéros de version. Le choix de la contre mesure la plus appropriée dépend de l'environnement. Cependant, envoyer des informations moins détaillées dans les valeurs de clé peut être une contre mesure acceptable dans de nombreux environnements, car cela fait un compromis entre l'envoi de trop d'informations et l'autre contre mesure plus complète de ne pas envoyer de clé du tout ou d'utiliser IPsec.

En plus des considérations de sécurité qui impliquent la transmission du contenu de la clé, toute méthode d'enregistrement utilisée pour les valeurs de clé DOIT garder les informations à l'abri des intrus. Pour toutes les mises en œuvre, les exigences pour traiter ces problèmes de sécurité sont les suivants :

a) L'affichage de l'enregistrement NE DOIT être possible qu'avec les droits administratifs sur le nœud.
b) Les options pour désactiver l'enregistrement sur le disque et garder les enregistrements pour une durée fixée DEVRAIENT être fournies.

Finalement, il est important de noter que des nœuds différents peuvent avoir des niveaux de risque différents, et ces différences peuvent affecter la mise en œuvre. Les composantes du risque incluent des biens, des menaces, et des vulnérabilités. Considérons l'exemple de nœuds iSCSI suivants, qui montre les diférences de biens et vulnérabilités des nœuds, et, par suite, des différences de mise en œuvre :

a) Une cible iSCSI fondée sur un système d'exploitation spécial : comme la cible iSCSI contrôle l'accès à la mémorisation de données qui contient les biens de la compagnie, le niveau des biens paraît très élevé. Aussi, à cause du système d'exploitation spécial, dans lequel les vulnérabilités sont moins bien connues, le niveau de vulnérabilé paraît très bas.
b) Plusieurs initiateurs iSCSI dans une ferme d'herbages, ayant chacun un système d'exploitation général : le niveau des biens de chaque nœud est vu comme bas, car les herbages sont remplaçables et de faible coût. Cependant, le niveau de vulnérabilité est vu comme élevé, car il peut y avoir de nombreuses vulnérabilités de ce système d'exploitation généraliste. Pour cette cible, une mise en œuvre appropriée pourrait être d'enregistrer les valeurs de clés reçues mais de ne pas transmettre la clé. Pour cet initiateur, une mise en œuvre appropriée pourrait être la transmission de la clé mais de ne pas enregistrer les valeurs de clé reçues.

9.5 Considérations sur le contrôle d'accès SCSI

iSCSI est un protocole de transport SCSI et à ce titre n'applique aucun contrôle d'accès sur les opérations de niveau SCSI comme les fonctions de gestion des tâches SCSI (par exemple, réinitialisation de LU ; voir le paragraphe 11.5.1). Les contrôles d'accès de niveau SCSI (par exemple, ACCESS CONTROL OUT ; voir [SPC3]) doivent être déployés de façon appropriée en pratique pour traiter les considérations de sécurité de niveau SCSI, en plus des mécanismes de sécurité via une connexion iSCSI et de protection de paquer qui ont déjà été discutées dans les paragraphes précédents.

10. Notes de mise en œuvre

Cette section note des considérations de performances et de fiabilité du protocole iSCSI. Ce protocole a été conçu pour permettre des mises en œuvre efficaces des matériels et logiciels. Le mécanisme d'étiquettes de tâches d'iSCSI a été conçu

pour permettre le placement de données direct (DDP, *Direct Data Placement*) une forme de DMA au niveau iSCSI ou inférieur.

L'hypothèse qui a guidé la conception de ce protocole est que les cibles sont resteintes en ressources par rapport aux initiateurs.

Il est aussi conseillé aux mises en œuvre de considérer les conséquences du modèle de transposition de iSCSI en SCSI soulignées au paragraphe 4.4.3.

10.1 Adaptateurs multi réseaux

Le protocole iSCSI permet plusieurs connexions, dont toutes n'ont pas besoin de se faire sur le même adapteur réseau. Si plusieurs connexions réseau doivent être utilisées avec un support de matériel, l'allégeance à l'état des données de commande du protocole SCSI d'une connexion TCP assure qu'il n'est pas besoin de dupliquer les informations à travers les adaptateurs réseau ou qu'ils coopèrent d'autre façon.

Cependant, certaines commandes de gestion de tâches peuvent exiger certaines formes lâches de coopération ou de duplication au moins sur la cible.

10.1.1 Réutilisation prudente des ISID

Historiquement, le modèle SCSI (et les mises en œuvre et applications fondées sur ce modèle) a supposé que les accès SCSI sont des entités physiques statiques. De récentes extensions au modèle SCSI ont tiré parti des noms uniques au monde persistents pour ces accès. Dans iSCSI, cependant, les accès d'initiateur SCSI sont les points d'extrémité créés de façon dynamique, de sorte que les présomptions de "statique et physique" ne s'appliquent pas. Dans tous les cas, la section "modèles" (en particulier, le paragraphe 4.4.1) fournit des noms persistants, réutilisables pour les accès SCSI de type iSCSI même quand il n'y a pas besoin d'un lien physique de l'entité à ces noms.

Pour à la fois minimiser l'interruption des applications traditionnelles et faciliter les caractéristiques qui s'appuient sur des noms persistants pour les accès SCSI, les mises en œuvre iSCSI DEVRAIENT tenter de fournir une présentation stable des accès d'initiateur SCSI (aux couches supérieures d'OS et aux cibles auxquelles elles se connectent). Cela peut être réalisé dans une mise en œuvre d'initiateur par une réutilisation prudente des ISID. En d'autres termes, le même ISID devrait être utilisé dans le processus d'établissement de connexion à plusieurs groupes de portails cibles (de la même cible iSCSI ou de cibles iSCSI différentes). La règle ISID (paragraphe 4.4.3) interdit seulement la réutilisation sur le même groupe de portails cibles. Elle n'empêche pas la réutilisation sur d'autres groupes de portails cibles. Le principe de réutilisation prudente "encourage" à la réutilisation sur d'autres groupes de portails cibles. Quand un appareil cible SCSI voit la même paire (Nom d'initiateur, ISID) dans différentes sessions sur des groupes portails cibles différents, il peut identifier l'accès d'initiateur SCSI sous-jacent sur chaque session comme étant le même accès SCSI. En effet, il peut reconnaître plusieurs chemins à partir de la même source.

10.1.2 Utilisation du nom iSCSI, d'ISID, et TPGT

Les concepteurs du protocole iSCSI savent que les transports SCSI traditionnels s'appuient sur l'identité de l'initiateur pour allouer l'accès à des ressources de mémorisation. Bien que de nouvelles techniques qui simplifient le contrôle d'accès soient disponibles, la prise en charge de schémas de configuration et d'authentification qui se fondent sur l'identité de l'initiateur est réputée importante pour la prise en charge des systèmes traditionnels et de logiciels d'administration. iSCSI prend donc en charge la notion qu'il devrait être possible d'allouer l'accès à des ressources de mémorisation sur la base de l'identité de "l'appareil initiateur".

Lorsque il y a plusieurs composants de matériel ou logiciel coordonnés sur un seul nœud iSCSI, il doit y avoir une entité (logique) qui représente le nœud iSCSI qui rend le nom de nœud iSCSI disponible à tous les composants impliqués dans la création de session et connexion. De façon similaire, cette entité qui représente le nœud iSCSI doit être capable de coordonner les ressources d'identifiant de session (le ISID pour les initiateurs) pour appliquer la règle ISID et la règle TSIH (voir le paragraphe 4.4.3).

Pour les cibles, à cause de l'environnement clos, la mise en œuvre de cette entité devrait être directe. Cependant, les fabricants de matériel iSCSI (par exemple, NIC ou HBA) ont prévu que les cibles DEVRAIENT fournir des mécanismes pour la configuration du nom de nœud iSCSI à travers les groupes portails instanciés par plusieurs instances de ces composants au sein d'une cible.

Cependant, les cibles complexes qui utilisent plusieurs étiquettes de portail cible peuvent les reconfigurer pour réaliser divers objectifs de qualité. Les initiateurs ont deux mécanismes à leur disposition pour découvrir et/ou vérifier la reconfiguration des cibles – le type de session Discovery et une clé retournée par la cible durant l'établissement pour

confirmer le TPGT. Un initiateur devrait tenter de "redécouvrir" la configuration de la cible chaque fois qu'une session se termine de façon inattendue.

Pour les initiateurs, à long terme, il est prévu que les fabricants de système d'exploitation s'occupent du rôle de cette entité et fournissent des API standard qui puissent informer les composants de leur nom de nœud iSCSI et puissent configurer et/ou coordonner l'allocation, l'utilisation, et la réutilisation d'ISID.

Reconnaissant que de telles API d'initiateur ne sont pas disponibles aujourd'hui, d'autres mises en œuvre du rôle de cette entité sont possibles. Par exemple, un humain peut instancier le nom de nœud (commun) au titre du processus d'installation de chaque composant iSCSI impliqué dans la création et l'établissement de session. Cela peut être fait en pointant le composant sur une localisation spécifique de fabricant pour ces données ou sur une localisation à l'échelle du système. La structure de l'espace de noms d'ISID (voir le paragraphe 11.12.5 et la [RFC3721]) facilite la mise en œuvre de la coordination d'ISID en permettant à chaque fabricant de composant de coordonner indépendamment (des autres composants de fabricant) l'allocation, l'utilisation, et la réutilisation de sa propre partition de l'espace de noms d'ISID d'une manière spécifique du fabricant. Le partitionnement de l'espace de noms d'ISID au sein de groupes portails d'initiateur gérés par ce fabricant permet à chacun de ces groupes portails d'initiateur d'agir de façon indépendante de tous les autres groupes portails quand ils choisissent un ISID pour un établissement ; cela facilite l'application de la règle ISID (voir le paragraphe 4.4.3) chez l'initiateur.

Un fabricant de matériel iSCSI (par exemple, NIC ou HBA) dont l'utilisation est prévue chez les initiateurs DOIT mettre en œuvre un mécanisme pour configurer le nom de nœud iSCSI. Les fabricants et les administrateurs doivent s'asurer que les noms de nœuds iSCSI sont uniques au monde. Il est donc important que quand on choisit de réutiliser le nom de nœud iSCSI d'une unité désactivée on ne réalloue pas ce nom à l'unité originale sauf si son unicité peut à nouveau être certifiée.

De plus, un fabricant de matériel iSCSI doit mettre en œuvre un mécanisme pour configurer et/ou coordonner les ISID pour toutes les sessions gérées par plusieurs instances de ce matériel au sein d'un certain nœud iSCSI. Une telle configuration peut être préallouée de façon permanente en usine (d'une façon nécessairement unique au monde) allouée de façon statique (par exemple, partitionnée à travers tous les NIC à l'initialisation d'une façon localement unique) ou allouée dynamiquement (par exemple, allocation en ligne, aussi de façon unique en local). Dans les deux dernier cas, la configuration peut être via des API publiques (peut-être avec un logiciel indépendant du fabricant, comme le fabricant du système d'exploitation) ou des API privées pilotées par le propre logiciel du fabricant.

Le processus d'allocation et de coordination de nom doit être aussi englobant et automatisé que possible, car des années d'utilisation traditionnellle ont montré qu'il est très enclin à l'erreur. Il devrait être mentionné que le SCSI a des schémas de remplacement pour le contrôle d'accès qui peuvent être utilisés par tous les transports, et leur sécurité ne dépend pas d'une stricte coordination des désignations

10.2 Autosense et allégeance auto contingeante (ACA, Auto Contingent Allegiance)

"Autosense" se réfère au retour automatique des données de sens à l'initiateur dans les cas où une commande ne s'achève pas sur un succès. Les initiateurs et les cibles iSCSI DOIVENT prendre en charge et utiliser Autosense.

ACA aide à préserver l'ordre d'exécution des commandes en présence d'erreurs. Comme il peut y avoir de nombreuses commandes en cours entre un initiateur et une cible, dans certains systèmes d'exploitation la fonctionnalité d'initiateur SCSI dépend de l'ACA pour mettre en application l'ordre d'exécution des commandes durant la récupération d'erreur, et donc les mises en œuvre d'initiateur iSCSI pour ces systèmes d'exploitation doivent prendre en charge ACA. Afin de prendre en charge la récupération d'erreur pour ces systèmes d'exploitation et initiateurs iSCSI, les cibles iSCSI DEVRAIENT prendre en charge ACA.

10.3 Temporisations iSCSI

Les actions de récupération iSCSI dépendent souvent de la reconnaissance des fins de temporisation iSCSI et des actions qui sont faites sur elles avant les fins de temporisation SCSI. Déterminer les bonnes temporisations à utiliser pour les diverses actions iSCSI (accusés de réception de commande attendus, accusés de réception d'état, etc.) dépend beaucoup de l'infrastructure (par exemple, matériel, liaisons, pile TCP/IP, pilote iSCSI). Comme guide, la mise en œuvre peut utiliser un délai moyen de passage NOP-Out/NOP-In multiplié par un "facteur de sécurité" (par exemple, 4) comme bonne estimation du délai de base de la pile iSCSI pour une certaine connexion. Le facteur de sécurité devrait tenir compte de la variabilité de la charge pour un réseau de mise en œuvre. Pour la suppression de connexion, la mise en œuvre peut vouloir aussi considérer les pratiques courantes de TCP pour l'infrastructure considérée.

Les négociations text PEUVENT aussi être soumises à des limites de temps ou de nombre d'échanges. Ces limites DEVRAIENT être assez généreuses pour éviter d'affecter l'interopérabilité (par exemple, en permettant que chaque clé soit négociée sur un échange séparé).

La relations entre fins de temporisation iSCSI et fins de temporisation SCSI devraient aussi être considérées. Les fins de temporisation SCSI devraient être plus longues que les fins de temporisation iSCSI plus le temps requis pour la récupération iSCSI chaque fois que la récupération iSCSI est prévue. Autrement, une mise en œuvre peut choisir de solidariser les fins de temporisation et récupérations iSCSI avec les fins de temporisation SCSI afin que la récupération SCSI ne devienne active que lorsque la récupération iSCSI n'est pas prévue ou a échoué.

La mise en œuvre peut aussi vouloir considérer l'interaction entre divers événement iSCSI exceptionnels -- comme un échec de résumé -- et les fins de temporisations suivantes. Quand la récupération d'erreur iSCSI est active, un échec de résumé va probablement résulter en la découverte d'une commande ou PDU de données manquante. Dans ces cas, une mise en œuvre peut vouloir diminuer les valeurs de temporisation pour permettre une initiation plus rapide des procédures de récupération.

10.4. Réessai de commande et nettoyage des instances de vieilles commandes

Pour éviter d'avoir l'apparition de vieilles instances de commandes réessayées dans une fenêtre de commande valide après le retour à zéro des numéros de séquence de commandes, le protocole exige (voir au paragraphe 4.2.2.1) que sur chaque connexion sur laquelle un réessai a été produit, une commande non immédiate soit produite et acquittée dans un intervalle de 2**31 - 1 commandes du numéro de séquence de commande de la commande réessayée. Cette exigence peut être satisfaite de plusieurs façons par une mise en œuvre.

La technique la plus simple est d'envoyer une commande SCS non immédiate (qui ne soit pas un réessai) (ou un NOP si aucune commande SCSI n'est disponible pour un moment) après chaque réessai de commande sur la connexion sur laquelle le réessai a été tenté. Parce que des erreurs sont supposées être des événements rares, cette technique est probablement la plus efficace, car elle n'implique pas de vérifications supplémentaires chez l'initiateur lors de la production de commandes.

10.5 Couche Sync et Steering et performances

Bien que la couche Sync et Steering soit facultative, un initiateur/cible chez qui elle ne fonctionne pas vis à vis d'une cible/initiateur qui demande sync et steering peut subir une dégradation de performances causée par le déclassement et la perte de paquets. Fournir un mécanisme de sync et steering est recommandé pour toutes les mises en œuvre à haut débit.

10.6 Considérations sur les appareils dépendants de l'état et les opérations SCSI de longue durée

Les appareils en accès séquentiel fonctionnent selon le principe que la position de l'appareil se fonde sur la dernière commande traitée. À ce titre, l'ordre de traitement des commandes, et la connaissance que la commande précédente a été traitée ou non, sont de la plus grande importance pour conserver l'intégrité des données. Par exemple, des réessais par inadvertence de commandes SCSI quand on ne sait pas si la commande SCSI précédente a été traitée est un risque potentiél pour l'intégrité des données.

Pour un appareil à accès séquentiel, considérons le scénario dans lequel une commande SCSI SPACE pour faire une espace arrière sur une marque de fichier est procuite et ensuite produite à nouveau sans avoir reçu d'état pour la commande. Si la première commande SPACE a en fait été traitée, la commande SPACE répétée, si elle est traitée, va causer le changement de la position. Donc, une opération d'écriture suivante va écrire des données au mauvais endroit, et toutes les données précédant cette position seront écrasées.

Pour un appareil à changement de support, considérons le scénario dans lequel une commande EXCHANGE MEDIUM (les adresses de source de de destination sont les mêmes, effectuant donc un échange) est produite, puis reproduite sans qu'un état soit reçu pour la commande. Si la première commande EXCHANGE MEDIUM a en fait été traitée, la deuxième commande EXCHANGE MEDIUM, si elle est traitée, va effectuer à nouveau l'échange. L'effet net est qu'aucun échange n'a été effectué, mettant donc en danger l'intégrité des données.

Toutes les commandes qui changent l'état de l'appareil (par exemple,les commandes SPACE pour les appareils à accès séquentiel et les commandes EXCHANGE MEDIUM pour les appareils à changement de support) DOIVENT être produites comme commandes non immédiates en vue d'une livraison déterministe et ordonnée aux cibles iSCSI.

Pour beaucoup de ces commamdes à changement d'état, le modèle d'éxécution suppose aussi que la commande est exécutée exactement une fois. Les appareils qui mettent en œuvre READ POSITION et LOCATE fournissent un moyen de récupération de commande au niveaur SCSI, et les nouveau appareils de classe ruban devraient prendre en charge ces

commandes. En leur absence, un réessai au niveau SCSI est difficile, et la récupération d'erreur est conseillée au niveau iSCSI.

Les appareils qui fonctionnent sur des sous systèmes à longs délais de livraison et effectuent des opérations SCSI qui durent longtemps peuvent avoir besoin de mécanismes qui permettent le remplacement de connexion alors que les commandes courent toujours (par exemple, durant une opération de copie importante).

10.6.1 Détermination du niveau de récupération d'erreur approprié

La mise en œuvre et l'utilisation d'un niveau de récupération d'erreur *(error recovery level)* spécifique devraient être déterminées sur la base de scénarios de déploiement d'une certaine mise en œuvre iSCSI. Généralement, les facteurs suivants doivent être considérés avant de décider du niveau de récupération approprié :

a) la résilience de l'application aux défaillances d'entrée/sortie,
b) le niveau requis de disponibilité en face de défaillances de connexion de transport,
c) la probabilité d'un "échappement de somme de contrôle" de couche transport (message d'erreur non détectée par la somme de contrôle TCP – voir dans la [RFC3385] la discussion qui s'y rapporte). Ceci à son tour décide de la fréquence de défaillance de résumé iSCSI et donc de la criticité de la récupération d'erreur de niveau iSCSI. Les détails de l'estimation de cette probabilité sortent du domaine d'application de ce document.

La considération de ces facteurs par exemple pour un appareil SCSI suggère que les mises en œuvre DEVRAIENT utiliser Niveau de récupération d'erreur=1 quand une défaillance de connexion de transport n'est pas un problème et que la récupération de niveau SCSI n'est pas disponible, et Niveau de récupération d'erreur=2 quand il y a une forte probabilité de défaillance de connexion durant une sauvegarde/restitution.

Pour les opérations de copie importantes, les mises en œuvre DEVRAIENT utiliser Niveau de récupération d'erreur=2 chaque fois qu'il y a une probabilité relativement importante de défaillance de connexion.

10.7 Considérations sur la mise en œuvre de l'interruption multi tâches

Les opérations d'interruption multi tâches sont normalement produites en cas d'urgence, comme de nettoyer un verrouillage d'appareil, une reprise sur défaillance de HA, etc. Dans ces circonstances, il est souhaitable d'effectuer rapidement le processus de traitement d'erreur, par opposition à l'attente par la cible de plusieurs initiateurs tiers qui peuvent n'être même plus fonctionnels – en particulier si l'urgence est déclenchée par la défaillance d'un de ses initiateurs. Donc, les mises en œuvre de cible et d'initiateur iSCSI DEVRAIENT toutes deux prendre en charge la sémantique FastAbort d'interruption multi tâche (paragraphe 4.2.3.4).

Noter que la sémantique standard (paragraphe 4.2.3.3) et celle de FastAbort (paragraphe 4.2.3.4) permettent les transferts de données en cours même après que l'achèvement de TMF est rapporté sur la session productrice. Dans le cas de iSCSI/iSER [RFC7145], il y aurait des transferts de données étiquetés pour les STag non possédées par des tâches actives. Que les mémoires tampon réelles prennent ou non en charge ces transferts de données dépend de la mise en œuvre. Cependant, les transferts de données DOIVENT logiquement être éliminés en silence par la couche iSCSI de la cible dans tous les cas. Une cible PEUT, sur une fin de temporisation définie en interne par la mise en œuvre, choisir aussi d'abandonner les connexions sur lesquelles elle n'a pas reçu les séquences Data-Out attendues (paragraphe 4.2.3.3) ou les accusés de réception NOP-Out (paragraphe 4.2.3.4) de façon à réclamer les ressources associées de mémoire tampon, STag, et TTT comme approprié.

11. Formats de PDU iSCSI

Tous les entiers multi octets qui sont spécifiés dans les formats définis dans le présent document sont à représenter dans l'ordre des octets du réseau (c'est-à-dire, gros boutien). Tout champ qui apparaît dans le présent document suppose que l'octet de poids fort est l'octet de numéro inférieur et le bit de poids fort (au sein de l'octet ou du champ) est le bit du plus faible numéro, sauf mention contraire.

Tout envoyeur conforme DOIT régler tous les bits non définis et tous les champs réservés à 0, sauf mention contraire. Tout receveur conforme DOIT ignorer tout bit non défini et tous les champs réservés sauf mention contraire. La réception de valeurs de code réservées dans des champs définis DOIT être rapportée comme une erreur de protocole.

Les champs réservés sont marqués du mot "réservé", une abréviation de "réservé", ou par "." pour les bits individuels quand aucune autre forme de marquage n'est techniquement faisable.

11.1 Longueur et bourrage de PDU iSCSI

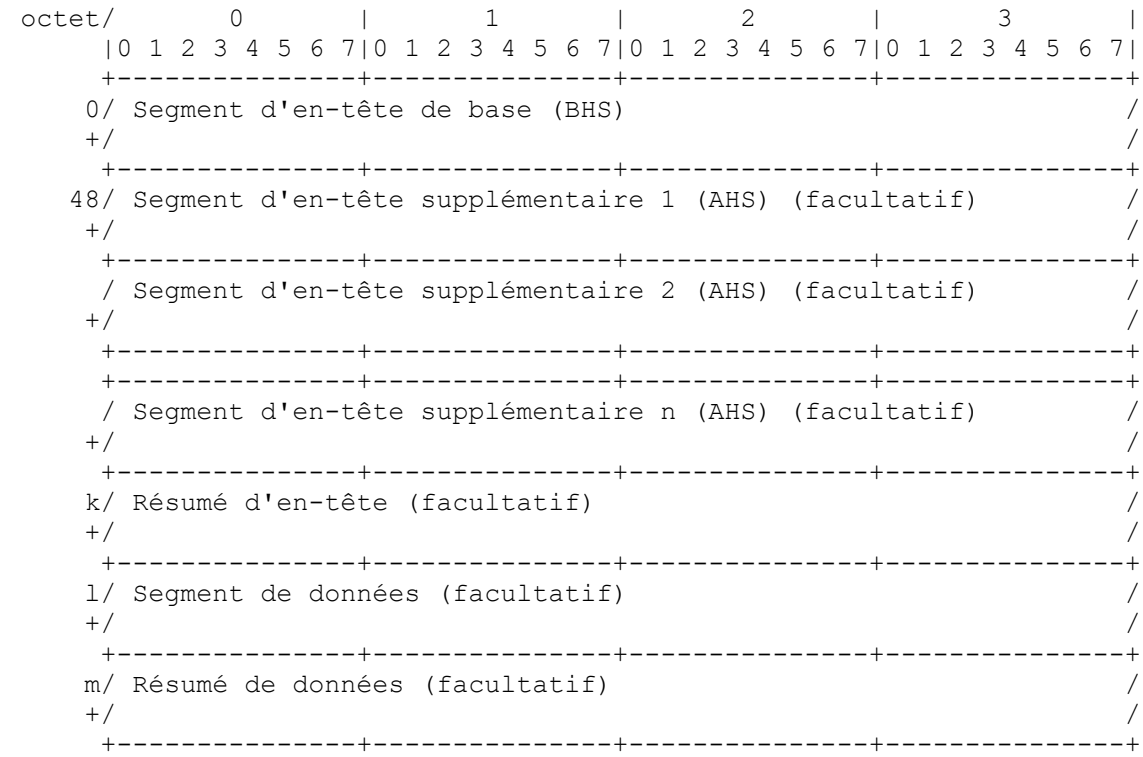
Les PDU iSCSI sont bourrées au plus proche nombre entier de mot de quatre octets. Les octets de bourrage DEVRAIENT être envoyés à 0.

11.2 Gabarit, en-tête et opcodes de PDU

Toutes les PDU iSCSI ont un ou plusieurs segments d'en-tête et, facultativement, un segment de données. Après le groupe entier de segments d'en-tête, un résumé d'en-tête PEUT suivre. Le segment de données PEUT aussi être suivi par un résumé des données.

Le segment d'en-tête de base (BHS, *Basic Header Segment*) est le premier segment dans toutes les PDU iSCSI. Le BHS est un segment d'en-tête d'une longueur fixe de 48 octets. Il PEUT être suivi par des segments d'en-tête supplémentaire (AHS, *Additional Header Segment*), un résumé d'en-tête, un segment de données, et/ou un résumé de données.

La structure globale d'une PDU iSCSI est la suivante :



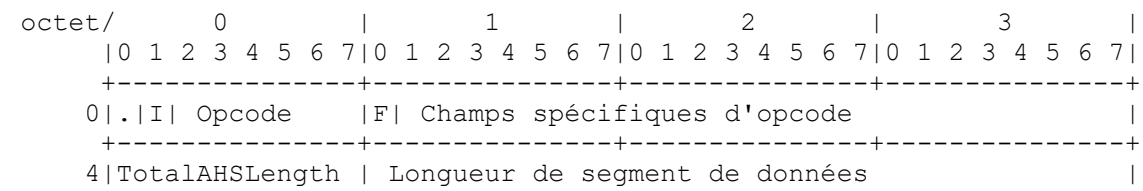
Tous les segments et résumés de PDU sont bourrés au plus proche nombre entier de mots de 4 octets. Par exemple, tous les segments et résumés de PDU commencent à une limite de mot de 4 octets, et le bourrage va de 0 à 3 octets. Les octets de bourrage DEVRAIENT être envoyés à 0.

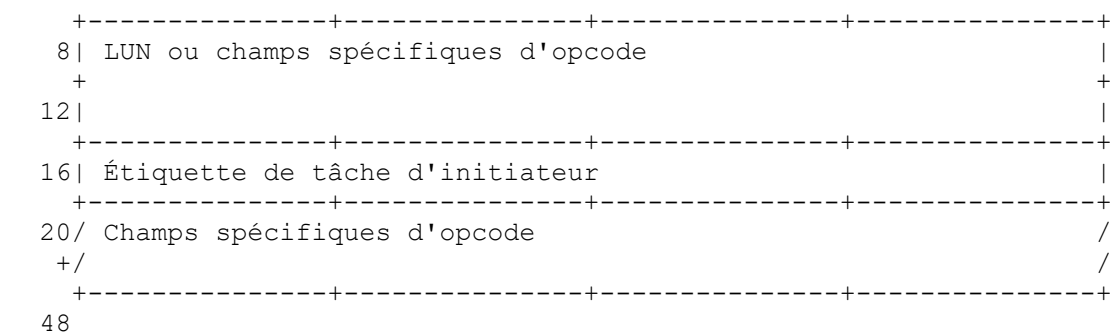
Les PDU de réponse iSCSI n'ont pas de segment AH.

11.2.1 Segment d'en-tête de base (BHS)

Le BHS est long de 48 octets. Les champs Opcode et Longueur de segment de données apparaissent dans toutes les PDU iSCSI. De plus, quand ils sont utilisés, l'étiquette de tâche d'initiateur et le nombre d'unités logiques apparaissent toujours au même endroit dans l'en-tête.

Le format du BHS est :





11.2.1.1 Bit I (Immédiat)

Pour les PDU de demande, le bit I réglé à 1 est un marqueur de livraison immédiate.

11.2.1.2 Opcode

Le Opcode indique le type de PDU iSCSI que l'en-tête encapsule.

Les Opcodes sont divisés en deux catégories : Opcodes d'initiateur et Opcodes de cible. Les Opcodes d'initiateur sont des PDU envoyées par l'initiateur (PDU de demande). Les Opcodes de cible sont des PDU envoyées par la cible (PDU de réponse).

Les initiateurs NE DOIVENT PAS utiliser des Opcodes de cible, et les cibles NE DOIVENT PAS utiliser des Opcodes d'initiateur.

Les Opcodes d'initiateur définis dans cette spécification sont :
0x00 : NOP-Out
0x01 : Commande SCSI (encapsule un bloc de descripteur de commande SCSI)
0x02 : Demande de fonction de gestion de tâche SCSI
0x03 : Demande d'établissement de connexion
0x04 : Demande Text
0x05 : Data-Out SCSI (pour des opérations d'écriture)
0x06 : Demande de fin d'établissement
0x10 : Demande SNACK
0x1c-0x : codes spécifiques de fabricant

Les Opcodes de cible sont :
0x20 : NOP-In
0x21 : Réponse SCSI – contient l'état SCSI et éventuellement des informations de sens ou d'autres informations de réponse
0x22 : Réponse de fonction de gestion de tâche SCSI
0x23 : Réponse d'établissement de connexion
0x24 : Réponse Text
0x25 : Data-In SCSI (pour des opérations de lecture)
0x26 : Réponse de fin d'établissement
0x31 : Prêt à transférer (R2T) – envoyé par la cible quand elle est prête à recevoir des données
0x32 : Message asynchrone – envoyé par la cible pour indiquer certaines conditions particulières
0x3c-0x3e : Codes spécifiques du fabricant
0x3f : Rejet

Tous les autres Opcodes sont non alloués.

11.2.1.3 Bit F (Final)

Quand réglé à 1, il indique la PDU finale (ou la seule) d'une séquence.

11.2.1.4 Champs spécifiques d'Opcode

Ces champs ont des significations différentes pour les différents types d'Opcode.

11.2.1.5 TotalAHSLength

C'est la longueur totale de tous les segments d'en-tête AHS en unités de mots de 4 octets, , incluant le bourrage, s'il en est.

TotalAHSLength n'est utilisé que dans les PDU qui ont un AHS et DOIT être 0 dans toutes les autres PDU.

11.2.1.6 DataSegmentLength

C'est la longueur de charge utile du segment de données en octets (excluant le bourrage). Longueur du segment de données *(DataSegmentLength)* DOIT être 0 chaque fois que la PDU n'a pas de segment de données.

11.2.1.7 LUN

Certains Opcodes opèrent sur une LU spécifique. Le champ Numéro d'unité logique (LUN, *Logical Unit Number*) identifie quelle LU. Si le Opcode ne se rapporte pas à une LU, ce champ est soit ignoré, soit peut être utilisé d'une façon spécifique de l'Opcode. Le champ LUN fait 64 bits et devrait être formaté conformément à [SAM2]. Par exemple, LUN[0] de [SAM2] est l'octet 8 de BHS et ainsi de suite jusqu'à LUN[7] de [SAM2], qui est l'octet 15 de BHS.

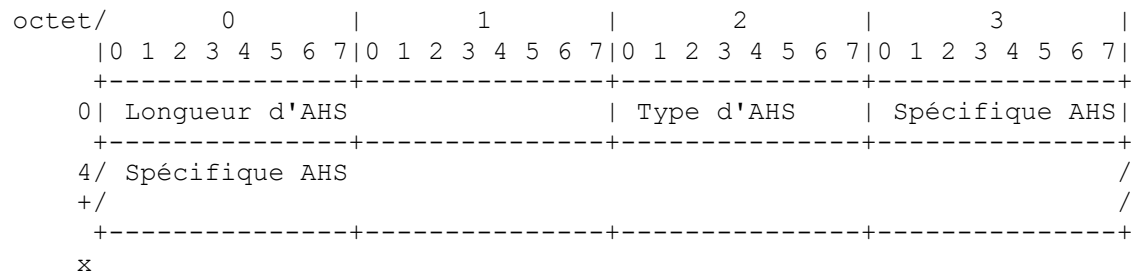
11.2.1.8 Étiquette de tâche d'initiateur

L'initiateur alloue une étiquette de tâche à chaque tâche iSCSI qu'il produit. Quand une tâche existe, cette étiquette DOIT identifier de façon univoque la tâche pour toute la session. SCSI peut aussi utiliser l'étiquette de tâche d'initiateur au titre de l'identifiant de tâche SCSI quand le temps durant lequel l'étiquette de tâche d'initiateur iSCSI doit être unique s'étend sur l'espace de temps pendant lequel une étiquette de tâche SCSI doit être unique. Cependant, l'étiquette de tâche d'initiateur iSCSI doit exister et être unique même pour les commandes SCSI non étiquetées.

La valeur d'ITT de 0xffffffff est réservée et NE DOIT PAS être allouée pour une tâche par l'initiateur. La seule instance dans laquelle elle peut être vue sur le réseau est dans une PDU NOP-In initiée par la cible (paragraphe 11.19) et dans la réponse de l'initiateur à cette PDU, si nécessaire.

11.2.2 Segment d'en-tête supplémentaire (AHS)

Le format général d'un AHS est :



11.2.2.1 Type d'AHS

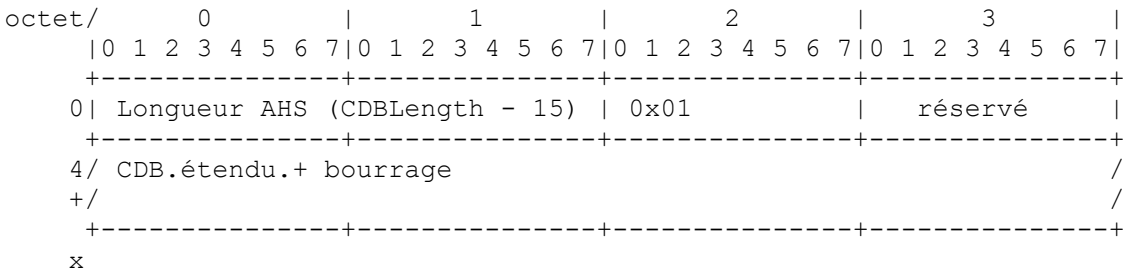
Le champ Type d'AHS est codé comme suit :
bit 0-1 : réservé
bit 2-7 : code d'AHS :
- 0 : réservé
- 1 : CDB étendu
- 2 : Longueur bidirectionnelle du transfert de données de lecture attendu
- 3 : 63 réservé

11.2.2.2 Longueur d'AHS

Ce champ contient la longueur effective en octets de l'AHS, excluant les champs Type d'AHS et Longueur d'AHS et le bourrage, s'il en est. Le AHS est bourré au plus petit nombre entier de mots de 4 octets (c'est-à-dire, de 0 à 3 octets de bourrage).

11.2.2.3 AHS CDB étendu

Le format de l'AHS CDB étendu est :

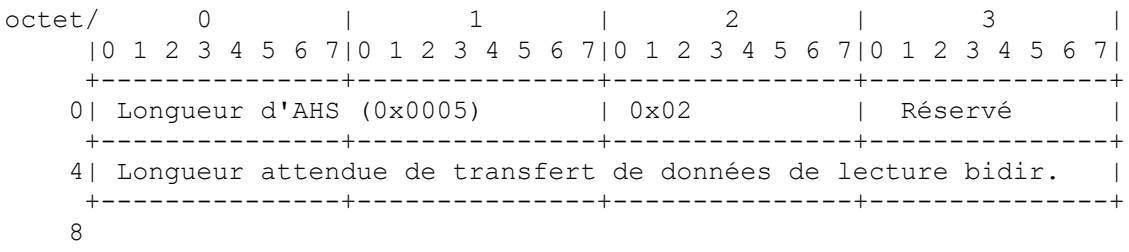


Ce type d'AHS NE DOIT PAS être utilisé si la longueur de CDB est inférieure à 17.

La longueur inclut l'octet réservé 3.

11.2.2.4 AHS Longueur attendue de transfert de données de lecture bidirectionnelle

Le format de l'AHS Longueur attendue de transfert de données de lecture bidirectionnelle est :



11.2.3 Résumé d'en-tête et résumé de données

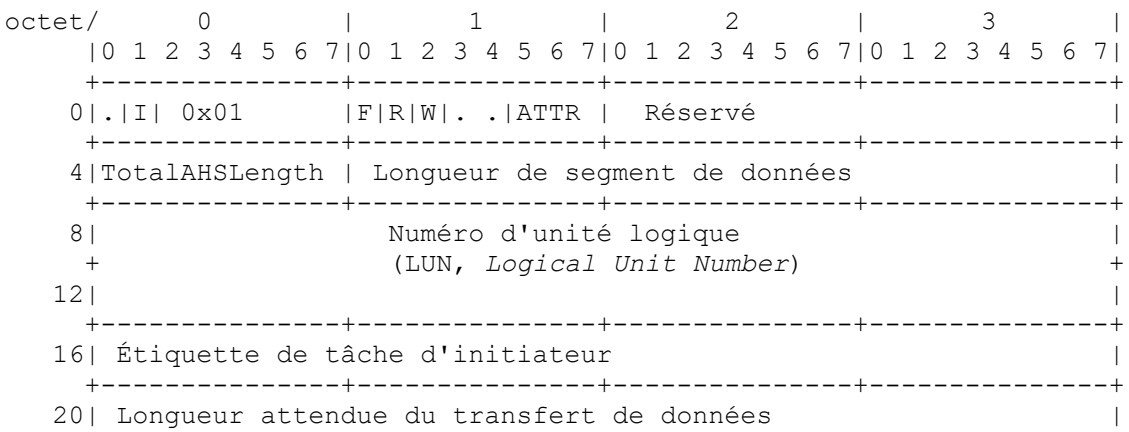
Les résumés d'en-tête et de données facultatifs protègent l'intégrité, respectivement de l'en-tête et des données. Les résumés, s'ils sont présents, sont situés, respectivement, après l'en-tête et les données spécifiques de PDU et couvrent, respectivement, l'en-tête et les données de PDU, chacun incluant les octets de bourrage, s'il en est.
L'existence et le type des résumés sont négociés durant la phase d'établissement *(Login)*.
La séparation des résumés d'en-tête et de données est utile dans les applications d'acheminement iSCSI dans lesquelles seul l'en-tête change quand un message est transmis. Dans ce cas, seul le résumé d'en-tête devrait être recalculé.
Les résumés ne sont pas inclus dans les champs de longueur de données ou d'en-tête.
Un segment de données de longueur zéro implique aussi un résumé de données de longueur zéro.

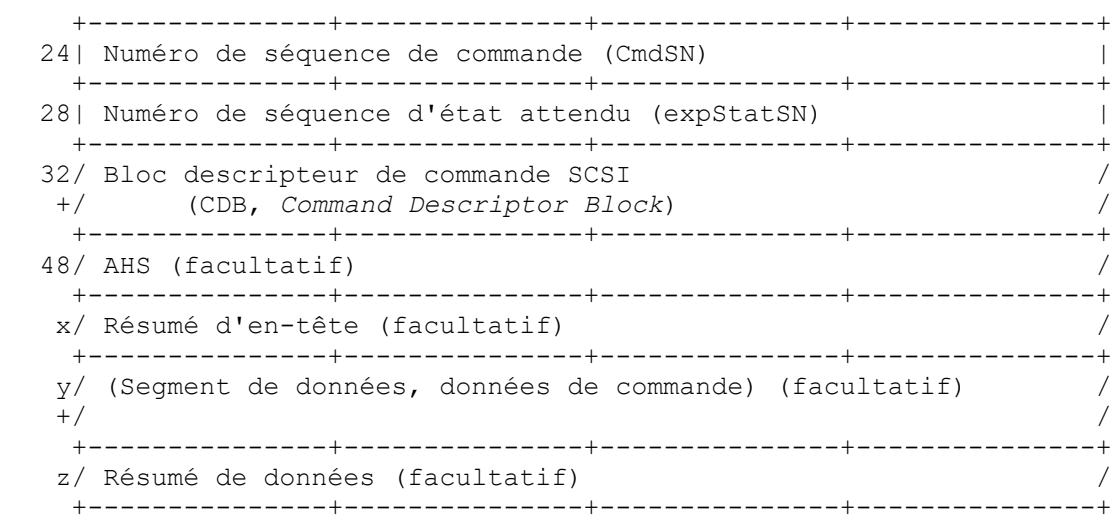
11.2.4 Segment de données

Le segment de données (facultatif) contient les données associées à la PDU. Sa longueur effective de charge utile est fournie dans le champ BHS – Longueur de segment de données. Le segment de données est aussi bourré jusqu'à un nombre entier de mots de quatre octets.

11.3 Commande SCSI

Le format de la PDU Commande SCSI est :





11.3.1 Fanions et attributs de tâche (octet 1)

Les fanions pour une PDU de commande SCSI sont :

bit 0 (F) : réglé à 1 quand aucune PDU Data-Out SCSI non sollicitée ne suit cette PDU. Quand F = 1 pour une écriture et si Longueur attendue de transfert de données est supérieur à Longueur de segment de données, la cible peut solliciter des données supplémentaires via R2T.

bit 1 (R) : réglé à 1 quand on s'attend à ce que la commande entre des données.

bit 2 (W) : réglé à 1 quand on s'attend à ce que la commande envoie des données.

bit 3-4 : réservé.

Bit 5-7 : contient des attributs de tâches.

Les attributs de tâches (ATTR, *Task Attribute*) ont une des valeurs d'entier suivantes (voir les détails dans [SAM2]) :
0 : non étiqueté
1 : simple
2 : ordonné
3 : tête de file d'attente
4 : ACA
5-7 : réservé

Au moins un des bits W et F DOIT être réglé à 1.

L'un ou/et l'autre de R et W PEUT être à 1 quand la longueur attendue de transfert de données et/ou de la longueur attendue de transfert de données bidirectionnelles de lecture est 0, mais ils NE DOIVENT PAS être tous deux à 0 quand la longueur attendue de transfert de données et/ou de la longueur attendue de transfert de données bidirectionnelles de lecture n'est pas 0 (c'est-à-dire, quand un transfert de données est attendu, la direction du transfert est indiquée par le bit R et/ou W).

11.3.2 CmdSN - numéro de séquence de commande

Le CmdSN permet la livraison ordonnée à travers plusieurs connexions dans une seule session.

11.3.3 ExpStatSN – numéro de séquence d'état attendu

Les réponses de commandes jusqu'à ExpStatSN - 1 (modulo 2**32) ont été reçues (état acquitté) sur la connexion.

11.3.4 Longueur de transfert de données attendue

Pour les opérations unidirectionnelles, le champ Longueur attendue de transfert de données contient le nombre d'octets de données impliqué dans cette opération SCSI. Pour une opération unidirectionnelle d'écriture (fanion W réglé à 1 et fanion R réglé à 0) l'initiateur utilise ce champ pour spécifier le nombre d'octets de données qu'il s'attend à transférer pour cette

opération. Pour une opération unidirectionelle de lecture (fanion W réglé à 0 et fanion R réglé à 1) l'initiateur utilise ce champ pour spécifier le nombre d'octets de données qu'il s'attend que la cible lui transfère. Cela correspond au compte d'octets SAM-2.

Pour les opérations bidirectionnelles (les deux fanions R et W sont réglés à 1) ce champ contient le nombre d'octets de données impliqué dans le transfert en écriture. Pour les opérations bidirectionnelles, un segment d'en-tête supplémentaire DOIT être présent dans la séquence d'en-tête qui indique la longueur attendue de transfert de données bidirectionnel en lecture. Les champs Longueur attendue de transfert de données et Longueur attendue de transfert de données bidirectionnel en lecture correspondent au compte d'octtets SAM-2.

Si la longueur attendue de transfert de données pour une écriture et la longueur de la partie de données immédiates qui suivent la commande (si il en est) sont les mêmes, aucune autre PDU de données n'est attendue à suivre. Dans ce cas, le bit F DOIT être réglé à 1.

Si la longueur attendue de transfert de données est supérieure à la longueur de première salve *(FirstBurstLength)* (quantité maximum négociée de données non sollicitées que la cible va accepter) l'initiateur DOIT envoyer la quantité maximum de données non sollicitées OU SEULEMENT les données immédiates, s'il en est.

À l'achèvement d'un transfert de données, la cible informe l'initiateur (par les comptes résiduels) du nombre d'octets réellement traités (envoyés et/ou reçus) par la cible.

11.3.5 CDB – bloc de descripteur de commande SCSI

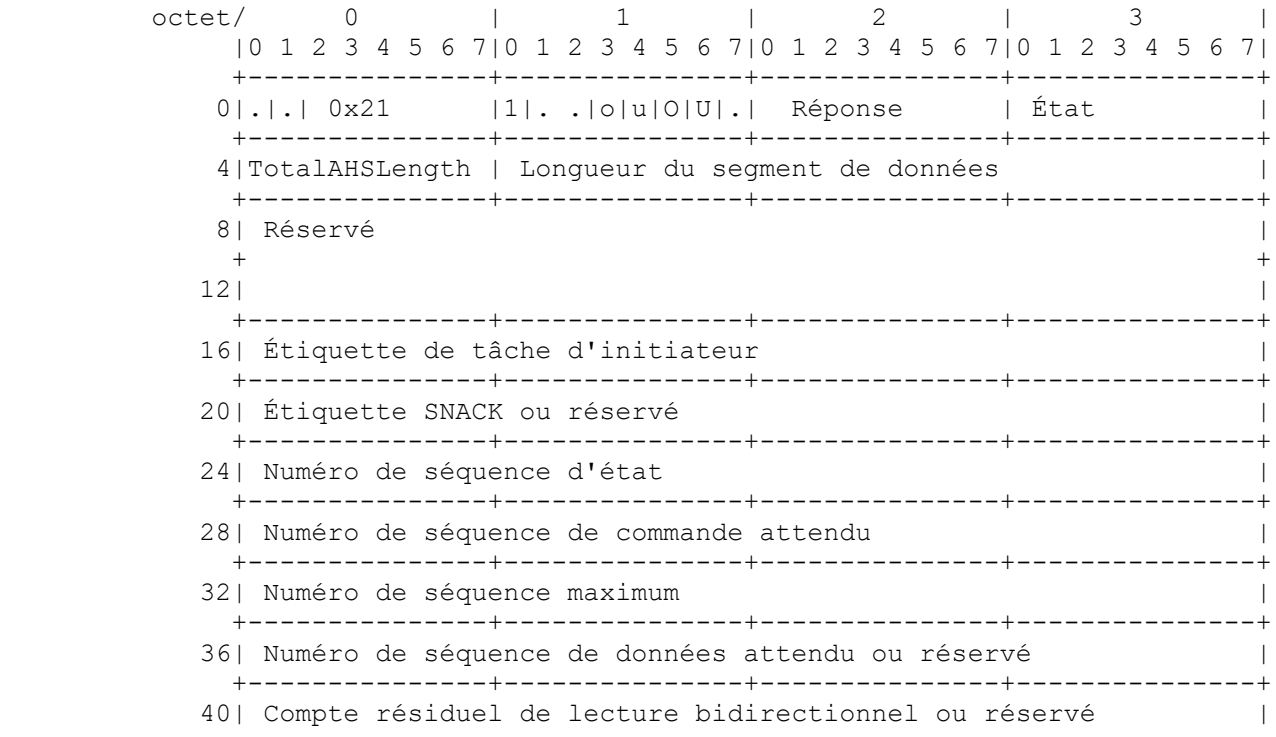
Il y a 16 octets dans le champ CDB pour s'accommoder des CDB couramment utilisés. Chaque fois que le CDB fait plus de 16 octets, un CDB AHS étendu DOIT être utilisé pour contenir le surplus de CDB.

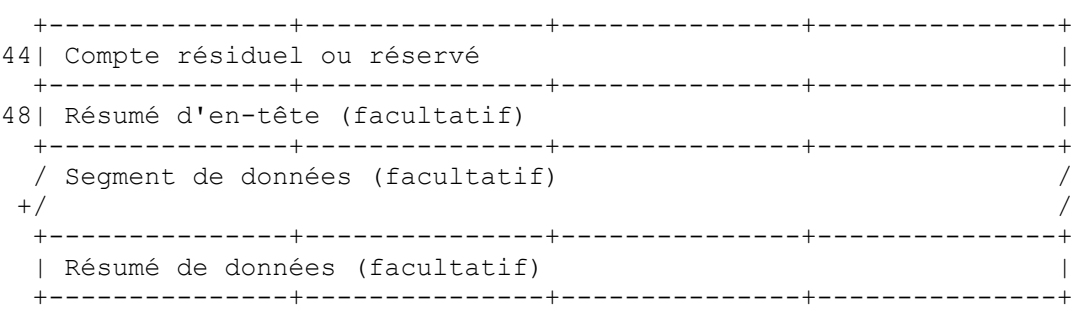
11.3.6 Segment de données – données de commande

Certaines commandes SCSI exigent des données de paramètres supplémentaires pour accompagner la commande SCSI. Ces données peuvent être placées au delà des limites de l'en-tête iSCSI dans un segment de données. Autrement, les données d'utilisateur (par exemple, venant d'une opération d'écriture) peuvent être placées dans le segment de données (les deux cas sont appelés des données immédiates). Ces données sont gouvernées par les règles pour les données sollicitées/non sollicitées mentionnées au paragraphe 4.2.5.2.

11.4 Réponse SCSI

Le format de la PDU Réponse SCSI est :





11.4.1 Fanions (octet 1)

bit 1-2 : réservé.

bit 3 - (o) établi pour un débordement de résidu de lecture bidirectionnel. Dans ce cas, le compte résiduel de lecture bidirectionnel indique le nombre d'octets qui n'ont pas été transférés à l'initiateur parce que la longueur attendue de transfert de données bidirectionnelles en lecture de l'initiateur n'était pas suffisante.

bit 4 - (u) établi pour une sous estimation du résidu de lecture bidirectionnel. Dans ce cas, le compte du résidu de lecture bidirectionnel indique le nombre d'octets qui n'ont pas été transférés à l'initiateur sur le nombre d'octets attendus pour le transfert.

bit 5 - (O) établi pour un débordement de résidu. Dans ce cas, le compte résiduel indique le nombre d'octets qui n'ont pas été transférés à cause de l'insuffisance de la longueur attendue de transfert de données de l'initiateur. Pour une opération bidirectionnelle, le compte résiduel contient le résidu pour l'opération d'écriture.

bit 6 - (U) établi pour la sous estimation du résidu. Dans ce cas, le compte résiduel indique le nombre d'octets qui n'ont pas été transférés sur le nombre d'octets qui étaient attendus du transfert. Pour une opération bidirectionnelle, le compte résiduel contient le résidu pour l'opération d'écriture.

bit 7 - (0) réservé.

Les bits O et U et les bits o et u sont mutuellement exclusifs (c'est-à-dire, avoir les deux o et u ou O et U réglés à 1 est une erreur de protocole).

Pour une réponse autre que "Commande achevée à la cible", les bits 3 à 6 DOIVENT être 0.

11.4.2 État

Le champ État est utilisé pour rapporter l'état SCSI de la commande (comme spécifié dans [SAM2]) et n'est valide que si le code de réponse est “Commande achevée à la cible”.

Certains des codes d'état définis dans [SAM2] sont :
0x00 : Bon
0x02 : Condition de vérification
0x08 : Occupé
0x18 : Conflit de réservation
0x28 : Jeu de tâches plein
0x30 : ACA actif
0x40 : Tâche interrompue

Voir dans [SAM2] la liste complète et les définitions.

Si une erreur d'appareil SCSI est détectée alors que des données de l'initiateur sont encore attendues (la PDU de commande ne contenait pas toutes les données et la cible n'a pas reçu une PDU de données avec le bit Final établi) la cible DOIT attendre de recevoir une PDU de données avec le bit F établi dans la dernière séquence attendue avant d'envoyer la PDU de réponse.

11.4.3 Réponse

Ce champ contient la réponse de service iSCSI.

Les codes de réponse de service iSCSI définis dans la présente spécification sont :
0x00 : Commande achevée à la cible
0x01 : Échec de la cible
0x80-0xff : spécifiques du fabricant

Tous les autres codes de réponse sont réservés.

Le champ Réponse est utilisé pour rapporter une réponse de service. La transposition du code de réponse en une valeur de code de réponse de service SCSI, si nécessaire, sort du domaine d'application de ce document. Cependant, en termes symboliques, la valeur de réponse 0x00 se transpose en la réponse de service SCSI (voir [SAM2] et [SPC3]) de TASK COMPLETE ou LINKED COMMAND COMPLETE. Toutes les autres valeurs de réponse se transposent en la réponse de service SCSI de SERVICE DELIVERY OR TARGET FAILURE.

Si une PDU Réponse SCSI n'arrive pas avant la fin de la session, la réponse de service SCSI est SERVICE DELIVERY OR TARGET FAILURE.

Un champ de réponse non à zéro indique un échec d'exécution de la commande, et dans ce cas les champs État et Fanions sont indéfinis et DOIVENT être ignorés à réception.

11.4.4 Étiquette SNACK

Ce champ contient une copie de l'étiquette SNACK de la dernière étiquette SNACK acceptée par la cible sur la même connexion et pour la commande pour laquelle la réponse est produite. Autrement, il est réservé et devrait être réglé à 0.

Après avoir produit un SNACK R-Data, l'initiateur doit éliminer tout état SCSI sauf contenu dans une PDU Réponse SCSI portant la même étiquette SNACK comme le dernier SNACK R-Data produit pour la commande SCSI sur la connexion en cours.

Pour une discussion détaillée sur le SNACK R-Data, voir le paragraphe 11.16.3.

11.4.5 Compte résiduel

11.4.5.1 Sémantique du champs

Le champ Compte résiduel DOIT être valide dans le cas où le bit U ou le bit O serait établi. Si aucun des deux bits n'est établi, le champ Compte résiduel DOIT être ignoré à réception et DEVRAIT être réglé à 0 à l'envoi. Les cibles peuvent établir le compte résiduel, et les initiateurs peuvent l'utiliser quand le code de réponse est “Commande achevée à la cible” (même si l'état retourné n'est pas BON). Si le bit O est établi, le compte résiduel indique le nombre d'octets qui n'ont pas été transférés parce que la longueur attendue de transfert de données de l'initiateur n'était pas suffisante. Si le bit U est établi, le compte résiduel indique le nombre d'octets qui n'ont pas été transférés sur le nombre d'octets attendus du transfert.

11.4.5.2 Généralités sur le concept de résiduel

"Longueur de transfert de données présentées par SCSI” (SPDTL, *SCSI-Presented Data Transfer Length*)" est le terme qu'utilise le présent document (voir la définition au paragraphe 2.2) pour représenter la longueur des données agrégées que la couche SCSI de la cible tente de transférer en utilisant la couche locale iSCSI pour une tâche. "Longueur attendue de transfert des données” (EDTL, *Expected Data Transfer Length*)" est le terme iSCSI qui représente la longueur des données que la couche iSCSI s'attend à transférer pour une tâche. EDTL est spécifié dans la PDU de commande SCSI.

Quand SPDTL = EDTL pour une tâche, la couche iSCSI de la cible achève la tâche sans résidu. Chaque fois que SPDTL diffère de EDTL pour une tâche, cette tâche est dite avoir un résidu.

Si SPDTL > EDTL pour une tâche, le débordement iSCSI DOIT être signalé dans la PDU Réponse SCSI comme spécifié au paragraphe 11.4.5.1. Le compte résiduel DOIT être réglé à la valeur numérique de (SPDTL - EDTL).

Si SPDTL < EDTL pour une tâche, la sous estimation iSCSI DOIT être signalée dans la PDU Réponse SCSI comme spécifié au paragraphe 11.4.5.1. Le compte résiduel DOIT être réglé à la valeur numérique de (EDTL - SPDTL).

Noter que les scénarios de débordement et de sous estimation sont indépendant de Data-In et Data-Out. L'un et l'autre scénarios sont logiquement possibles dans l'une et l'autre direction du transfert de données.

11.4.5.3 Commande SCSI REPORT LUNS et débordement résiduel

Ce paragraphe expose les questions de débordement résiduel, citant l'exemple de la commande SCSI REPORT LUNS. Noter, cependant, qu'il y a plusieurs commandes SCSI (par exemple, INQUIRY) avec des champs ALLOCATION LENGTH qui suivent les mêmes règles sous-jacentes. La sémantique dans le reste de ce paragraphe s'applique à toutes ces commandes SCSI.

La spécification de la commande SCSI REPORT LUNS exige que la cible SCSI limite la quantidé de données transférés à une taille maximum (ALLOCATION LENGTH) fournie par l'initiateur dans le CBD REPORT LUNS.

Si la longueur attendue du transfert de données (EDTL) dans l'en-tête iSCSI de la PDU de commande SCSI pour une commande REPORT LUNS est réglée à au moins autant que ALLOCATION LENGTH, la troncature de couche SCSI empêche un débordement résiduel iSCSI de se procuire. Un initiateur SCSI peut détecter qu'une telle troncature s'est produite via dautres informations à la couche SCSI. La suite de ce paragraphe développe ce comportement exigé.

La commande SCSI REPORT LUNS demande à la couche SCSI de la cible de retourner un inventaire de LU (liste LUN) à la couche SCSI de l'initiateur (voir au paragraphe 6.21 de [SPC3]). La taille de cette liste LUN peut n'être pas connue de la couche SCSI de l'initiateur quand il produit la commande REPORT ; pour éviter de transférer plus de données de liste LUN que ce à quoi est préparé l'initiateur, le CDB REPORT LUNS contient un champ ALLOCATION LENGTH pour spécifier la quantité maximum de données à transférer à l'initiateur pour cette commande. Si la couche SCSI de l'initiateur a sous estimé le nombre de LU à la cible, il est possible que l'inventaire complet de LU ne tienne pas dans la ALLOCATION LENGTH spécifiée. Dans cette situation, le paragraphe 4.3.4.6 de [SPC3] exige que la couche SCSI de la cible "termine les transferts dans la mémoire tampon Data-In" quand le nombre d'octets spécifié par le champ ALLOCATION LENGTH a été transféré.

Donc, en réponse à une commande REPORT LUNS, la couche SCSI à la cible présente au plus ALLOCATION LENGTH octets de données (inventaire de LU) à iSCSI à transférer à l'initiateur. Pour une commande REPORT LUNS, si l'EDTL iSCSI est au moins égal à ALLOCATION LENGTH, la troncature SCSI assure que l'EDTL va s'accommoder de toutes les données à transférer. Si toutes les données de l'inventaire de LU présentées à la couche iSCSI -- c'est-à-dire, les données qui restent après une troncature SCSI -- sont transférées à l'initiateur par la couche iSCSI, un débordement résiduel iSCSI ne s'est pas produit et le bit (O) iSCSI NE DOIT PAS être établi dans la PDU Réponse SCSI ou la PDU Data-Out finale SCSI. Noter que ce comportement est impliqué au paragraphe 11.4.5.1, avec la spécification de la commande REPORT LUNS dans [SPC3]. Cependant, si l'EDTL iSCSI est plus grand que ALLOCATION LENGTH dans ce scénario, noter que la sous estimation iSCSI DOIT être signalée dans la PDU Réponse SCSI. Une sous estimation iSCSI DOIT aussi être signalée quand l'EDTL iSCSI est égal au ALLOCATION LENGTH mais que les données d'inventaire de LU présentées à la couche iSCSI sont plus petittes que ALLOCATION LENGTH.

Le champ LUN LIST LENGTH dans l'inventaire de LU (premier champ de l'inventaire) n'est pas affecté par la troncature de l'inventaire pour qu'il tienne dans ALLOCATION LENGTH ; cela permet à un initiateur SCSI de déterminer que l'inventaire reçu est incomplet en remarquant que le LUN LIST LENGTH de l'inventaire est plus grand que ALLOCATION LENGTH qui a été envoyée dans le CDB REPORT LUNS. Un comportement courant de l'initiateur dans cette situation est de produire à nouveau la commande REPORT LUNS avec une ALLOCATION LENGTH supérieure.

11.4.6 Compte résiduel en lecture bidirectionnel

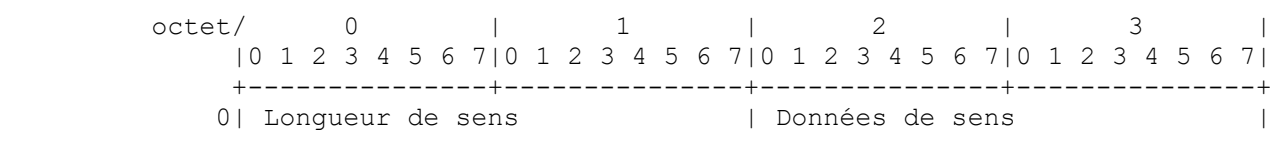
Le champ Compte résiduel en lecture bidirectionnel DOIT être valide dans le cas où est établi le bit u ou le bit o. Si aucun d'eux n'est établi, le champ Compte résiduel en lecture bidirectionnel est réservé. Les cibles peuvent établir le compte résiduel en lecture bidirectionnel, et les initiateurs peuvent l'utiliser quand le code de réponse est “Commande achevée à la cible”. Si le bit o est établi, le compte résiduel de lecture bidirectionnel indique le nombre d'octets qui n'ont pas été transférés à l'initiateur parce que la longueur attendue du transfert de données en lecture bidirectionnel de l'initiateur n'était pas suffisante. Si le bit u est établi, le compte résiduel en lecture bidirectionnel indique le nombre d'octets qui n'ont pas été transférés à l'initiateur sur le nombre d'octets attendus du transfert.

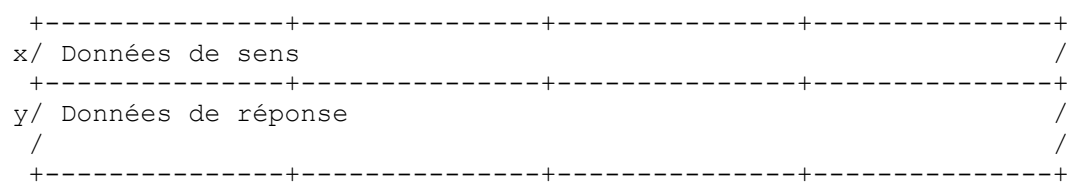
11.4.7 Segment de données - sens et réponse

Les cibles iSCSI DOIVENT prendre en charge et activer Autosense. Si l'état est Vérifier la condition (0x02) le segment de données DOIT contenir des données de sens pour la commande en échec.

Pour certaines réponses iSCSI, le segment de données de réponse PEUT contenir des informations relatives à la réponse (par exemple, pour une défaillance de cible, il peut contenir une description détaillée de la défaillance, spécifique du fabricant).

Si la longueur du segment de données (DataSegmentLength) n'est pas 0, le format du segment de données est :





11.4.7.1 Longueur de sens

Ce champ indique la longueur des données de sens.

11.4.7.2 Données de sens

Les données de sens contiennent des informations détaillées sur une vérification de condition. [SPC3] spécifie le format et le contenu des données de sens.

Certaines conditions iSCSI résultent en ce que la commande se termine à la cible (code de réponse de “Commande achevée à la cible”) avec un état SCSI de Vérification de condition comme mentionné dans le tableau suivant :

Condition iSCSI	**Clé de sens**	**Code et qualificatif de sens supplémentaire**
Données inattendues non sollicitées	Commande 0B interrompue	ASC = 0x0c ASCQ = 0x0c Erreur d'écriture
Quantité de données incorrecte	Commande 0B interrompue	ASC = 0x0c ASCQ = 0x0d Erreur d'écriture
Erreur CRC de service de protocole	Commande 0B interrompue	ASC = 0x47 ASCQ = 0x05 Erreur de CRC détectée
SNACK rejeté	Commande 0B interrompue	ASC = 0x11 ASCQ = 0x13 Erreur de lecture

La cible rapporte la condition de "quantité de données incorrecte" si, durant la sortie des données, la longueur de données totale à sortir est supérieure à la longueur de première salve *(FirstBurstLength)* et si l'initiateur a envoyé des données non sollicitées non immédiates mais que la quantité totale de données non sollicitées est différente de Longueur de première salve. La cible rapporte la même erreur quand la quantité de données envoyées en réponse à un R2T ne correspond pas à la quantité demandée.

11.4.8 ExpDataSN

Ce champ indique le nombre de PDU Data-In (lecture) que la cible a envoyées pour la commande.

Ce champ DOIT être 0 si le code de réponse n'est pas “Commande achevée à la cible ou si la cible n'a pas envoyé de PDU Data-In pour la commands.

11.4.9 StatSN - numéro de séquence d'état

Numéro de séquence d'état est un numéro de séquence que la couche iSCSI de la cible génère par connexion et qui à son tour permet à l'initiateur d'accuser réception de l'état. Numéro de séquence d'état est incrémenté de 1 pour chaque réponse/état envoyé sur une connexion, sauf pour les réponses envoyées par suite d'un réessai ou d'un SNACK. Dans le cas de réponses envoyées suite à une demande de retransmission, le numéro de séquence d'état DOIT être le même que la première fois que la PDU a été envoyée, sauf si la connexion a été redémarrée depuis.

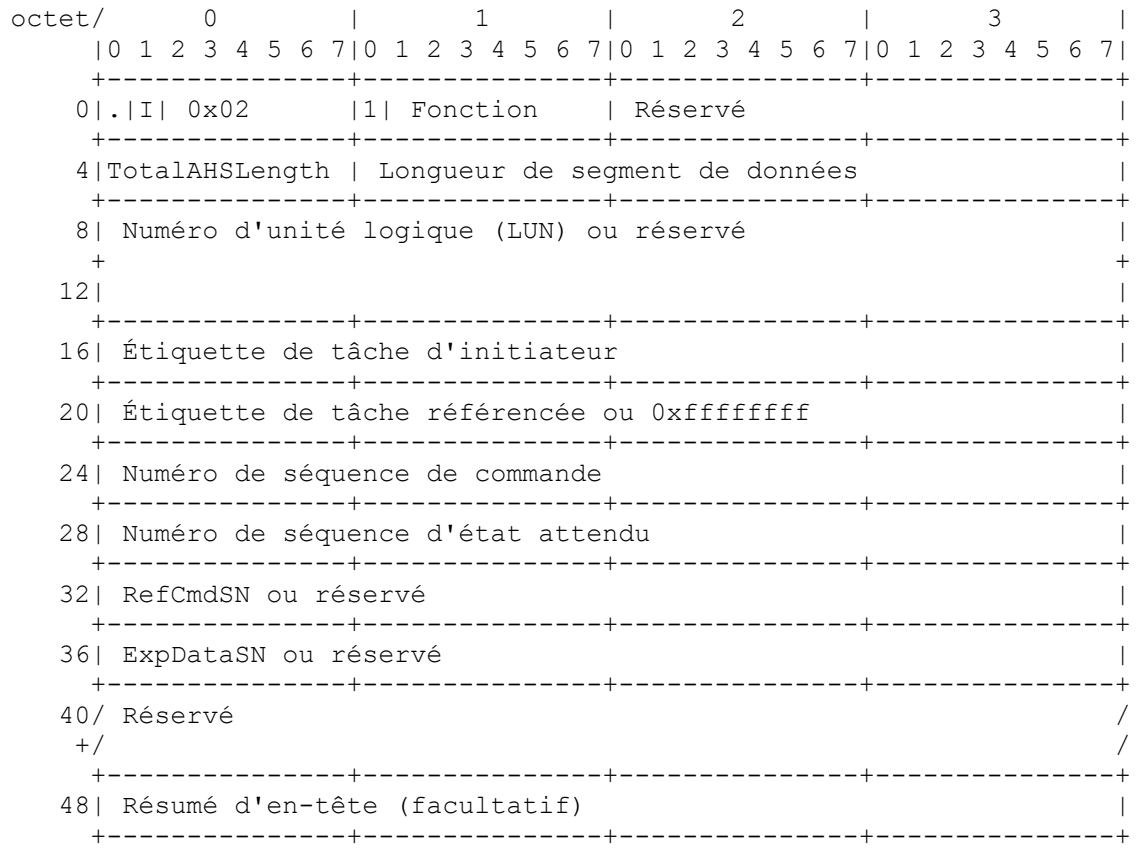
11.4.10 Numéro de séquence de commande attendu – Prochain CmdSN attendu à partir de cet initiateur

ExpCmdSN est un numéro de séquence que la cible iSCSI retourne à l'initiateur pour accuser réception de la commande. Il est utilisé pour mettre à jour une variable locale du même nom. Un ExpCmdSN égal à MaxCmdSN + 1 indique que la cible ne peut pas accepter de nouvelles commandes.

11.4.11 MaxCmdSN - CmdSN maximum à partir de cet initiateur

MaxCmdSN est un numéro de séquence que la cible iSCSI retourne à l'initiateur pour indiquer le CmdSN maximum que l'initiateur peut envoyer. Il est utilisé pour mettre à jour une variable locale du même nom. Si MaxCmdSN est égal à ExpCmdSN - 1, cela indique à l'initiateur que la cible ne peut pas recevoir de commandes supplémentaires. Quand MaxCmdSN change chez la cible alors qu'elle n'a pas de PDU en cours pour porter ces informations à l'initiateur, elle DOIT générer un NOP-In pour porter le nouveau MaxCmdSN.

11.5 Demande de fonction de gestion de tâche



11.5.1 Fonction

Les fonctions de gestion des tâches fournissent à un initiateur un moyen pour contrôler explicitement l'exécution d'une ou plusieurs tâches (SCSI et iSCSI). Les codes de fonction de gestion des tâches sont énumérés ci-dessous. Pour une description plus détaillée de la gestion des tâches SCSI, voir [SAM2].

1 : ABORT TASK – interrompt la tâche identifiée par le champ d'étiquette de tâche référencée.
2 : ABORT TASK SET – interrompt toutes les tâches produites via cette session sur la LU.
3 : CLEAR ACA – supprime la condition d'allégeance auto contingente.
4 : CLEAR TASK SET – interrompt les tâches dans l'ensemble de tâches approprié comme défini par le champ TST dans la page de mode de contrôle (voir [SPC3]).
5 : LOGICAL UNIT RESET *(réinitialisation d'unité logique)*
6 : TARGET WARM RESET *(réinitialisation de cible à chaud)*
7 : TARGET COLD RESET *(réinitialisation de cible à froid)*
8 : TASK REASSIGN – réalloue l'allégeance de connexion pour la tâche identifiée par le champ Étiquette de tâche d'initiateur à cette connexion, reprenant donc les échanges iSCSI pour la tâche.

Les valeurs 9 à 12 sont allouées dans la [RFC7144]. Toutes les autres valeurs possibles pour le champ Fonction sont non allouées.

Pour toutes ces fonctions, la réponse de fonction de gestion de tâche DOIT être retournée comme précisé au paragraphe 11.6. Toutes ces fonctions s'appliquent aux tâches référencées, sans considération de si ce sont des tâches propres de SCSI ou des opérations iSCSI étiquetées. Les demandes de gestion de tâches doivent agir sur toutes les commandes provenant de la même session qui ont un CmdSN inférieur au CmdSN de gestion de la tâche. LOGICAL UNIT RESET, TARGET WARM RESET, et TARGET COLD RESET peuvent affecter les commandes d'autres sessions ou des commandes de la même session, sans considération de leur valeur de CmdSN.

Si la demande de gestion de tâche est marquée pour livraison immédiate, elle doit être considérée immédiatement pour exécution, mais les opérations impliquées (en tout ou partie) peuvent être différées pour permettre à la cible de recevoir toutes les tâches pertinentes. Conformément à [SAM2], pour toutes les tâches couvertes par la réponse de gestion de tâche (c'est-à-dire, avec un CmdSN inférieur au CmdSN de la commande de gestion de tâche) sauf la réponse de gestion de tâche à une TASK REASSIGN *(réallocation de tâche)* des réponses supplémentaires NE DOIVENT PAS être livrées à la couche SCSI après la réponse de gestion de la tâche. L'initiateur iSCSI PEUT livrer à la couche SCSI toutes les réponses reçues avant la réponse de gestion de la tâche (c'est-à-dire, c'est l'affaire de la mise en œuvre que les réponses iSCSI qui sont reçues avant la réponse de gestion de la tâche mais après la production de la demande de gestion de tâche soient livrées à la

couche SCSI par la couche iSCSI chez l'initiateur). La cible iSCSI DOIT s'assurer qu'aucune réponse pour les tâches couvertes par une fonction de gestion des tâches sont livrées à l'initiateur iSCSI après la réponse de gestion de tâche, sauf pour une tâche couverte par une réallocation de tâche.

Pour ABORT TASK SET et CLEAR TASK SET, l'initiateur producteur DOIT continuer de répondre à toutes les étiquettes de tranfert de cible valides (reçues via R2T, réponse Text, NOP-In, ou PDU SCSI Data-In) qui se rapportent à l'ensemble de tâches affectées, même après avoir produit la demande de gestion de tâche.

L'initiateur producteur DEVRAIT cependant terminer (c'est-à-dire, en réglant le bit F à 1) ces séquences de réponse aussi vite que possible. De son côté la cible DOIT attendre les réponses sur toutes les étiquettes de transfert de cible affectées avant d'agir sur l'une ou l'autre de ces demandes de gestion de tâche. Si tout ou partie de la séquence de réponses n'est pas reçu (due à des erreurs de résumé) pour une TTT valide, la cible PEUT la traiter comme un cas d'erreur de récupération d'erreur au sein de la commande (voir au paragraphe 7.1.4.1) si elle prend en charge Niveau de récupération d'erreur ≥ 1 ou, autrement, elle peut abandonner la connexion pour achever la fonction d'ensemble de tâches demandée.

Si un ABORT TASK est produit pour une tâche créée par une commande immédiate, le RefCmdSN DOIT alors être celui de la demande de gestion de tâche elle-même (c'est-à-dire, CmdSN et RefCmdSN sont égaux) ; autrement, le RefCmdSN DOIT être réglé au CmdSN de la tâche à interrompre (plus faible que le CmdSN).

Si la connexion est encore active (c'est-à-dire, n'est pas en train d'effectuer un désétablissement implicite ou explicite) un ABORT TASK DOIT être produit sur la même connexion à laquelle la tâche à interrompre est allégeante au moment où la demande de gestion de tâche est produite. Si la connexion est implicitement ou explicitement en désétablissement (c'est-à-dire, si aucune autre demande ne sera produite sur la connexion défaillante et aucune autre réponse ne sera reçue sur la connexion défaillante) une demande de fonction ABORT TASK peut alors être produite sur une autre connexion. Cette demande de gestion de tâche va alors établir une nouvelle allégeance pour la commande à interrompre ainsi que va l'interrompre (c'est-à-dire, la tâche à interrompre n'aura pas à être réessayée ni réallouée, et son état, si il est envoyé mais pas acquitté, sera renvoyé et suivi par la réponse de gestion de la tâche).

À la cible, une fonction ABORT TASK NE DOIT PAS être exécutée sur une demande de gestion de tâche ; une telle demande DOIT résulter en une réponse de gestion de tâche de "Fonction rejetée".

Pour la fonction LOGICAL UNIT RESET *(réinitialisation d'unité logique)*, la cible DOIT se comporter comme indiqué par la fonction réinitialisation d'unité logique dans [SAM2].

La mise en œuvre de la fonction TARGET WARM RESET et de la fonction TARGET COLD RESET est FACULTATIVE et, quand elle est effectuée, devrait agir comme décrit ci-dessous. TARGET WARM RESET est aussi soumis aux contrôle d'accès SCSI sur l'initiateur demandeur comme défini dans [SPC3]. Quand l'autorisation échoue à la cible, la réponse appropriée comme décrit au paragraphe 11.6.1 DOIT être retournée par la cible. La fonction TARGET COLD RESET n'est pas soumise aux contrôles d'accès SCSI, mais ses privilèges d'exécution peuvent être gérés par des mécanismes iSCSI comme l'authentification d'établissement.

Lors de l'exécution des fonctions TARGET WARM RESET et TARGET COLD RESET, la cible annule toutes les opérations en cours sur toutes les LU connues de l'initiateur producteur. Les deux fonctions sont équivalentes à la fonction TARGET RESET spécifiée dans [SAM2]. Elles peuvent affecter de nombreux autres initiateurs enregistrés à l'accès cible SCSI qui les dessert.

De plus, la cible DOIT traiter la fonction TARGET COLD RESET comme un événement de mise sous tension, terminant donc toutes ses connexions TCP à tous les initiateurs (toutes les sessions sont terminées). Pour cette raison, la réponse de service (définie par [SAM2]) pour cette fonction de gestion des tâches SCSI peut n'être pas livrée fiablement à l'accès de l'initiateur producteur.

Pour la fonction TASK REASSIGN, la cible devrait réallouer l'allégrance de connexion à la nouvelle connexion (et donc reprendre les échanges iSCSI pour la tâche). TASK REASSIGN DOIT être reçu par la cible seulement après que la connexion sur laquelle la commande était précédemmet exécutée a été désétablie avec succès. La réponse de gestion de tâche DOIT être produite avant que la réallocation devienne effective.

Voir le reste de la sémantique d'utilisation au paragraphe 7.2.

À la cible, une demande de fonction TASK REASSIGN NE DOIT PAS être exécutée pour réallouer l'allégeance de connexion d'une demande de fonction de gestion de tâche, une tâche active de négociation text, ou une tâche de fin d'établissement ; une telle demande DOIT résulter en une réponse de gestion de tâche de "Fonction rejetée".

TASK REASSIGN DOIT être produit comme commande immédiate.

11.5.2 TotalAHSLength et DataSegmentLength

Pour cette PDU, TotalAHSLength et DataSegmentLength DOIT être 0.

11.5.3 LUN

Ce champ est exigé pour les fonctions qui s'adressent à une LU spécifique (ABORT TASK, CLEAR TASK SET, ABORT TASK SET, CLEAR ACA, LOGICAL UNIT RESET) et est réservé dans tous les autres.

11.5.4 Étiquette de tâche référencée

C'est l'étiquette de tâche d'initiateur de la tâche à interrompre pour la fonction ABORT TASK ou réallouée pour la fonction TASK REASSIGN. Pour toutes les autres fonctions, ce champ DOIT être réglé à à la valeur réservée de 0xffffffff.

11.5.5 RefCmdSN

Si une ABORT TASK est produite pour une tâche créée par une commande immédiate, le RefCmdSN DOIT alors être celui de la demande de gestion de tâche elle-même (c'est-à-dire, le CmdSN et le RefCmdSN sont égaux).
Pour une ABORT TASK d'une tâche créée par une commande non immédiate, le RefCmdSN DOIT être réglé au CmdSN de la tâche identifiée par le champ Étiquette de tâche référencee. Les cibles doivent utiliser ce champ comme décrit au paragraphe 11.6.1 quand la tâche identifiée par le champ Étiquette de tâche référencee n'est pas avec la cible.
Autrement, ce champ est réservé.

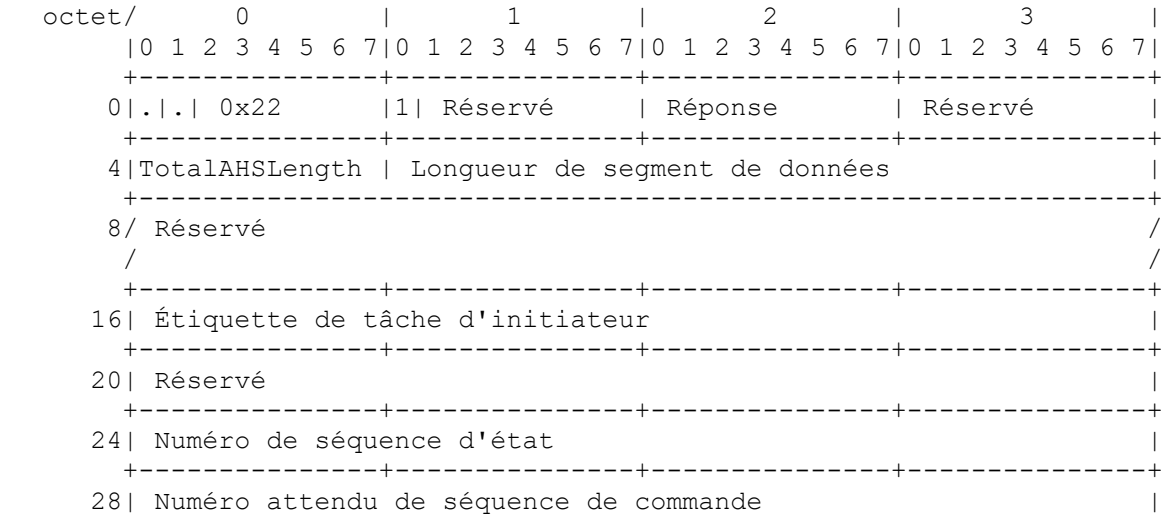
11.5.6 ExpDataSN

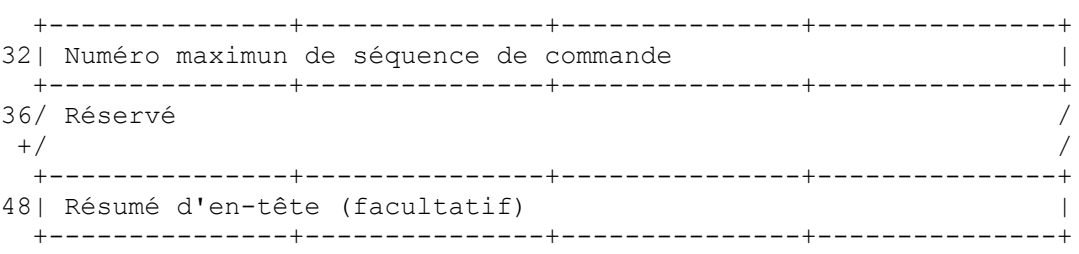
Pour les besoins de récupération, la cible iSCSI et l'initiateur tiennent un numéro de référence d'accusé de réception de données – le premier numéro d'entrée de DataSN non acquitté par l'initiateur. Quand on produit une nouvelle commande, ce numéro est réglé à 0. Si la fonction est TASK REASSIGN, qui établit une nouvelle allégeance de connexion pour une commande en lecture précédemment produite ou bidirectionnelle, le ExpDataSN va contenir un numéro de référence d'accusé de réception de données mis à jour ou la valeur 0 ; cette dernière indique que le numéro de référence d'accusé de réception de données est inchangé. L'initiateur DOIT éliminer toutes les PDU de données provenant de l'exécution précédente qu'il n'a pas acquittées, et la cible DOIT transmettre toutes les PDU Data-In (si il en est) en commençant par le numéro de référence d'accusé de réception de données. Le numéro des PDU retransmises peut ou non être le même que celui de la transmission d'origine, selon qu'il y a eu un changement de la MaxRecvDataSegmentLength dans la réallocation. La cible PEUT aussi ne pas envoyer plus de PDU Data-In si toutes les données ont été acquittées.

La valeur de ExpDataSN DOIT être 0 ou plus que le DataSN de la dernière PDU Data-In acquittée, mais pas plus grande que DataSN + 1 de la dernière PDU Data-IN envoyée par la cible. Toute autre valeur DOIT être ignorée par la cible.

Pour les autres fonctions, ce champ est réservé.

11.6 Réponse de fonction de gestion de tâche





Pour les fonctions ABORT TASK, ABORT TASK SET, CLEAR ACA, CLEAR TASK SET, LOGICAL UNIT RESET, TARGET COLD RESET, TARGET WARM RESET, et TASK REASSIGN, la cible effectue la fonction de gestion des tâches demandée et renvoie une réponse de gestion de tâche à l'initiateur. Pour TASK REASSIGN, la nouvelle allégeance de connexion DOIT devenir effective à la cible seulement après que la cible a produit la réponse de gestion de tâche.

11.6.1 Réponse

La cible fournit une réponse, qui peut prendre une des valeurs suivantes :
0 : Fonction achevée
1 : Tâche inexistante
2 : LUN inexistant
3 : Tâche encore allégeante
4 : Réallocation d'allégeance de tâche non prise en charge
5 : Fonction de gestion de tâche non prise en charge
6 : Échec d'autorisation de fonction
255 : Fonction rejetée

En plus des valeurs ci-dessus, la valeur 7 est définie par la [RFC7144].

Pour une discussion sur l'usage des codes de réponse 3 et 4, voir le paragraphe 7.2.2.

Pour les fonctions TARGET COLD RESET et TARGET WARM RESET, la cible annule toutes les opérations en cours sur toutes les LU connues de l'initiateur producteur. Pour la fonction TARGET COLD RESET, la cible DOIT alors clore toutes ses connexions TCP à tous les initiateurs (terminer toutes les sessions).

La transposition du code de réponse en une valeur de code de service SCSI, si nécessaire, sort du domaine d'application du présent document. Cependant, en termes symboliques, les valeurs de réponse 0 et 1 se transposent en la réponse de service SCSI de Fonction achevée. La valeur de réponse de 2 se transpose en réponse de service SCSI de LUN incorrect. Toutes les autres valeurs de réponse se transposent en la réponse de service SCSI de Fonction rejetée. Si une PDU de réponse de fonction de gestion de tâche n'arrive pas avant que la session soit terminée, la réponse de service SCSI est Échec de livraison de service ou de cible.

La réponse à Interrompre l'ensemble de tâches et Nettoyer l'ensemble de tâches NE DOIT être produite QUE par la cible après que toutes les commandes affectées ont été reçues par la cible, les fonctions de gestion des tâches correspondantes ayant été exécutées par la cible SCSI, et la livraison de toutes les réponses effectuée jusqu'à ce que l'achèvement de la fonction de gestion des tâches ait été confirmée (acquittées avec le ExpStatSN) par l'initiateur sur toutes les connexions de cette session. Pour la succession exacte des événements, voir les paragraphes 4.2.3.3 et 4.2.3.4.

Pour la fonction Interrompre la tâche,
a) si l'étiquette de tâche référencée identifie une tâche valide conduisant à une terminaison réussie, les cibles doivent alors retourner la réponse "Fonction achevée" ;
b) si l'étiquette de tâche référencée n'identifie pas une tâche existante mais si le CmdSN indiqué par le champ RefCmdSN dans la demande de fonction de gestion de tâche est dans la fenêtre valide de CmdSN et moins que le CmdSN de la demande de fonction de gestion de tâche elle-même, les cibles doivent alors considérer le CmdSN comme reçu et retourner la réponse "Fonction achevée" ;
c) si l'étiquette de tâche référencée n'identifie pas une tâche existante et si le CmdSN indiqué par le champ RefCmdSN dans la demande de fonction de gestion de tâche est en dehors de la fenêtre valide de CmdSN, alors les cibles doivent retourner la réponse "Tâche inexistante".

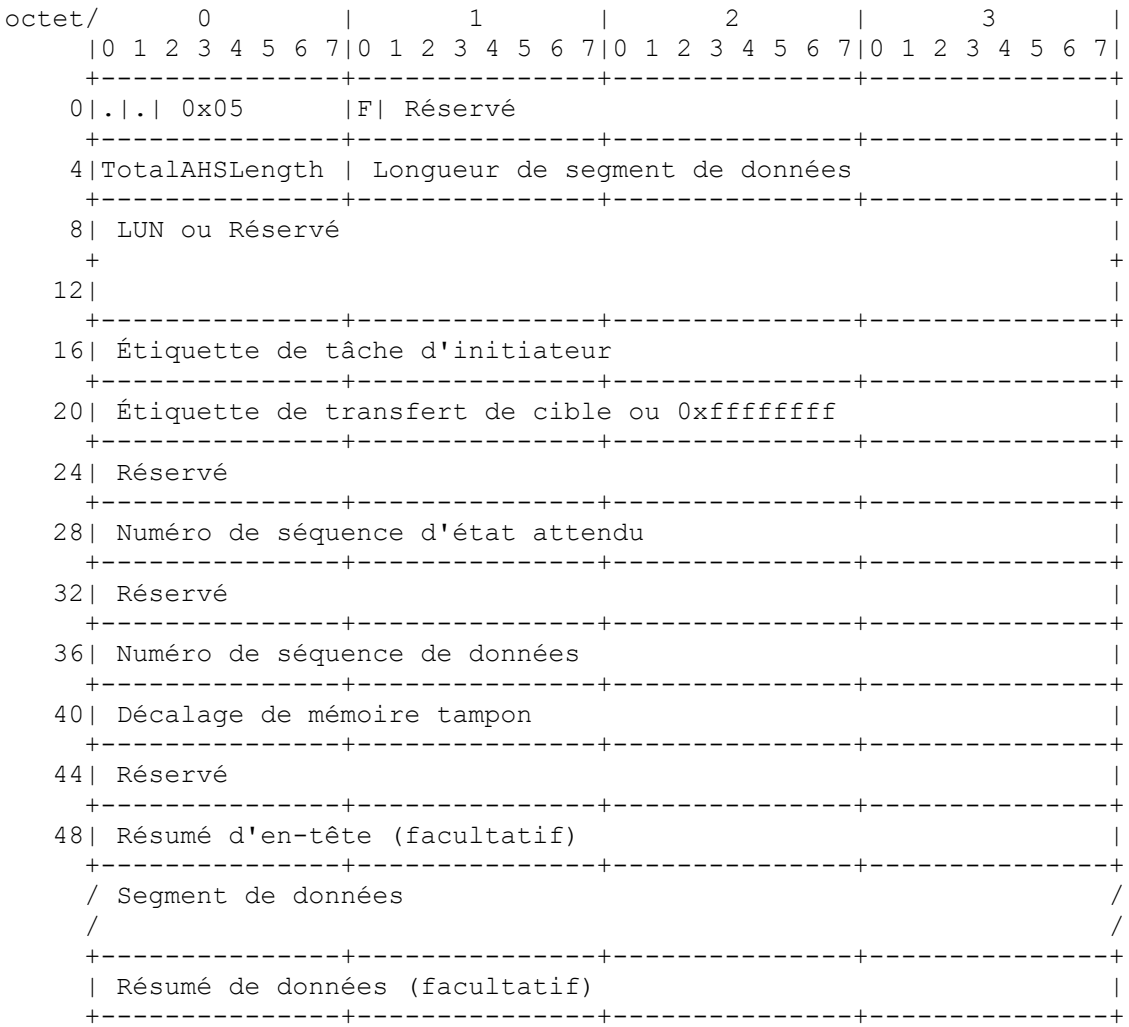
Pour la sémantique des réponses sur les types de fonction qui peuvent impacter plusieurs tâches actives sur la cible, voir le paragraphe 4.2.3.

11.6.2 TotalAHSLength et DataSegmentLength

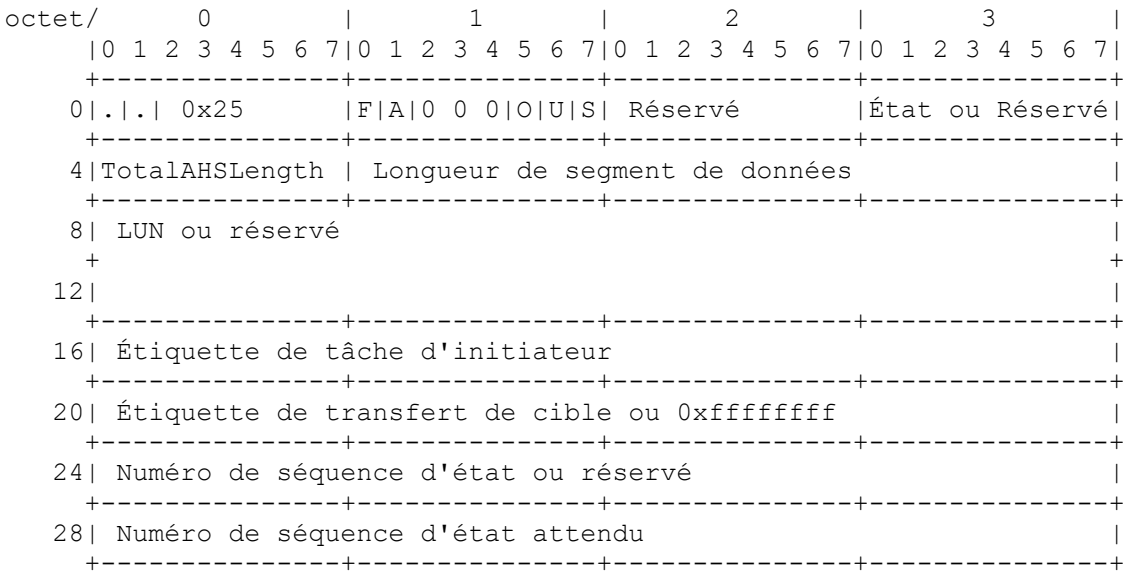
Pour cetre PDU, TotalAHSLength et DataSegmentLength DOIVENT être 0.

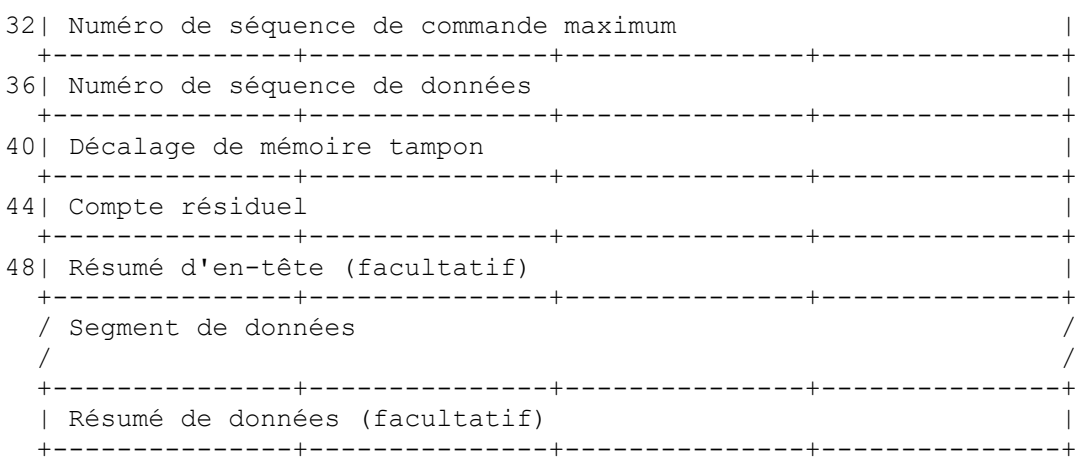
11.7 Data-Out et Data-In SCSI

La PDU SCSI Data-Out pour les opérations d'écriture a le format suivant :



La PDU SCSI Data-In pour les opérations de lecture a le format suivant :





L'état peut accompagner la dernière PDU Data-In si la commande ne s'est pas terminée par une exception (c'est-à-dire, l'état est "bon état" -- BON, CONDITION SATISFAITE, ou CONDITION INTERMEDIAIRE SATISFAITE). La présence de l'état (et d'un compte résiduel) est signalé via le bit fanion S. Bien que les cibles PUISSENT choisir d'envoyer même un état non exceptionnel dans des réponses séparées, les initiateurs DOIVENT prendre en charge un état non exceptionnel dans des PDU Data-In.

11.7.1 Bit F (Final)

Pour les données sortantes, ce bit est à 1 pour la dernière PDU de données non sollicitée ou la dernière PDU d'une séquence qui répond à un R2T.

Pour les données entrantes, ce bit est à 1 pour la dernière PDU de données d'entrée (lecture) d'une séquence. L'entrée peut être partagée en plusieurs séquences, ayant chacune son propre bit F. Partager le flux de données en séquences n'affecte pas le compte de DataSN sur les PDU Data-In. Il PEUT être utilisé comme indication de "changement de direction" pour les opérations bidirectionnelles qui ont besoin d'un tel changement.

Longueur de segment de données NE DOIT PAS excéder MaxRecvDataSegmentLength pour la direction dans laquelle il est envoyé, et le total de toutes les longueurs de segment de données de toutes les PDU d'une séquence NE DOIT PAS excéder Longueur maximale de salve (ou Longueur de première salve pour les données non sollicitées). Cependant, le nombre de PDU individuelles dans une séquence (ou au total) peut être supérieur au rapport de Longueur maximale de salve (ou Longueur de première salve) sur Longueur maximale de segment de données reçu (car les PDU peuvent être limitées en longueur par les capacités de l'envoyeur). Utiliser une longueur de segment de données de 0 peut augmenter au delà de ce qui est raisonnable avec le nombre de PDU et devrait donc être évité.

Pour les opérations bidirectionnelles, le bit F est à 1 pour la fin des séquences d'entrée et la fin des séquences de sortie.

11.7.2 Bit A (Accusé de réception)

Pour les sessions avec le niveau de récupération d'erreur = 1 ou plus, la cible règle ce bit à 1 pour indiquer qu'elle demande un accusé de réception positif de la part de l'initiateur pour les données reçues. La cible devrait utiliser avec modération le bit A ; elle ne PEUT établir le bit A qu'une fois tous les MaxBurstLength octets, ou sur la dernière PDU Data-In qui conclut l'entier transfert de données en lecture demandé pour la tâche du point de vue de la cible, et elle NE DOIT PAS le faire plus fréquemment. La cible NE DOIT PAS régler à 1 le bit A pour les sessions avec le niveau de récupération d'erreur = 0. L'initiateur DOIT ignorer le bit A réglé à 1 pour les sessions qui ont le niveau de récupération d'erreur = 0.

À réception d'une PDU Data-In PDU avec le bit A réglé à 1 sur une session avec le niveau de récupération d'erreur supérieur à 0, si il n'y a pas de trou dans les données de lecture jusqu'à cette PDU Data-In, l'initiateur DOIT produire un SNACK de type DataACK, sauf quand il est capable d'accuser réception de l'état pour la tâche immédiatement via le ExpStatSN sur d'autres PDU sortantes si l'état pour la tâche est aussi reçu. Dans le dernier cas (accusé de réception avec ExpStatSN) l'envoi d'un SNACK de type DataACK en réponse au bit A est FACULTATIF, mais si c'est fait, il ne doit pas être envoyé après l'accusé de reception d'état avec le ExpStatSN. Si l'initiateur a détecté des trous dans les données de lecture avant cette PDU Data-In, il DOIT différer de produire le SNACK de type DataACK jusqu'à ce que les trous soient remplis. Un initiateur NE DOIT PAS non plus accuser réception de l'état pour la tâche avant que ces trous soient remplis. Un accusé de réception d'état pour une tâche qui a généré les PDU Data-In est considéré par la cible comme un accusé de réception implicite des PDU Data-In si un tel accusé de réception était demandé par la cible.

11.7.3 Fanions (octet 1)

Le dernier paquet de données SCSI envoyé d'une cible à un initiateur pour une commande SCSI qui s'est achevée avec succès (avec un état de BON, CONDITION SATISFAITE, INTERMEDIAIRE, ou CONDITION INTERMEDIAIRE SATISFAITE) peut aussi facultativement contenir l'état pour le transfert de données. Dans ce cas, les données de sens ne peuvent pas être envoyées avec l'état de commande. Si la commande est achevée avec une erreur, la réponse et les données de sens DOIVENT alors être envoyées dans une PDU Réponse SCSI (c'est-à-dire, NE DOIT PAS être envoyée dans un paquet de données SCSI). Pour les commandes bidirectionnelles, l'état DOIT être envoyé dans une PDU Réponse SCSI.

bits 2-4 : réservés.
bits 5-6 : utilisés de la même façon qu'une réponse SCSI. Ces bits ne sont valides que quand S est réglé à 1. Pour les détails, voir le paragraphe 11.4.1.
bit 7 S (état) : établi pour indiquer que le champ État de commande contient l'état. Si ce bit est réglé à 1, le bit F DOIT aussi être réglé à 1.

Les champs Numéro de séquence d'état, État, et Compte résiduel n'ont de contenu significatif que si le bit S est réglé à 1. Les valeurs pour ces champs sont définies au paragraphe 11.4.

11.7.4 Étiquette de transfert de cible et LUN

Sur des données sortantes, l'étiquette de transfert de cible est fournie à la cible si le transfert respecte un R2T. Dans ce cas, le champ Étiquette de transfert de cible est la réplique de l'étiquette de transfert de cible fournie avec le R2T.

Sur des données entrantes,l'étiquette de transfert de cible et le LUN DOIVENT être fournis par la cible si le bit A est réglé à 1 ; autrement, ils sont réservés. L'étiquette de transfert de cible et le LUN sont copiés par l'initiateur dans le SNACK de type DataACK qu'il produit par suite de la réception d'une PDU SCSI Data-In avec le bit A réglé à 1.

Les valeurs d'étiquette de transfert de cible ne sont pas spécifiées par le présent protocole, sauf que la valeur 0xffffffff est réservée et signifie que l'étiquette de transfert de cible n'est pas fournie. Si l'étiquette de transfert de cible est fournie, le champ LUN DOIT alors contenir une valeur valide et être cohérent avec ce qui était spécifié dans la commande ; autrement, le champ LUN est réservé.

11.7.5 DataSN

Pour les PDU d'entrée (lecture) ou Data-In bidirectionnelles, le numéro de séquence de données est numéro de PDU d'entrée au sein du transfert de données pour la commande identifiée par l'étiquette de tâche d'initiateur.

Les PDU R2T et Data-In, dans le contexte de commandes bidirectionnelles, partagent la séquence de numérotation (voir le paragraphe 4.2.2.4).

Pour les PDU de données en sortie (écriture) le numéro de séquence de données est le numéro de PDU Data-Out au sein de la séquence de sortie actuelle. Soit la séquence de sortie actuelle est identifié par l'étiquette de tâche d'initiateur (pour les données non sollicitées) soit c'est une séquence de données générée pour un R2T (pour les données sollicitées par un R2T).

11.7.6 Décalage de mémoire tampon

Le champ Décalage de mémoire tampon contient le décalage des données de charge utile de cette PDU au sein du transfert de données complet. La somme du décalage et de la longueur de la mémoire tampon ne devrait pas excéder la longueur du transfert attendue pour la commande.

L'ordre des PDU de données au sein d'une séquence est déterminé par DataPDUInOrder. Quand il est réglé à Oui, cela signifie que les PDU doivent être en ordre de décalage de mémoire tampon croissant et les débordements sont interdits.

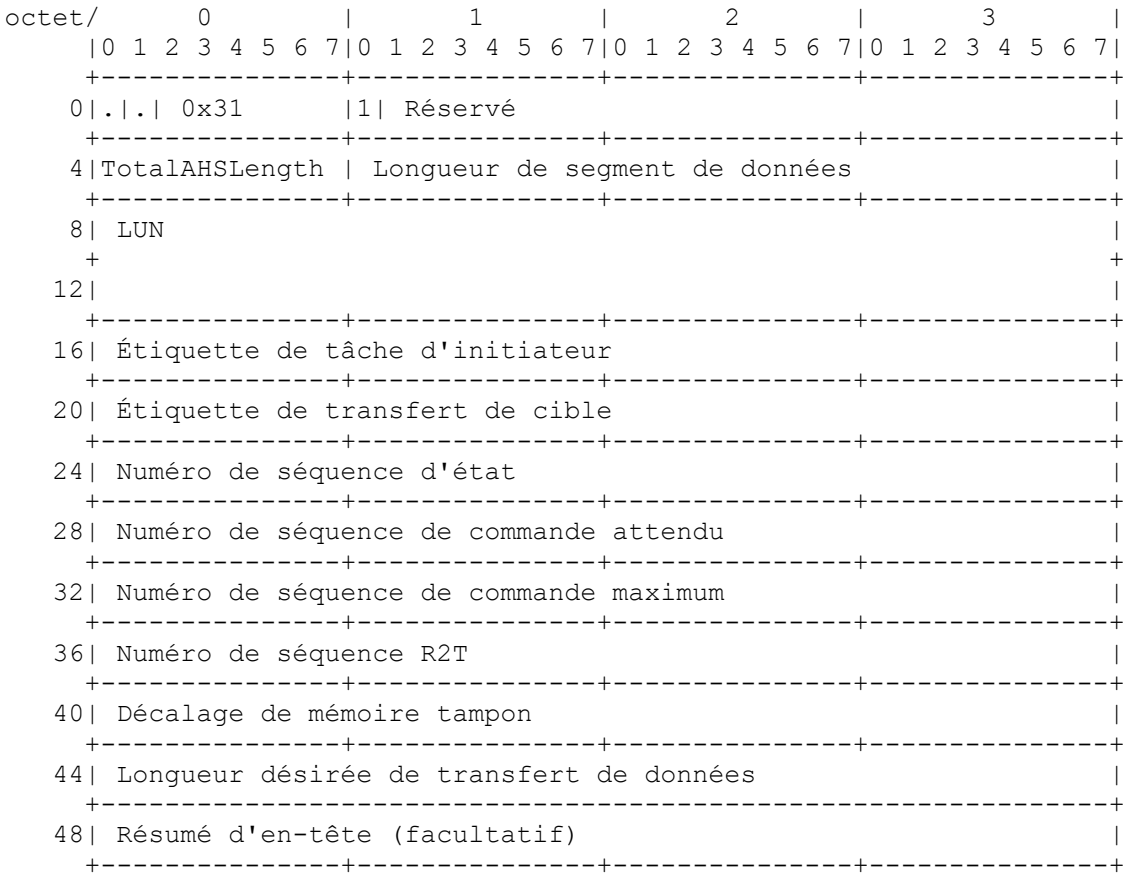
L'ordre entre séquences est déterminé par DataSequenceInOrder. Quand il est réglé à Oui, cela signifie que les séquences doivent être en ordre croissant de décalage de mémoire tampon, et que les débordements sont interdits.

11.7.7 DataSegmentLength

C'est la longueur de charge utile de données d'une PDU SCSI Data-In ou Data-Out. L'envoi de segments de données de longueur 0 devrait être évité, mais les initiateurs et les cibles DOIVENT être capables de recevoir correctement des segments de données de longueur 0.

Les segments de données de PDU Data-In et Data-Out DEVRAIENT être remplis à un nombre entier de mots de quatre octets (charge utile réelle) sauf si le bit F est réglé à 1.

11.8 Prêt au transfert (R2T, Ready To Transfer)



Quand un initiateur a soumis une commande SCSI avec des données qui passent de l'initiateur à la cible (écriture) la cible peut spécifier quels blocs de données elle est prête à recevoir. La cible peut demander que les blocs de données soient livrés dans l'ordre qui convient à la cible à ce moment particulier. Ces informations sont passées de la cible à l'initiateur dans la PDU Prêt au transfert (R2T, *Ready To Transfer*).
Afin de permettre des opérations d'écriture sans un R2T initial explicite, l'initiateur et la cible DOIVENT avoir négocié la clé InitialR2T à Non durant l'établissement.
Un R2T PEUT avoir pour réponse une ou plusieurs PDU SCSI Data-Out avec une étiquette de transfert de cible correspondante. Si un R2T a pour réponse une seule PDU Data-Out, le décalage de mémoire tampon dans la PDU de données DOIT être le même que celui spécifié par le R2T, et la longueur de données de la PDU de données DOIT être la même que la longueur désirée de transfert de données spécifiée dans le R2T. Si le R2T a pour réponse une séquence de PDU de données, le décalage et la longueur de mémoire tampon DOIVENT être dans la gamme spécifiée par le R2T, et la dernière PDU DOIT avoir le bit F réglé à 1. Si la dernière PDU (marquée du bit F) est reçue avant que la longueur désirée de transfert de données soit transférée, une cible PEUT choisir de rejeter cette PDU avec le code de cause "Erreur de protocole". DataPDUInOrder gouverne l'ordre des PDU Data-Out. Si DataPDUInOrder est réglé à Oui, les décalages et longueurs de mémoire tampon pour les PDU consécutives DOIVENT former une gamme continue qui ne se chevauche pas, et les PDU DOIVENT être envoyées en ordre de décalage croissant.
La cible peut envoyer plusieurs PDU R2T. Il peut donc y avoir un certain nombre de transferts de données en cours. Le nombre de PDU R2T en cours est limité par la valeur de la clé MaxOutstandingR2T négociée. Dans une tâche, les R2T en cours DOIVENT être exécutés par l'initiateur dans l'ordre dans lequel ils ont été reçus.
Les PDU R2T PEUVENT aussi être utilisées pour récupérer des PDU Data-Out. Un tel R2T (R2T de récupération) est généré par une cible quand elle détecte la perte d'une ou plusieurs PDU Data-Out dues à :
- une erreur de résumé
- une erreur de séquence
- une fin de temporisation de réception de séquence

Un R2T de récupération porte le prochain numéro de séquence de R2T non utilisé mais demande une partie de la salve de données ou la salve entière qu'un R2T antérieur (avec un R2TSN inférieur) avait déjà demandée.

DataSequenceInOrder gouverne l'ordre du décalage de mémoire tampon dans les R2T consécutifs. Si DataSequenceInOrder est Oui, les R2T consécutifs DOIVENT alors se référer à des gammes continues non chevauchantes, sauf pour les R2T de récupération.

11.8.1 TotalAHSLength et DataSegmentLength

Pour cette PDU, TotalAHSLength et DataSegmentLength DOIVENT être à 0.

11.8.2 R2TSN

R2TSN est le numéro de PDU d'entrée de PDU R2T au sein de la commande identifiée par l'étiquette de tâche d'initiateur. Pour les commandes bidirectionnelles, les PDU R2T et Data-In partagent la séquence de numérotation de PDU d'entrée (voir le paragraphe 4.2.2.4).

11.8.3 Numéro de séquence d'état

Le champ StatSN va contenir le prochain numéro de séquence d'état *(StatSN)*. Le Numéro de séquence d'état pour cette connexion n'est pas augmenté après l'envoi de cette PDU.

11.8.4 Longueur de transfert de données désirée et décalage de mémoire tampon

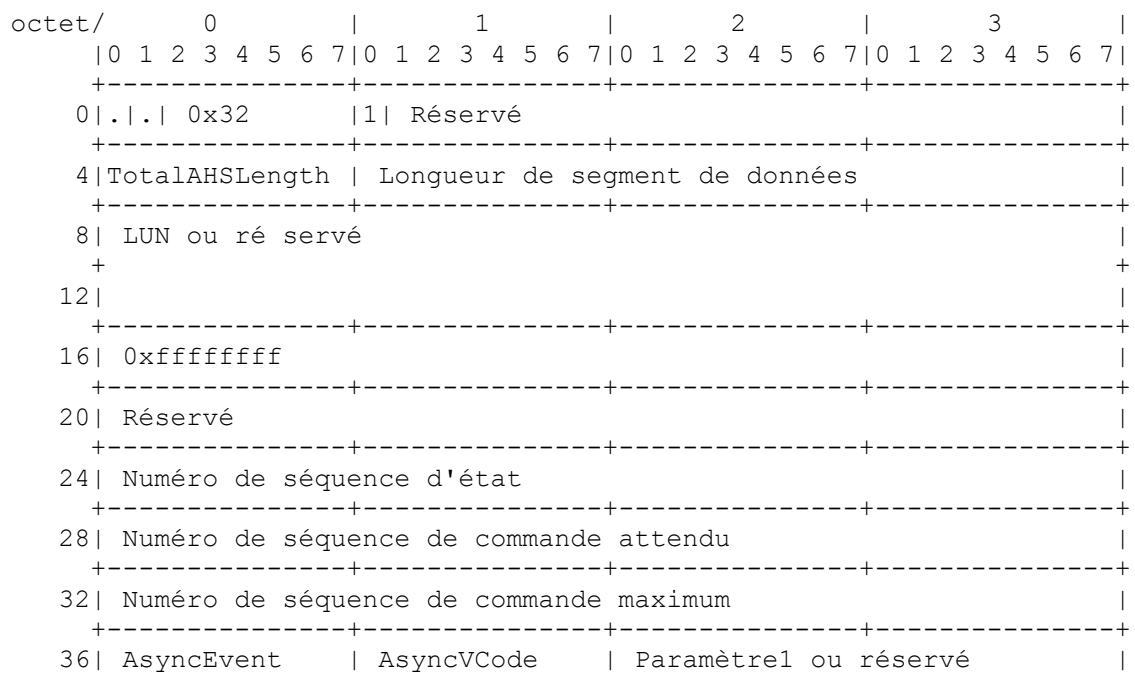
La cible spécifie combien d'octets elle veut que l'initiateur envoie à cause de cette PDU R2T. La cible peut demander les données de l'initiateur en plusieurs tronçons, pas nécessairement dans l'ordre d'origine des données. La cible spcécifie donc aussi un décalage de mémoire tampon qui indique le point auquel le transfert de données devrait commencer, par rapport au commencement du transfert de données total. La longueur désirée de transfert de données NE DOIT PAS être 0 et NE DOIT PAS excéder MaxBurstLength.

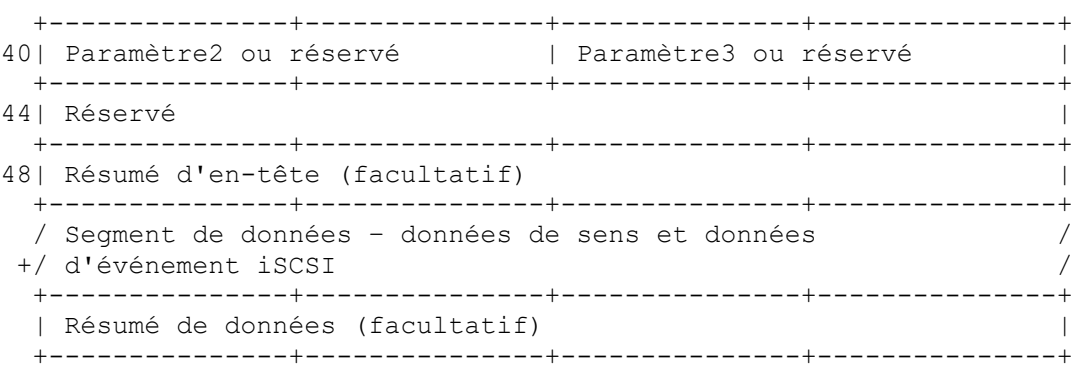
11.8.5 Étiquette de transfert de cible

La cible alloue sa propre étiquette à chaque demande R2T qu'elle envoie à l'initiateur. Cette étiquette peut être utilisée par la cible pour identifier facilement les données qu'elle reçoit. L'étiquette de transfert de cible et le LUN sont copiés dans les PDU de données sortantes et ne sont utilisés que par la cible. Il n'y a pas de règle de protocole sur l'étiquette de transfert de cible, sauf que la valeur 0xffffffff est réservée et NE DOIT PAS être envoyée par une cible dans un R2T.

11.9 Message Asynchrone

Un message Asynchrone peut être envoyé de la cible à l'initiateur sans correspondre à une commande particulière. La cible spécifie la raison de l'événement et des données de sens.





Certains messages asynchrones sont en rapport strict avec iSCSI, tandis que d'autres se rapportent à SCSI [SAM2].

Numéro de séquence d'état compte cette PDU comme un événement digne d'un accusé de réception (le numéro de séquence d'état est augmenté) ce qui permet la synchronisation de l'état à l'initiateur et à la cible.

11.9.1 AsyncEvent

Les codes utilisés pour les messages iSCSI Asynchrone (événement) sont :

0 (événement SCSI asynchrone) : un événement SCSI asynchrone est rapporté dans les données de sens. Les données de sens qui accompagnent le rapport, dans le segment de données, identifient la condition. L'envoi d'un événement SCSI ("rapport d'un événement asynchrone" dans la terminologie SCSI) dépend de si la cible prend en charge le rapport des événements asynchrones SCSI (voir [SAM2]) comme indiqué dans les données INQUIRY standard (voir [SPC3]). Son utilisation peut être activée par des paramètres dans la page de mode de contrôle SCSI (voir [SPC3]).

1 (demande de désétablissement) : la cible demande le désétablissement. Ce message asynchrone DOIT être envoyé sur la même connexion que celle qui demande le désétablissement. L'initiateur DOIT satisfaire cette demande en produisant un “Logout” aussitôt que possible mais pas plus tard que Paramètre3 secondes. L'initiateur DOIT envoyer un “Logout” avec un code de cause de "clore la connexion" ou "clore la session" pour clore toutes les connexions. Une fois que ce message est reçu, l'initiateur NE DEVRAIT PAS produire de nouvelles commandes iSCSI sur la connexion à désétablir. La cible PEUT rejeter toute nouvelle demande d'entrée/sortie qu'elle reçoit après ce message avec le code de cause "Attente de désétablissement". Si l'initiateur ne se désétablit pas dans les Paramètre3 secondes, la cible devrait envoyer une PDU Async avec le code d'événement iSCSI "Connexion abandonnée" si possible ou simplement terminer la connexion de transport. Paramètre1 et Paramètre2 sont réservés.

2 (Notification d'abandon de connection) : la cible indique qu'elle va abandonner la connexion. Le champ Paramètre1 indique le CID de la connexion qui va être abandonnée. Le champ Paramètre2 (Temps d'attente) indique, en secondes, le délai minimum d'attente avant de tenter une reconnexion ou réallocation. Le champ Paramètre3 (Temps de conservation) indique la durée maximale permise pour la réallocation des commandes après l'attente initiale (dans Paramètre2). Si l'initiateur ne tente pas de reconnecter et/ou réallouer les commandes en attente dans le délai spécifié par Paramètre3, ou si Paramètre3 est 0, la cible va terminer toutes les commandes en instance sur cette connexion. Dans ce cas, aucune autre réponse ne devrait être attendue de la cible pour les commandes en instance sur cette connexion. Une valeur de 0 pour Paramètre2 indique que la reconnexion peut être tentée immédiatement.

3 (Notification d'abandon de session) : la cible indique qu'elle va abandonner toutes les connexions de cette session. Le champ Paramètre1 est réservé. Le champ Paramètre2 (Temps d'attente) indique, en secondes, le délai minimum d'attente avant de tenter de reconnecter. Le champ Paramètre3 (Temps de conservation) indique le délai maximum permis pour réallouer les commandes après l'attente initiale (dans Paramètre2). Si l'initiateur ne tente pas de reconnecter et/ou réallouer les commandes en instance dans le délai spécifié par Paramètre3, ou si Paramètre3 est 0, la session se termine. Dans ce cas, la cible va terminer toutes les commandes en instance dans cette session ; aucune autre réponse ne devrait être attendue de la cible pour les commandes en instance dans cette session. Une valeur de 0 pour Paramètre2 indique que la reconnexion peut être tentée immédiatement.

4 (Demande de négociation) : la cible demande la négociation de paramètres sur cette connexion. L'initiateur DOIT satisfaire cette demande en produisant une demande Text (qui peut être vide) sur la même connexion aussitôt que possible, mais pas plus tard que Paramètre3 secondes, sauf si une demande Text est déjà en cours sur la connexion, ou en produisant une demande de désétablissement. Si l'initiateur ne produit pas une demande Text, la cible peut répéter le message Asynchrone qui demande la négociation de paramètres.

5 (Terminaison de tâche) : toutes les tâches actives pour une LU avec un champ LUN qui correspond dans la PDU Message asynchrone sont terminées. La couche iSCSI de l'initiateur receveur répond à ce message avec les étapes suivantes, dans cet ordre :

- Arrêt des transferts Data-Out sur cette connexion pour toutes les TTT actives pour les LUN affectés cités dans la PDU Message asynchrone,
- Accuser réception du numéro de séquence d'état de la PDU Message asynchrone via une PDU NOP-Out avec ITT=0xffffffff (c'est-à-dire, une saveur non ping) tout en copiant le champ LUN du message asynchrone à NOP-Out.

Cette valeur de AsyncEvent NE DOIT cependant PAS être utilisée sur une session iSCSI sauf si la nouvelle clé text TaskReporting définie au paragraphe 13.23 a été négociée à FastAbort sur la session.

248-255 (unique au fabricant) : événement iSCSI spécifique du fabricant. AsyncVCode détaille de code de fabricant, et des données PEUVENT accompagner le rapport.

Tous les autres codes d'événement sont non alloués.

11.9.2 AsyncVCode

AsyncVCode est un code de détails spécifique du fabricant qui n'est valide que si le champ AsyncEvent indique un événement spécifique de fabricant. Autrement, il est réservé.

11.9.3 LUN

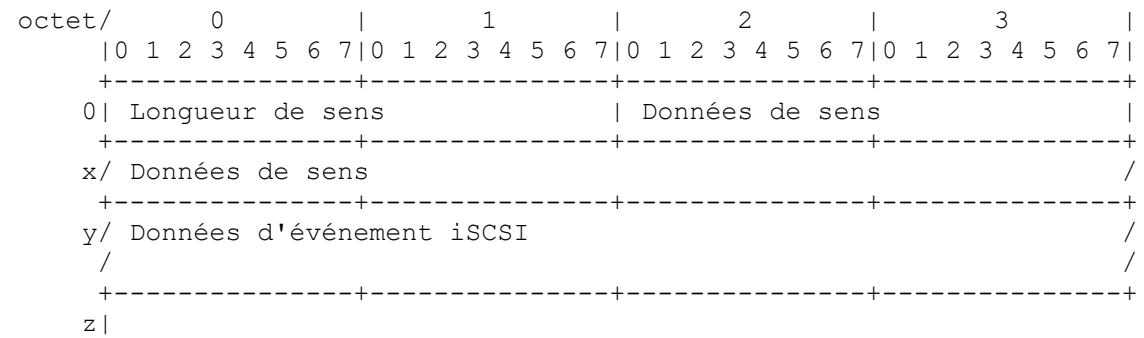
Le champ LUN DOIT être valide si Événement asynchrone *(AsyncEvent)* est 0. Autrement, ce champ est réservé.

11.9.4 Données de sens et données d'événement iSCSI

Pour un événement SCSI, ces données accompagnent le rapport dans le segment de données et identifient la condition.

Pour un événement iSCSI, des données uniques au fabricant supplémentaires PEUVENT accompagner l'événement asynchrone. Les initiateurs PEUVENT ignorer les données quand elles ne sont pas comprises, tout en traitant le reste de la PDU.

Si Longueur du segment de données *(DataSegmentLength)* n'est pas 0, le format de Segment de données *(DataSegment)* est le suivant :

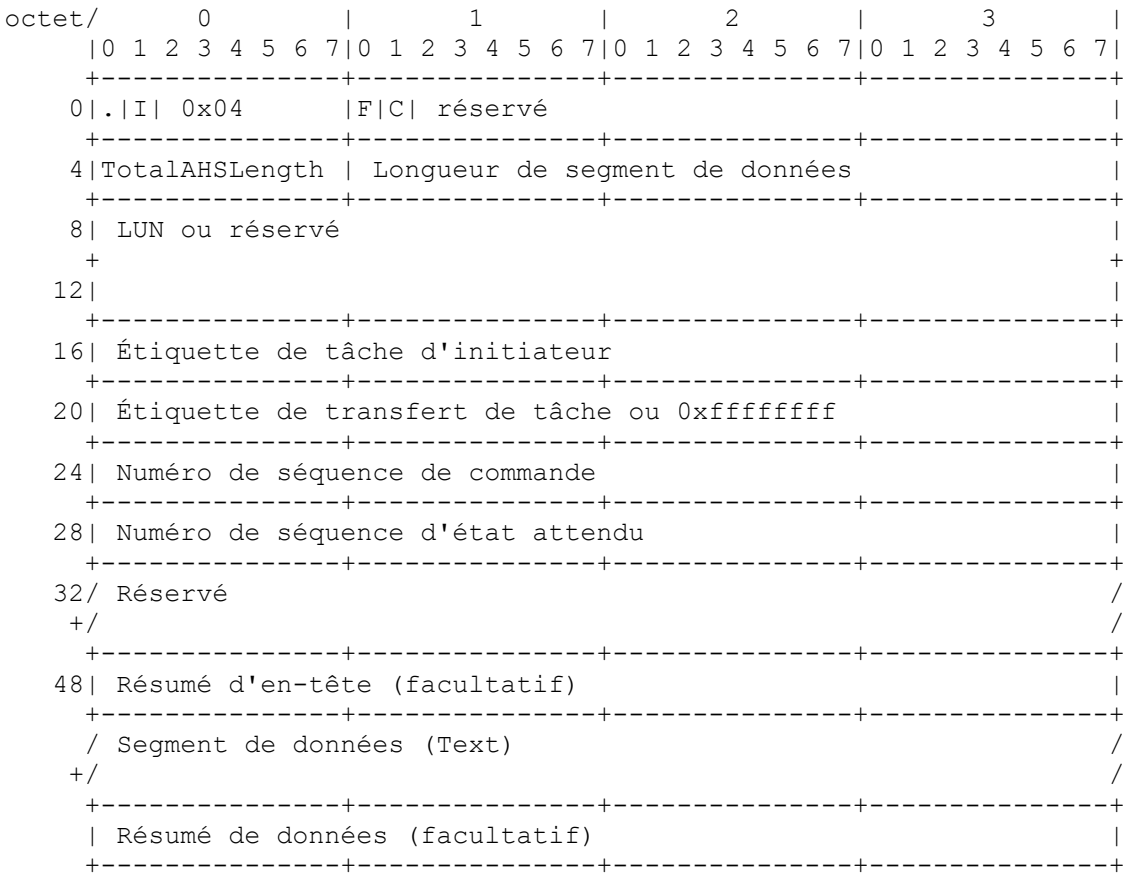


11.9.4.1 Longueur de sens *(SenseLength)*

C'est la longueur des données de sens. Quand le champ Données de sens est vide (par exemple, l'événement n'est pas un événement SCSI), SenseLength est à 0.

11.10 Demande Text

La demande Text est fournie pour permettre l'échange d'informations et de futures extensions. Cela permet à l'initiateur d'informer une cible de ses capacités ou de demander des opérations particulières.



Un initiateur NE DOIT PAS avoir plus d'une demande Text en instance sur une connexion à tout instant.

Sur une défaillance de connexion, un initiateur doit soit interrompre explicitement toute tâche active de négociation de texte allégeante, soit causer la terminaison implicite de cette tâche par la cible.

11.10.1 Bit F (Final)

Lorsque réglé à 1, ce bit indique que c'est la dernière ou la seule demande Text dans une séquence de demandes Text ; autrement, il indique que plus de demandes Text vont suivre.

11.10.2 Bit C (Continue)

Lorsque réglé à 1, ce bit indique que le texte (ensemble de paires clé=valeur) dans cette demande Text n'est pas complet (il va se continuer sur les demandes Text suivantes) ; autrement, il indique que cette demande Text termine un ensemble de paires clé=valeur. Une demande Text avec le bit C réglé à 1 DOIT avoir le bit F réglé à 0.

11.10.3 Étiquette de tâche d'initiateur

C'est l'identifiant alloué par l'initiateur pour cette demande Text. Si la commande est envoyée au titre d'une séquence de demandes et réponses Text, l'étiquette de tâche d'initiateur DOIT être la même pour toutes les demandes au sein de la séquence (similaire aux commandes SCSI reliées). Le bit I pour toutes les demandes d'une séquence DOIT aussi être le même.

11.10.4 Étiquette de transfert de cible

Quand l'étiquette de transfert de cible est réglée à la valeur réservée de 0xffffffff, cela dit à la cible que c'est une nouvelle demande, et la cible réinitialise tout état interne associé à l'étiquette de tâche d'initiateur (elle réinitialise l'état de négociation en cours).

La cible règle règle l'étiquette de transfert de cible dans une réponse Text à une valeur autre que la valeur réservée 0xffffffff chaque fois qu'elle indique qu'il y a plus de données à envoyer ou plus d'opérations à effectuer qui sont associées à l'étiquette de tâche d'initiateur spécifiée. Elle DOIT faire ainsi chaque fois qu'elle règle le bit F à 0 dans la réponse. En

copiant l'étiquette de transfert de cible de la réponse dans la prochaine demande Text, l'initiateur dit à la cible de continuer l'opération pour l'étiquette de tâche d'initiateur spécifiée. L'initiateur DOIT ignorer l'étiquette de transfert de cible dans la réponse Text quand le bit F est réglé à 1.

Ce mécanisme permet à l'initiateur et à la cible de transférer de grandes quantiés de données de texte sur une séquence d'échanges de commande de texte - réponse de texte ou d'effectuer des séquences de négociation étendues.

Si l'étiquette de transfert de cible n'est pas 0xffffffff, le champ LUN DOIT être envoyé par la cible dans la réponse Text.

Une cible PEUT réinitialiser son état de négociation interne si un échange est bloqué par l'initiateur pendant longtemps ou si elle est à bout de ressources.

Les longues réponses de texte sont traitées comme indiqué par l'exemple suivant :
I->T Text SendTargets=All (F = 1, TTT = 0xffffffff)
T->I Text <partie 1> (F = 0, TTT = 0x12345678)
I->T Text <vide> (F = 1, TTT = 0x12345678)
T->I Text <partie 2> (F = 0, TTT = 0x12345678)
I->T Text <vide> (F = 1, TTT = 0x12345678)
...
T->I Text <partie n> (F = 1, TTT = 0xffffffff)

11.10.5 Text

Les longueurs de données d'une demande Text NE DOIVENT PAS excéder le paramètre iSCSI MaxRecvDataSegmentLength de la cible (un paramètre qui est négocié par connexion et par direction). Le format text est spécifié au paragraphe 6.2.
Les Sections 12 et 13 font la liste de quelques paires clé=valeur Text de base, dont certaines peuvent être utilisées dans les demandes/réponses d'établissement et certaines dans les demandes/réponses Text.

Une paire clé=valeur peut s'étendre au delà des limites de demande ou réponse Text. Une paire clé=valeur peut commencer dans une PDU et continuer sur la suivante. En d'autres termes, la fin d'une PDU ne signale pas nécessairement la fin d'une paire clé=valeur.

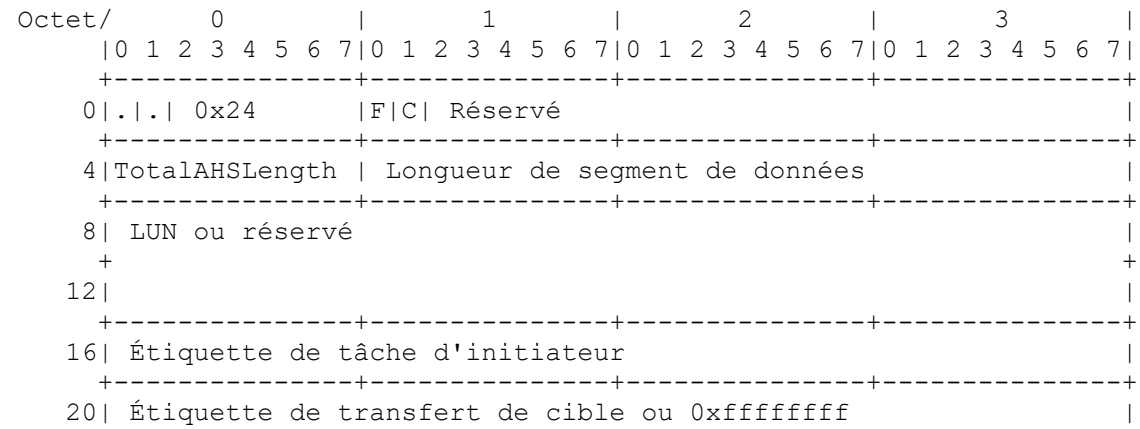
La cible répond en renvoyant sa réponse à l'initiateur. Le format de la réponse Text est similaire au format de la demande Text. La réponse Text PEUT se référer aux paires clé=valeur présentées dans une demande Text antérieure, et le texte dans la demande peut se référer à des réponses antérieures.

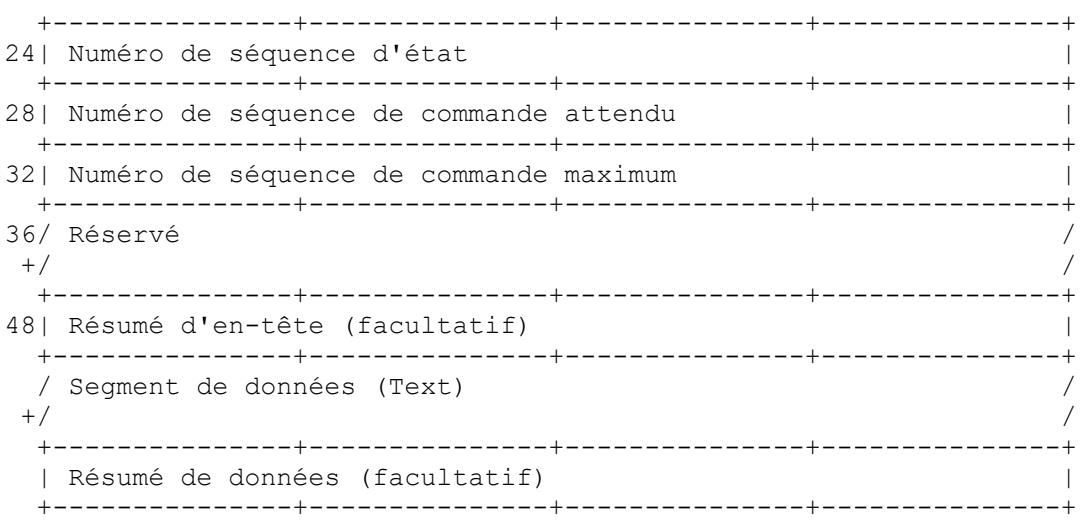
Le paragraphe 6.2 détaille les règles pour les demandes et réponses Text.

Les opérations de texte sont généralement destinées aux réglages/négociations de paramètres mais peuvent aussi être utilisées pour effectuer des opérations qui durent longtemps. Les opérations Text qui durent longtemps devraient être placées dans leur propre demande Text.

11.11 Réponse Text

La PDU Réponse Text contient les réponses de la cible aux demandes Text de l'initiateur. Le format du champ Text correspond à celui de la demande Text.





11.11.1 Bit F (Final)

Lorsque réglé à 1 dans une réponse à une demande Text avec le bit Final réglé à 1, le bit F indique que la cible a fini l'opération entière. Autrement, si réglé à 0 dans la réponse à une demande Text avec le bit Final réglé à 1, il indique que la cible a plus de travail à faire (invite à une demande Text suivante). Une réponse Text avec le bit F réglé à 1 en réponse à une demande Text qui a le bit F réglé à 0 est une erreur de protocole.

Une réponse Text avec le bit F réglé à 1 NE DOIT PAS contenir de paires clé=valeur qui peuvent requérir des réponses supplémentaires de l'initiateur.

Une réponse Text avec le bit F réglé à 1 DOIT avoir un champ Étiquette de transfert de cible réglé à la valeur réservée 0xffffffff.

Une réponse Text avec le bit F réglé à 0 DOIT avoir un champ Étiquette de transfert de cible réglé à une valeur autre que la valeur réservée 0xffffffff.

11.11.2 Bit C (Continue)

Lorsque réglé à 1, ce bit indique que le texte (ensemble de paires clé=valeur) dans cette réponse Text n'est pas complet (il va se continuer sur les réponses Text suivantes) ; autrement, il indique que cette réponse Text termine un ensemble de paires clé=valeur. Une réponse Text avec le bit C réglé à 1 DOIT avoir le bit F réglé à 0.

11.11.3 Étiquette de tâche d'initiateur

L'étiquette de tâche d'initiateur correspond à l'étiquette utilisée dans la demande Text initiale.

11.11.4 Étiquette de transfert de cible

Lorsque une cible a plus de travail à faire (par exemple, elle ne peut pas transférer toutes les données de texte restantes dans une seule réponse Text ou doit continuer la négociation) et a assez de ressources pour poursuivre, elle DOIT régler l'étiquette de transfert de cible à une valeur autre que la valeur réservée 0xffffffff. Autrement, l'étiquette de transfert de cible DOIT être réglée à 0xffffffff.

Lorsque l'étiquette de transfert de cible n'est pas 0xffffffff, le champ LUN peut être significatif.

L'initiateur DOIT copier l'étiquette de transfert de cible et le LUN dans sa prochaine demande pour indiquer qu'il veut le reste des données.

Lorsque la cible reçoit une demande Text avec l'étiquette de transfert de cible réglée à la valeur réservée 0xffffffff, elle réinitialise ses informations internes (elle réinitialise l'état) associées à l'étiquette de tâche d'initiateur concernée (recommence la négociation).

Lorsque une cible ne peut pas finir l'opération dans une seule réponse Text et n'a pas assez de ressources pour continuer, elle rejette la demande Text avec le code de rejet approprié.

Une cible peut réinitialiser son état interne associé à une étiquette de tâche d'initiateur (l'état de la négociation en cours) comme exprimé par l'étiquette de transfert de cible si l'initiateur échoue à continuer l'échange pendant un certain temps. La cible peut rejetter les demandes Text suivantes avec l'étiquette de transfert de cible réglée à la valeur "périmé".

11.11.5 StatSN

La variable StatSN *(numéro de séquence d'état)* de la cible est augmentée par chaque réponse Text envoyée.

11.11.6 Réponse Text Data

La longueur de données d'une réponse Text NE DOIT PAS excéder le paramètre iSCSI MaxRecvDataSegmentLength de l'initiateur (paramètre négocié par connexion et par direction).

Le texte dans la réponse Données de texte est gouverné par les mêmes règles que le texte dans la demande Données de texte (voir le paragraphe 11.11.2).

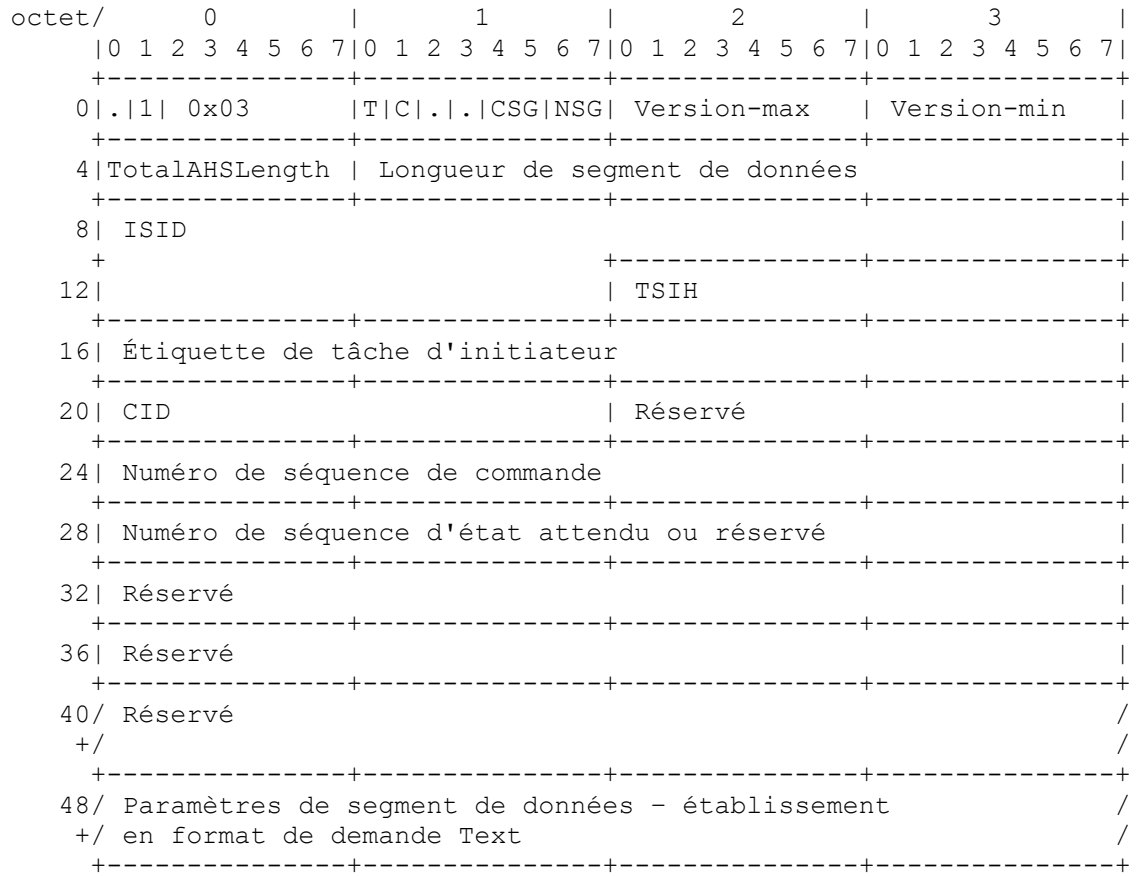
Bien que l'initiateur soit la partie demandeuse et qu'il contrôle l'initiation et la terminaison des demandes/réponses, la cible peut offrir des paires clé=valeur de son propre chef au titre d'une séquence et pas seulement en réponse à l'initiateur.

11.12 Demande d'établissement (Login)

Après l'établissement d'une connexion TCP entre un initiateur et une cible, l'initiateur DOIT commencer une phase d'établissement (Login) pour obtenir l'accès aux ressources de la cible.

La phase Login (voir le paragraphe 6.3) consiste en une séquence de demandes Login et de réponses Login qui portent la même étiquette de tâche d'initiateur.

Les demandes Login sont toujours considérées comme immédiates.



11.12.1 Bit T (Transit)

Lorsque réglé à 1, ce bit indique que l'initiateur est prêt à transiter à l'étape suivante.

Si le bit T est réglé à 1 et si le NSG est réglé à FullFeaturePhase, cela indique ausi que l'initiateur est prêt pour la réponse finale Login (voir le paragraphe 6.3).

11.12.2 Bit C (Continue)

Lorsque réglé à 1, ce bit indique que le texte (ensemble de paires clé=valeur) dans cette demande Login n'est pas complet (il va se continuer sur des demandes Login suivantes) ; autrement, il indique que cette demande Login termine un ensemble de paires clé=valeur. Une demande Login avec le bit C réglé à 1 DOIT avoir le bit T réglé à 0.

11.12.3 CSG et NSG

Par ces champs – Étape actuelle (CSG, *Current Stage*) et Prochaine étape (NSG, *Next Stage*) – les demandes et réponse de négociation d'établissement sont associées à une étape spécifique dans la session (Négociation de sécurité, Négociation d'établissement opérationnel, Phase de pleines caractéristiques) et peuvent indiquer la prochaine étape à laquelle ils veulent passer (voir le paragraphe 6.3). La valeur de prochaine étape n'est valide que quand le bit T est 1; autrement, elle est réservée.

Les codes d'étapes sont :
0 : Négociation de sécurité
1 : Négociation d'établissement opérationnel
3 : Phase de pleines caractéristiques
Tous les autres codes sont réservés.

11.12.4 Version

Le numéro de version pour le présent document est 0x00. Donc, Version-min et Version-max DOIVENT tous deux être réglés à 0x00.

11.12.4.1 Version-max

Version-max indique le numéro de version maximum pris en charge.

Toutes les demandes d'établissement au sein de la phase Login DOIVENT porter la même Version-max.

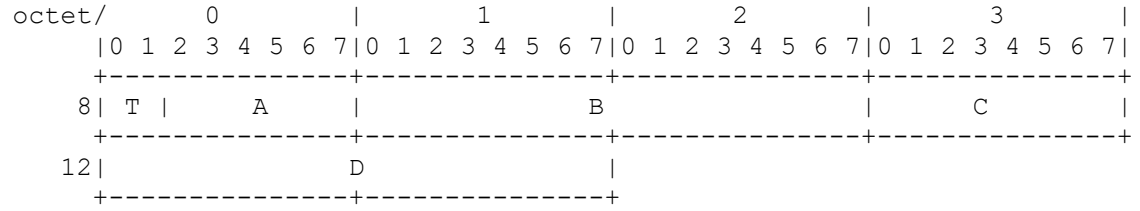
La cible DOIT utiliser la valeur présentée dans la première demande Login.

11.12.4.2 Version-min

Toutes les demandes Login au sein de la phase Login DOIVENT porter la même Version-min. La cible DOIT utiliser la valeur présentée dans la première demande Login.

11.12.5 ISID

C'est un composant défini par l'initiateur de l'identifiant de session et il est structuré comme suit (voir les détails au paragraphe 10.1.1) :



Le champ T identifie le format et l'usage de A, B, C, et D comme indiqué ci-dessous :
T
00b : Format OUI
A et B : OUI de 22 bits ; (les bits I/G et U/L sont omis)
C et D : Qualificatif de 24 bits

01b : Format EN (numéro d'entreprise IANA)
A : réservé
B et C : EN (numéro d'entreprise IANA)
D : Qualificatif
10b : "aléatoire"
A : réservé
B et C : aléatoire
D : Qualificatif
11b : A, B, C, et D : réservé

Pour les valeurs de champ T 00b et 01b, une combinaison de A et B (pour 00b) ou B et C (pour 01b) identifie le fabricant ou organisation dont le composant (logiciel ou matériel) génère cet ISID. Un fabricant ou organisation avec un ou plusieurs OUI, ou un ou plusieurs numéros d'entreprise, DOIT utiliser au moins un de ces numéros et choisir la valeur appropriée pour le champ T quand ses composants génèrent des ISID. Un OUI ou un EN DOIT être établi dans les champs correspondants dans l'ordre des octets du réseau (octets gros boutiens).

Si le champ T est 10b, B et C sont réglés à une valeur d'entier non signé aléatoire de 24 bits dans l'ordre des octets du réseau (octets gros boutiensn). Voir dans la [RFC3721] comment cela affecte le principe de "réutilisation prudente".

Le champ Qualificatif est une valeur d'entier non signé de 16 bits ou 24 bits qui donne une gamme de valeurs possibles pour l'ISID au sein de l'espace de noms choisi. Il peut être réglé à toute valeur dans les contraintes spécifiées dans le protocole iSCSI (voir les paragraphes 4.4.3 et 10.1.1).

La valeur de champ T de 11b est réservée.

Si l'ISID est déduit de quelque chose qui est alloué à un adapateur ou interface de matériel par un fabricant comme une valeur pré réglée par défaut, il DOIT être configurable à une valeur allouée conformément au comportement d'accès SCSI désiré par le système dans lequel il est installé (voir aux paragraphes 10.1.1 et 10.1.2). L'ISID résultant DOIT aussi être persistent à travers les cycles d'alimentation, réamorçages, changements de cartes, etc.

11.12.6 TSIH

Le descripteur identifiant de session cible (TSIH, *Target Session Identifying Handle*) doit être établi dans la première demande Login. La valeur réservée 0 DOIT être utilisée sur la première connexion pour une nouvelle session. Autrement, le TSIH envoyé par la cible à la conclusion de l'établissement réussi de la première connexion pour cette session DOIT être utilisé. Le TSIH identifie à la cible la session existante associée pour cette nouvelle connexion. Toutes les demandes Login au sein d'une phase Login DOIVENT porter le même TSIH. La cible DOIT vérifier la valeur présentée avec la première demande Login et agir comme spécifié au paragraphe 6.3.1.

11.12.7 Identifiant de connexion (CID)

Le CID fournit un identifiant univoque pour cette connexion au sein de la session. Toutes les demandes Login au sein de la phase Login DOIVENT porter le même CID. La cible DOIT utiliser la valeur présentée dans la première demande Login.
Une demande Login avec un TSIH non zéro et un CID égal à celui d'une connexion existante implique un désétablissement de la connexion suivi par un établissement (voir le paragraphe 6.3.4). Pour les détails concernant la demande de désétablissement implicite, voir le paragraphe 11.14.

11.12.8 Numéro de séquence de commande (CmdSN)

Le CmdSN est soit le numéro de séquence de commande initial d'une session (pour la première demande Login d'une session -- le login "de tête") ou le numéro de séquence de commande dans le flux des commandes si l'établissement est pour une nouvelle connexion dans une session existante.

Exemples :
- Établissement sur une connexion de tête : si l'établissement de tête porte le CmdSN 123, toutes les autres demandes d'établissement dans la même phase Login portent le CmdSN 123, et la première commande non immédiate dans la phase de pleines caractéristique porte aussi le CmdSN 123.
- Étblissement sur une connexion autre que de tête : si le CmdSN courant au moment du premier établissement sur la connexion est 500, cette PDU porte alors CmdSN=500. Les demandes Login suivantes qui sont nécessaires pour achever cette phase d'établissement peuvent porter un CmdSN supérieur à 500 si des demandes non immédiates qui ont été produites sur d'autres connexions dans la même session augmentent le CmdSN.

Si la demande Login est une demande Login de tête, la cible DOIT utiliser la valeur présentée dans le CmdSN comme la valeur de la cible pour le numéro de séquence de commande attendu *(ExpCmdSN)*.

11.12.9 Numéro de séquence d'état attendu (ExpStatSN)

Pour la première demande Login sur une connexion, c'est le ExpStatSN pour l'ancienne connexion, et ce champ n'est valide que si la demande Login recommence une connexion (voir au paragraphe 6.3.4).

Pour les demandes Login suivantes, il est utilisé pour accuser réception des réponses Login avec leurs valeurs croissantes de numéro de séquence d'état.

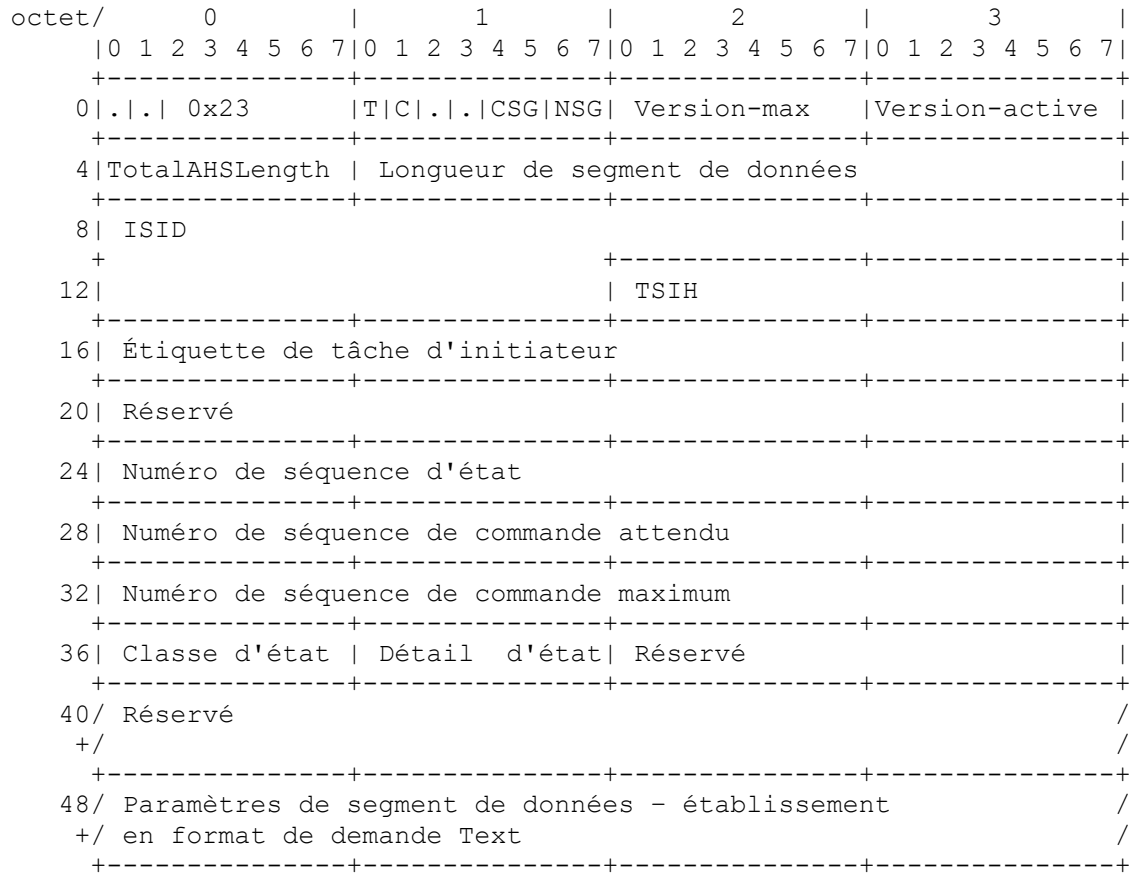
11.12.10 Paramètres d'établissement

L'initiateur DOIT fournir des paramètres de base afin de permettre à la cible de déterminer si l'initiateur peut utiliser les ressources de la cible et les paramètres de texte initiaux pour l'échange de sécurité.

Toutes les règles spécifiées au paragraphe 11.10.5 pour la demande Text tiennent aussi pour les demandes d'établissement. Les clés et leur explication sont données à la Section 12 (clés de négociation de sécurité) et à la Section 13 (clés de négociation des paramètres de fonctionnement). Toutes les clés mentionnées à la Section 13, sauf les formats d'extension X, DOIVENT être prises en charge par les initiateurs et les cibles iSCSI. Les clés mentionnées à la Section 12 ont seulement besoin d'être prises en charge quand la fonction à laquelle elles se réfèrent est de mise en œuvre obligatoire.

11.13 Réponse d'établissement

La réponse d'établissement indique les progrès et/ou la fin de la phase d'établissement *(Login)*.



11.13.1 Version-max

C'est le plus haut numéro de version pris en charge par la cible.
Toutes les réponses Login au sein de la phase Login DOIVENT porter le même Version-max.
L'initiateur DOIT utiliser la valeur présentée comme une réponse à la première demande Login.

11.13.2 Version-active

Version-active indique la plus haute version prise en charge par la cible et l'initiateur. Si la cible ne prend pas en charge une version dans la gamme spécifiée par l'initiateur, la cible rejette l'établissement et ce champ indique la plus basse version prise en charge par la cible.
Toutes les réponses Login dans la phase Login DOIVENT porter le même Version-active.
L'initiateur DOIT utiliser la valeur présentée comme la réponse à la première demande Login.

11.13.3 TSIH

Le TSIH est le descripteur identifiant de session alloué par la cible. Son format et son contenu internes ne sont pas définis dans le protocole, sauf la valeur 0, qui est réservée. À l'exception de la réponse Login finale dans une nouvelle session, ce champ devrait être réglé au TSIH fourni par l'initiateur dans la demande Login. Pour une nouvelle session, la cible DOIT générer un TSIH non zéro et SEULEMENT le retourner dans la réponse finale de Login (voir le paragraphe 6.3).

11.13.4 Numéro de séquence d'état

Pour la première réponse Login (la réponse à la première demande Login) c'est le numéro de séquence d'état de début pour la connexion. La prochaine réponse de quelque sorte qu'elle soit – incluant la prochaine réponse Login, si il en est, dans la même phase Login – va porter ce numéro + 1. Ce champ n'est valide que si la classe d'état est 0.

11.13.5 Classe d'état et détails d'état

L'état retourné dans une réponse Login indique l'état d'exécution de la phase Login. L'état inclut :
la classe d'état
les détails d'état
Une classe d'état de 0 indique la réussite.

Une classe d'état non zéro indique une exception. Dans ce cas, la classe d'état est suffisante pour qu'un simple initiateur l'utilise quand il traite les exceptions, sans avoir à regarder le détail d'état.

Le détail d'état permet un traitement d'exception d'une granularité plus fine pour des initiateurs plus sophistiqués et pour de meilleures informations à enregistrer.

Les classes d'état sont les suivantes :

0 : Succès - indique que la cible iSCSI a bien reçu, compris et accepté la demande. Les champs de numérotation (Numéro de séquence d'état, Numéro de séquence de commande attendu, Numéro de séquence de commande maximum) ne sont valides que si Classe d'état est 0.

1 : Redirection - indique que l'initiateur doit accomplir d'autres actions pour terminer la demande . C'est généralement dû à ce que la cible s'est déplacée à une adresse différente. Toutes les réponses de classe d'état redirection DOIVENT retourner un ou plusieurs paramètres de clé de texte du type "CibleAdresse", qui indique la nouvelle adresse de la cible. Une réponse Redirection PEUT être produite par une cible avant ou après l'achèvement d'une négociation de sécurité si une négociation de sécurité est requise. Une redirection DEVRAIT être acceptée par un initiateur, même sans que la cible achève une négociation de sécurité si une négociation de sécurité est requise, et DOIT être acceptée par l'initiateur après l'achèvement de la négociation de sécurité si celle-ci est requise.

2 : Erreur de l'initiateur (pas une erreur formelle) - indique que l'initiateur a très probablement causé l'erreur. Cela PEUT être dû à une demande d'une ressource pour laquelle l'initiateur n'a pas la permission. La demande ne devrait pas être réessayée.

3 : Erreur de la cible - indique que la cible ne voit pas d'erreur dans la demande Login de l'initiateur mais est actuellement incapable de satisfaire la demande. L'initiateur peut réessayer la même demande Login ultérieurement.

Le tableau ci-dessous montre tous les codes d'état actuellement alloués. Les codes sont en hexadécimal; le premier octeet est la classe d'état, et le second octet est le détail d'état.

État	Code (hex)	Description
Réussite	0000	L'établissement est bien traité (*1).
Déplacement temporaire de la cible	0101	Le nom de cible iSCSI (ITN) demandé a quitté temporairement l'adresse fournie.
Déplacement permanent de la cible	0102	L'ITN demandé a quitté en permanence l'adresse fournie.
Erreur de l'initiateur	0200	Diverses erreurs d'initiateur iSCSI.
Échec d'authentification	0201	L'initiateur n'a pas pu être authentifié ou l'authentification par la cible n'est pas prise en charge.

Échec d'autorisation	0202	L'initiateur n'a pas la permission d'accéder à cette cible.
Pas trouvé	0203	L'ITN demandé n'existe pas à cette adresse.
Cible supprimée	0204	L'ITN demandé a été supprimé, et aucune adresse n'est fournie.
Version non prise en charge	0205	La gamme de version iSCSI demandée n'est pas prise en charge par la cible.
Trop de connexions	0206	Trop de connexions sur ce SSID.
Paramètre manquant	0207	Paramètres manquants(par exemple, nom d'initiateur iSCSI et/ou nom de cible).
Ne peut être inclus dans la session	0208	La cible ne prend pas en charge l'extension de session sur cette connexion (adresse).
Type de session non pris en charge	0209	La cible ne prend pas en charge ce type de session ou pas de cet initiateur.
Session non existante	020a	Tentative d'ajouter une connexion à une session non existante.
Invalide durant l'établissement	020b	Type de demande invalide durant l'établissement.
Erreur de cible	0300	Erreur de matériel ou logiciel cible.
Service indisponible	0301	Le service ou cible iSCSI n'est pas actuellement opérationnel.
Plus de ressources	0302	La cible a des ressources insuffisantes de session, connexion, ou autres.

(*1) Si le bit T de réponse est réglé à 1 dans la demande et dans la réponse correspondante, et si le NSG est réglé à FullFeaturePhase dans la demande et dans la réponse correspondante, la phase Login est finie, et l'initiateur peut procéder à produire des commandes SCSI.

Si la classe d'état n'est pas 0, l'initiateur et la cible DOIVENT clore la connexion TCP.

Si la cible souhaite rejeter la demande Login pour plus d'une raison, elle devrait retourner la raison principale pour le rejet.

11.13.6 Bit T (Transit)

Le bit T est réglé à 1 comme indicateur de fin d'étape. Si le bit T est réglé à 1 et si le NSG est réglé à FullFeaturePhase, ceci est alors aussi la réponse finale d'établissement (voir au paragraphe 6.3). Un bit T de 0 indique une réponse "partielle", qui signifie "plus de négociation nécessaire".

Une réponse Login avec le bit T réglé à 1 NE DOIT PAS contenir de paire clé=valeur qui peut exiger des réponses supplémentaires de l'initiateur dans la même étape.

Si la classe d'état est 0, le bit T NE DOIT PAS être réglé à 1 si le bit T dans la demande était réglé à 0.

11.13.7 Bit C (Continue)

Lorsque réglé à 1, ce bit indique que le texte (ensemble de paires clé=valeur) dans cette réponse Login n'est pas complet (il va continuer sur les réponses Login suivantes) ; autrement, il indique que cette réponse Login termine un ensemble de paires clé=valeur. Une réponse Login avec le bit C réglé à 1 DOIT avoir le bit T bit réglé à 0.

11.13.8 Paramètres d'établissement

La cible DOIT fournir des paramètres de base afin de permettre à l'initiateur de déterminer si il est connecté à l'accès correct et aux paramètres initiaux de texte pour l'échange de sécurité.

Toutes les règles spécifiées au paragraphe 11.11.6 pour les réponses Text tiennent aussi pour les réponses d'établissement. Les clés et leur explication figurent à la Section 12 (clés de négociation de sécurité) et à la Section 13 (clés de négociation des paramètres de fonctionnement). Toutes les clés mentionnées à la Section 13, sauf pour les formats d'extension X, DOIVENT être prises en charge par les initiateurs et les cibles iSCSI. Les clés mentionnées à la Section 12 ont seulement besoin d'être prises en charge quand la fonction à laquelle elles se réfèrent est de mise en œuvre obligatoire.

11.14 Demande de désétablissement (Logout)

La demande de désétablissement est utilisée pour effectuer une clôture contrôlée d'une connexion.

Un initiateur PEUT utiliser une demande de désétablissement pour supprimer une connexion d'une session ou pour clore une session entière.

Après l'envoi de la PDU “Logout Request”, un initiateur NE DOIT PAS envoyer de nouvelle demande iSCSI sur la connexion qui ferme. Si la demande de désétablissement est destinée à clore la session, de nouvelles demandes iSCSI NE DOIVENT être envoyées sur aucune des connexions qui participent à la session.

À réception d'une demande de désétablissement avec le code de cause "clore la connexion" ou "clore la session", la cible DOIT terminer toutes les commandes en instance, qu'elles soient acquittées via le numéro de séquence de commande attendu ou non, sur cette, respectivement, connexion ou session,.

Lorsque elle reçoit une demande de désétablissement avec le code de cause "supprimer la connexion pour récupération", la cible DOIT éliminer toutes les demandes non encore acquittées via le numéro de séquence de commande attendu qui ont été produites sur la connexion spécifiée et suspendre tous les transferts de données/états/R2T au nom des commandes en instance sur la connexion spécifiée.

La cible produit alors la réponse Logout et clôt la demi connexion TCP (envoie FIN). Après la réception de la réponse Logout et avoir tenté de recevoir le FIN (si c'est encore possible) l'initiateur DOIT clore complètement la connexion en cours de désétablissement. Pour les commandes terminées, aucune réponse supplémentaire ne devrait être attendue.

Un Logout pour un CID peut être effectué sur une connexion de transport différente quand la connexion TCP pour le CID a déjà été terminée. Dans ce cas, seule une "clôture" logique de la connexion iSCSI pour le CID est impliquée avec un Logout.

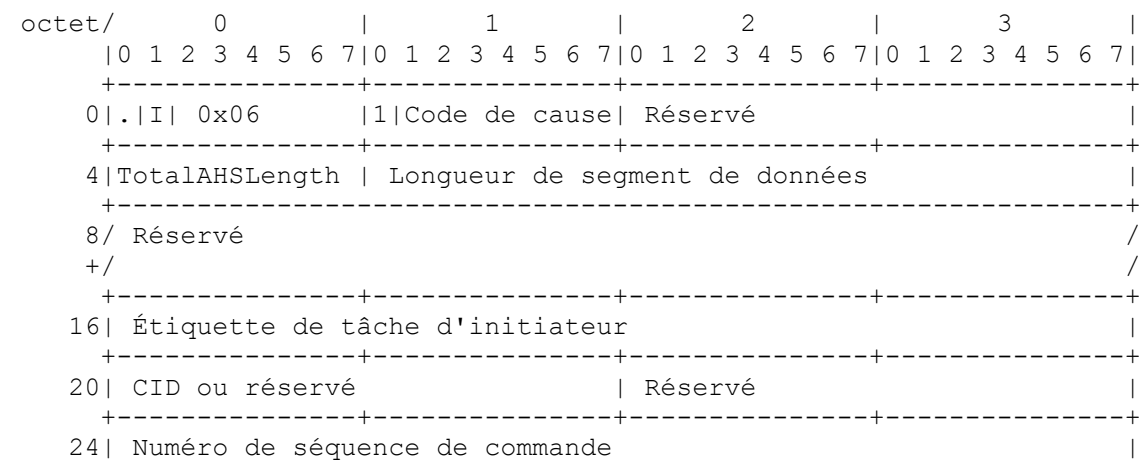
Toutes les commandes qui n'étaient pas terminées ou non achevées (avec état) et acquittées lorsque la connexion est complètement close peuvent être réallouées à une nouvelle connexion si la cible prend en charge la récupération de connexion.

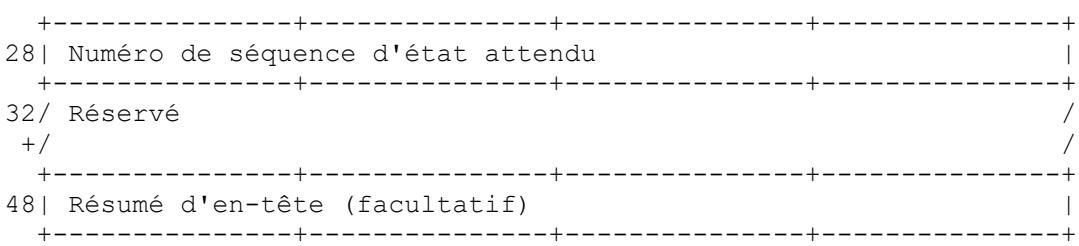
Si un initiateur entend commencer la récupération pour une connexion défaillante, il DOIT utiliser la demande de désétablissement pour "nettoyer" l'extrémité cible d'une connexion défaillante et permettre de commencer la récupération, ou utiliser la demande d'établissement avec un TSIH non zéro et le même CID sur une nouvelle connexion pour le même effet. Dans les sessions avec une seule connexion, la connexion peut être close et ensuite une nouvelle connexion rouverte. Un établissement de réinstallation de connexion peut être utilisé pour la récupération (voir au paragraphe 6.3.4).

Un achèvement réussi d'une demande de désétablissement avec le code de cause "clore la connexion" ou "supprimer la connexion pour récupération" résulte à la cible en l'élimination des commandes non acquittées reçues sur la connexion objet du désétablissement. Ce sont des commandes qui sont arrivées sur la connexion en cours de désétablissement mais qui n'ont pas été livrées à SCSI parce qu'une ou plusieurs commandes avec un plus petit numéro de séquence de commande n'ont pas été reçues par iSCSI (voir au paragraphe 4.2.2.1). Les trous résultants dans les numéros de séquence de commande devront être traités par une récupération appropriée (voir la Section 7) sauf si la session est aussi close.

Toute la discussion sur le désétablissement dans ce paragraphe est aussi applicable pour le désétablissement implicite réalisé au moyen d'une réinstallation de connexion ou de session. Lorsque une demande Login effectue un Logout implicite, celui-ci est effectué comme si il avait les codes de cause spécifiés ci-dessous :

Code de cause	**Type de Logout implicite**
0	réinstallation de session
1	réinstallation de connexion quand le niveau de récupération d'erreur opérationnel est inférieur à 2
2	réinstallation de connexion quand le niveau de récupération d'erreur opérationnel est égal à 2





11.14.1 Code de cause

Le champ Code de cause indique la raison du désétablissement comme suit :
0 : clore la session. Toutes les commandes associées à la session (si il en est) sont terminées.
1 : clore la connexion. Toutes les commandes associées à la connexion (si il en est) sont terminées.
2 : retirer la connexion pour récupération. La connexion est close, et toutes les commandes associées à elle, si il en est, doivent être prêtes à une nouvelle allégeance.
Toutes les autres valeurs sont réservées.

11.14.2 Longueur AHS totale et Longueur de segment de données

Pour cette PDU, Longueur AHS totale et Longueur de segment de données DOIVENT être 0.

11.14.3 CID

C'est l'identifiant de connexion de la connexion à clore (incluant de clore le flux TCP). Ce champ n'est valide que si le code de cause n'est pas "clore la session".

11.14.4 Numéro de séquence d'état attendu (ExpStatSN)

C'est la valeur du dernier ExpStatSN pour la connexion à clore.

11.14.5 Terminaison implicite des tâches

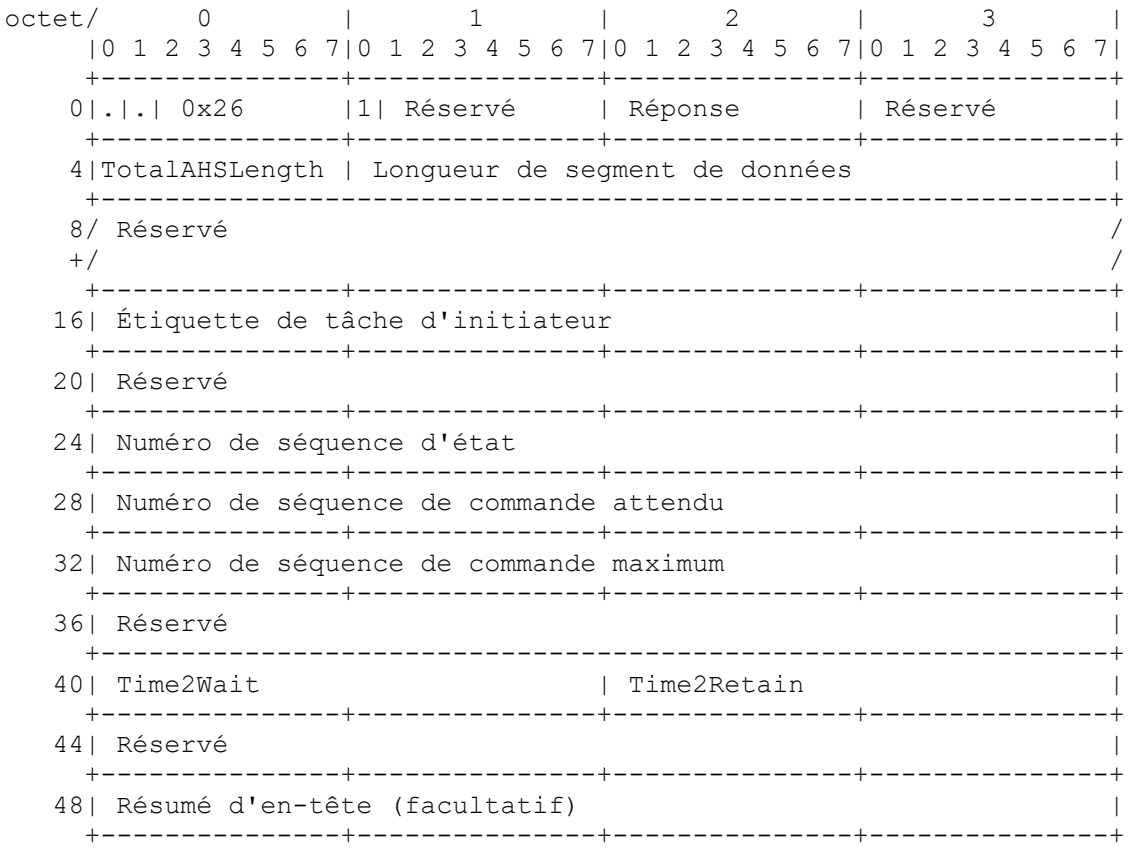
Une cible termine implicitement les tâches actives du fait du protocole iSCSI dans les cas suivants :
a) Lorsque une connexion est implicitement ou explicitement désétablie avec le code de cause "clore la connexion" et qu'il y a des tâches actives allégeantes à cette connexion.
b) Lorsque une connexion a une défaillance et que finalement son état arrive en fin de temporisation (transition d'état M1 au paragraphe 8.2.2) et qu'il y a des tâches actives allégeantes à cette connexion.
c) Lorsque un désétablissement de récupération réussi est effectué alors qu'il y a des tâches actives allégeantes à cette connexion et que ces tâches arrivent finalement en fin de temporisation après les périodes Time2Wait et Time2Retain sans réallocation d'allégeance.
d) Lorsque une connexion est implicitement ou explicitement désétablie avec le code de cause "clore la session" et qu'il y a des tâches actives dans cette session.

Si les tâches terminées dans un des cas ci-dessus sont des tâches SCSI, elles doivent être terminées en interne comme si elles étaient dans l'état Vérifier la condition. Cet état n'a de signification que pour le traitement approprié de l'état interne SCSI et les effets collatéraux SCSI à l'égard de l'ordre, parce que cet état n'est jamais communiqué à l'initiateur comme un état de terminaison. Cependant, des actions supplémentaires peuvent devoir être entreprises au niveau SCSI, selon le contexte SCSI comme défini par les normes SCSI (par exemple, commandes mises en file d'attente et ACA, UA pour la prochaine commande sur le nexus I_T dans les cas a), b), et c) ci-dessus). Après la fin des tâches, la cible DOIT rapporter une condition “Unit Attention” sur la prochaine commande traitée sur une connexion pour chaque nexus I_T_L affecté avec l'état de Vérifier la condition, la valeur ASC/ASCQ de 47h/7Fh ("certaines commandes ont été nettoyées par un événement du protocole iSCSI"), etc.; voir [SPC3].

11.15 Réponse Désétablissement (Logout)

La réponse Logout est utilisée par la cible pour indiquer si l'opération de nettoyage pour la ou les connexions s'est terminée.

Après un désétablissement, la connexion TCP référencée par le CID DOIT être close aux deux extrémités (ou toutes les connexions doivent être closes si la raison du désétablissement était la clôture de session).



11.15.1 Réponse

Les réglages du champ Réponse sont les suivants :
0 : connexion ou session close avec succès.
1 : CID non trouvé.
2 : Récupération de connexion non prise en charge (c'est-à-dire, le code de cause de désétablissement était "retirer la connexion pour réccupération" et la cible ne la prend pas en charge comme indiqué par le niveau de récupération d'erreur opérationnel).
3 – Échec de nettoyage pour diverses raisons.

11.15.2 Longueur AHS totale et Longueur de segment de données

Pour cette PDU, Longueur AHS totale et Longueur de segment de données DOIVENT être 0.

11.15.3 Time2Wait

Si le code de réponse de désétablissement est 0 et si le niveau de récupération d'erreur opérationnel est 2, c'est la durée minimum, en secondes, d'attente avant de tenter une réallocation de tâche. Si le code de réponse de désétablissement est 0 et si le niveau de récupération d'erreur opérationnel est moins de 2, ce champ est à ignorer.

Ce champ est invalide si le code de réponse de désétablissement est 1.

Si le code de réponse de désétablissement est 2 ou 3, ce champ spécifie la durée minimum d'attente avant une nouvelle tentative implicite ou explicite de désétablissement.

Si Time2Wait est 0, la réallocation ou un nouveau désétablissement peut être tenté immédiatement.

11.15.4 Time2Retain

Si le code de réponse de désétablissement est 0 et si le niveau de récupération d'erreur opérationnel est 2, c'est la durée maximum, en secondes, après l'attente initiale (Time2Wait) que la cible attend la réallocation d'allégeance pour toute tâche active, après quoi l'état de la tâche est éliminé. Si le code de réponse de désétablissement est 0 et si le niveau de récupération d'erreur opérationnel est moins de 2, ce champ est à ignorer.

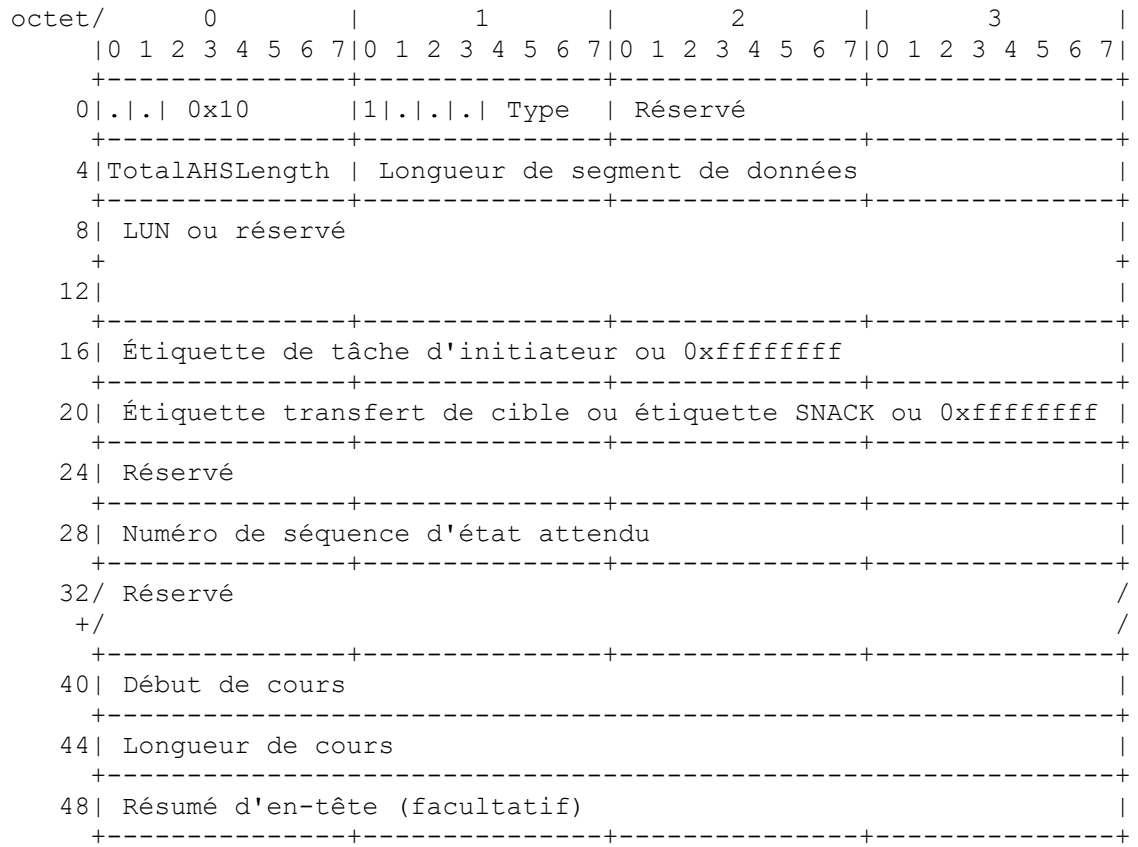
Ce champ est invalide si le code de réponse de désétablissement est 1.

Si le code de réponse de désétablissement est 2 ou 3, ce champ spécifie la durée maximum, en secondes, après l'attente initiale (Time2Wait) que la cible attend un nouveau désétablissement implicite ou explicite.

Si c'est la dernière connexion d'une session, tout l'état de session est éliminé après Time2Retain.

Si Time2Retain est 0, la cible a déjà éliminé l'état de la connexion (et éventuellement la session) avec les états de la tâche. Aucne réallocation ni désétablissement n'est requis dans ce cas.

11.16 Demande SNACK



Si la mise en œuvre prend en charge un niveau de récupération d'erreur supérieur à zéro, elle DOIT prendre en charge tous les types de SNACK.

Le SNACK est utilisé par l'initiateur pour demander la retransmission de PDU de réponses, données ou R2T numérotées à partir de la cible. La demande SNACK indique les réponses numérotées ou les “cours” de données dont la retransmission est demandée, où un cours commence par le premier numéro de séquence d'état, numéro de séquence de données, ou numéro de séquence de R2T dont la retransmission est demandée et indique le nombre de PDU d'état, de données, ou de R2T demandées, incluant la première. 0 a une signification spéciale lorsque utilisé comme numéro et longueur de début :
- Lorsque utilisé dans Longueur de cours, il signifie des PDU commençant à l'initial.
- Lorsque utilisé dans les deux Début de cours et Longueur de cours, il signifie toutes les PDU non acquittées.

Les réponses numérotées ou R2T demandés par un SNACK DOIVENT être livrés comme des répliques exactes de ceux que la cible a transmis à l'origine, excepté pour les champs Numéro de séquence de commande attendu, Numéro de séquence de commande maximum, et Numéro de séquence de données attendu, qui DOIVENT porter les valeurs courantes. Les R2T demandés par un SNACK DOIVENT aussi porter la valeur actuelle du numéro de séquence d'état.

Les PDU Données numérotées demandées par un SNACK Données DOIVENT être livrées comme des répliques exactes de celles que la cible a transmises à l'origine, sauf pour les champs Numéro de séquence de commande attendu et Numéro de séquence de commande maximum, qui DOIVENT porter les valeurs courantes, et sauf pour une resegmentation (voir le paragraphe 11.16.3).

Tout SNACK qui demande une réponse, données, ou R2T numérotés qui n'a pas été envoyé par la cible ou a déjà été acquitté par l'initiateur DOIT être rejeté avec un code de cause de "Erreur de protocole".

11.16.1 Type

Ce champ code la fonction SNACK comme suit :
0 : SNACK Données/R2T, demande la retransmission d'une ou plusieurs PDU Data-In ou R2T.
1 : SNACK d'état, demande la retransmission d'une ou plusieurs réponses numérotées.
2 : DataACK, accuse positivement la réception de PDU Data-In.
3 : SNACK R-Data, demande la retransmission de PDU Data-In avec une possible resegmentation et étiquetage d'état.
Toutes les autres valeurs sont réservées.

Un SNACK Données/R2T, un SNACK d'état, ou un SNACK R-Data pour une commande DOIT précéder l'accusé de réception d'état pour la commande concernée.

11.16.2 Accusé de rédeption de données

Si un initiateur opère au niveau de récupération d'erreur 1 ou plus, il DOIT produire un SNACK de type DataACK après réception d'une PDU Data-In avec le bit A réglé à 1. Cependant, si l'initiateur a détecté des trous dans la séquence d'entrée, il DOIT différer de produire le SNACK de type DataACK jusqu'à ce que les trous soient bouchés. Un initiateur PEUT ignorer le bit A si il estime que le bit a été établi aggressivement par la cible (c'est-à-dire, avant que soit atteinte la limite de Longueur maximale de salve).
Le DataACK est utilisé pour libérer des ressources à la cible et non pour demander ou impliquer une retransmission de données.
Un initiateur NE DOIT PAS demander de retransmission pour des données déjà acquittées.

11.16.3 Resegmentation

Si la longueur maximale de segment de données reçues de l'initiateur a changé entre la transmission d'origine et le moment où l'initiateur demande la retransmission, l'initiateur DOIT produire un SNACK R-Data (voir le paragraphe 11.16.1). Avec le SNACK R-Data, l'initiateur indique qu'il élimine toutes les données non acquittées et attend que la cible les renvoie. Il attend aussi la resegmentation. Dans ce cas, les PDU Data-In retransmises PEUVENT être différentes de celles originellement envoyées afin de refléter les changements de longueur maximale de segment de données reçues. Leur numéro de séquence de données *(DataSN)* commence avec le Début de cours du dernier accusé de réception de données *(DataACK)* reçu par la cible si il en a été reçu ; autrement, il commence à 0 et est augmente de 1 pour chaque PDU Data-In renvoyée.

Une cible qui a reçu un SNACK R-Data DOIT retourner une réponse SCSI qui contient une copie du champ Étiquette SNACK du SNACK R-Data dans le champ SNACK Étiquette de réponse SCSI comme sa dernière ou seule réponse. Par exemple, si il a déjà envoyé une réponse contenant une autre valeur dans le champ Étiquette SNACK, ou avait l'état inclus dans la dernière PDU Data-In, il doit envoyer une nouvelle PDU Réponse SCSI. Si une cible envoie plus d'une PDU Réponse SCSI due à cette règle, toutes les PDU Réponse SCSI doivent porter le même numéro de séquence d'état (voir le paragraphe 11.4.4). Si un initiateur tente de récupérer une réponse SCSI perdue (avec un SNACK d'état ; voir le paragraphe 11.16.1) alors que plus d'une réponse a été envoyée, la cible va envoyer la réponse SCSI avec le dernier contenu connu de la cible, incluant la dernière étiquette SNACK pour la commande.

Pour les considérations de réallocation d'allégeance d'une tâche à une connexion avec une longueur maximale de segment de données reçues différente, se référer au paragraphe 7.2.2.

11.16.4 Étiquette de tâche d'initiateur

Pour un SNACK d'état et un accusé de réception de données, l'étiquette de tâche d'initiateur DOIT être réglée à la valeur réservée 0xffffffff. Dans tous les autres cas, le champ Étiquette de tâche d'initiateur DOIT être réglé à l'étiquette de tâche d'initiateur de la commande référencée.

11.16.5 Étiquette de transfert de cible ou étiquette SNACK

Pour un SNACK R-Data, ce champ DOIT contenir une valeur différente de 0 ou 0xffffffff et est unique pour la tâche (identifiée par l'étiquette de tâche d'initiateur). Cette valeur DOIT être copiée par la cible iSCSI dans la dernière ou seule PDU Réponse SCSI qu'elle produit pour la commande.

Pour un accusé de réception de données (DataACK), l'étiquette de transfert de cible DOIT contenir une copie de l'étiquette de transfert de cible et du LUN fournis avec la PDU SCSI Data-In avec le bit A réglé à 1.
Dans tous les autres cas, le champ Étiquette de transfert de cible DOIT être réglé à la valeur réservée 0xffffffff.

11.16.6 Début de cours (BegRun)

Ce champ indique le le numéro de séquence de données, le numéro de séquence R2T, ou le numéro de séquence d'état de la première PDU dont la retransmission est demandée (SNACK Données/R2T et d'état) ou le prochain numéro de séquence de données attendu (SNACK DataACK).

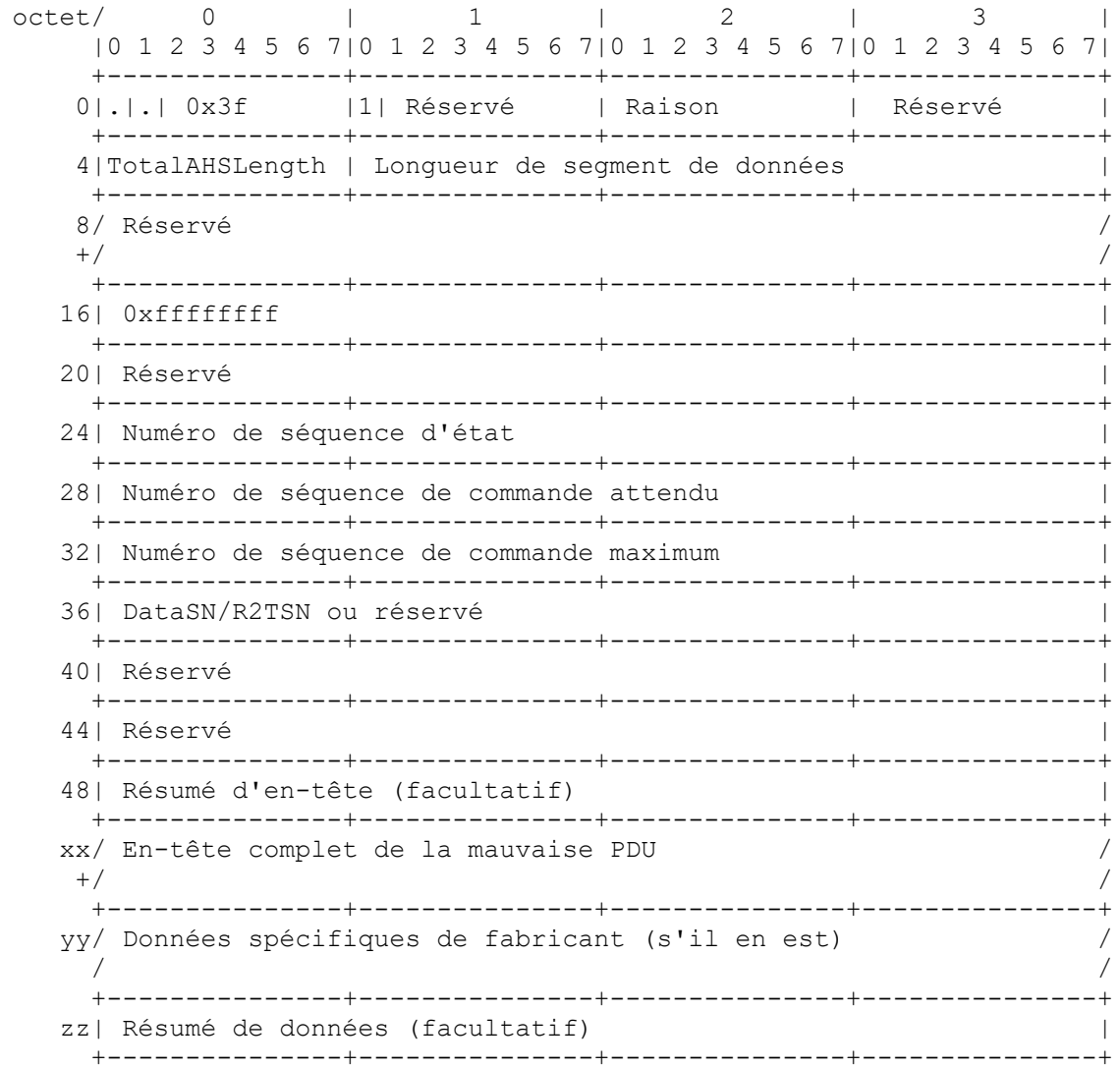
Un début de cours de 0, utilisé en conjonction avec une longueur de cours de 0, signifie "renvoyer toutes les PDU Data-In, R2T ou Réponse non acquittées".

BegRun DOIT être 0 pour un SNACK R-Data.

11.16.7 Longueur de cours (RunLength)

Ce champ indique le nombre de PDU dont la retransmission est demandée.
Une longueur de cours de 0 signale que toutes les PDU Data-In, R2T, ou Réponse portant les numéros égaux ou supérieurs à Début de cours doivent être renvoyées.
Longueur de cours DOIT aussi être 0 pour un SNACK DataACK en plus du SNACK R-Data.

11.17 Rejet



Rejet est utilisé pour indiquer une condition d'erreur iSCSI (protocole, option non prise en charge, etc.).

11.17.1 Raison

La raison du rejet est codée comme suit :

Code (hex)	Explication	La PDU originale peut-elle être renvoyée ?
0x01	réservé	non
0x02	Erreur de résumé de données (charge utile)	oui (Note 1)
0x03	Rejet SNACK	oui
0x04	Erreur de protocole (par exemple, demande SNACK pour un état qui a déjà été acquitté)	non
0x05	Commande non prise en charge	non
0x06	Rejet de commande immédiate – trop de commandes immédiates	oui
0x07	Tâche en cours	non
0x08	Accusé de réception de données invalide	non
0x09	Champ de PDU invalide	non (Note 2)
0x0a	Rejet d'opération longue – ne peut pas générer d'étiquette de transfert de cible – plus de ressources	oui
0x0b	Déconseillé; NE DOIT PAS être utilisé	N/A (Note 3)
0x0c	Attente de désétablissement	non

Note 1 : Pour iSCSI, la retransmission de PDU Data-Out n'est faite que si la cible demande la retransmission avec un R2T de récupération. Cependant, si c'est l'erreur de résumé de données sur des données immédiates, l'initiateur peut choisir de retransmettre la PDU entière, incluant les données immédiates.

Note 2 : Une cible devrait utiliser ce code de cause pour toutes les valeurs invalides des champs de PDU qui sont destinés à décrire une tâche, une réponse, ou un transfert de données. Des exemples sont des TTT/ITT invalides, un décalage de mémoire tampon, un LUN qualifiant un TTT, et un numéro de séquence invalide dans un SNACK.

Note 3 : Le code de cause 0x0b ("Réinitialisation de négociation") comme défini au paragraphe 10.17.1 de la [RFC3720] est déconseillé et NE DOIT PAS être utilisé par les mises en œuvre. Une mise en œuvre qui reçoit le code de cause 0x0b DOIT le traiter comme un échec de négociation qui termine la phase d'établissement et la connexion TCP, comme spécifié au paragraphe 7.12.

Toutes les autres valeurs de cause sont non allouées.

Dans tous les cas où une tâche SCSI pré-instanciée est terminée à cause du rejet, la cible DOIT produire une réponse de commande SCSI appropriée avec Vérifier la condition comme décrit au paragraphe 11.4.3. Dans les cas où un état pour la tâche SCSI a déjà été envoyé avant le rejet, aucun état supplémentaire n'est requis. Si l'erreur est détectée alors que des données de l'initiateur sont encore attendues (c'est-à-dire, si la PDU de commande ne contenait pas toutes les données et si la cible n'a pas reçu une PDU Data-Out avec le bit Final réglé à 1 pour les données non sollicitées, si il en est, et toutes les R2T en instance, si il en est) la cible DOIT attendre jusqu'à ce qu'elle reçoive les dernières PDU Data-Out attendues avec le bit F réglé à 1 avant d'envoyer la PDU de réponse.

Pour le reste de la sémantique d'usage de la PDU Rejet, voir le paragraphe 7.3.

11.17.2 DataSN/R2TSN

Ce champ n'est valide que si la PDU rejetée est un SNACK Data/R2T et si le code de cause de rejet est "Erreur de protocole" (voir le paragraphe 11.16). Le DataSN/R2TSN est le prochain numéro de séquence Data/R2T que la cible aurait envoyé pour la tâche, si il en est.

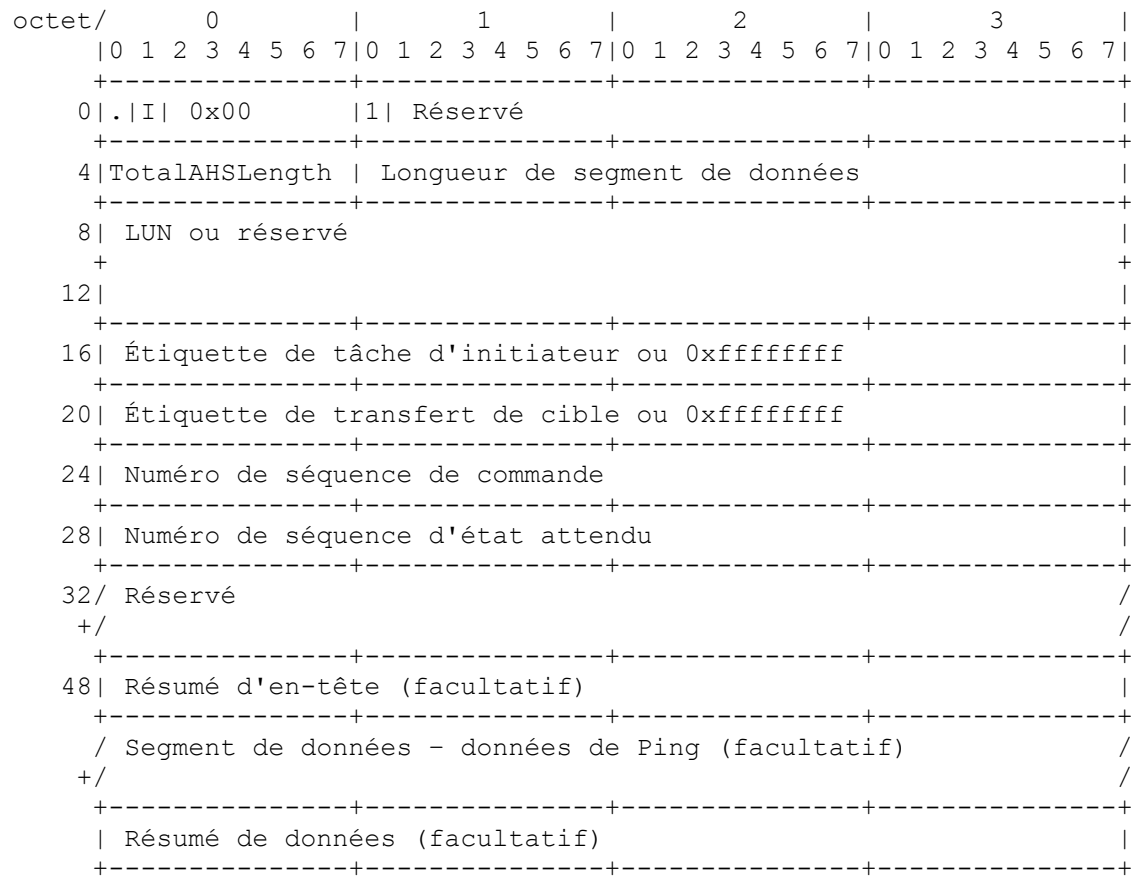
11.17.3 StatSN, ExpComSN, et MaxComSN

Ces champs portent leurs valeurs usuelles et ne sont pas en rapport avec la commande rejetée. Le numéro de séquence d'état est augmenté après un Rejet.

11.17.4 En-tête complet de mauvaise PDU

La cible retourne l'en-tête (non inclus le résumé) de la PDU erronée comme données de la réponse.

11.18 NOP-Out



NOP-Out peut être utilisé par un initiateur comme une "demande ping" pour vérifier qu'une connexion/session est toujours active et que tous ses composants sont opérationnels. La réponse NOP-In est "l'écho de ping".

Un NOP-Out est aussi envoyé par un initiateur en réponse à un NOP-In.

Un NOP-Out peut aussi être utilisé pour confirmer un changement de ExpStatSN si une autre PDU n'est pas disponible pendant longtemps.

À réception d'un NOP-In avec l'étiquette de transfert de cible réglé à une valeur valide (pas la valeur réservée 0xffffffff), l'initiateur DOIT répondre par un NOP-Out. Dans ce cas, l'étiquette de transfert de cible NOP-Out DOIT contenir une copie de l'étiquette de transfert de cible NOP-In. L'initiateur NE DEVRAIT PAS envoyer un NOP-Out en réponse à tout autre NOP-In reçu, afin d'éviter de longues séquences de PDU NOP-In et NOP-Out envoyées en réponse l'une à l'autre.

11.18.1 Étiquette de tâche d'initiateur

Le NOP-Out DOIT avoir l'étiquette de tâche d'initiateur réglée à une valeur valide seulement si une réponse sous la forme d'un NOP-In est demandée (c'est-à-dire, le NOP-Out est utilisé comme une demande de ping). Autrement, l'étiquette de tâche d'initiateur DOIT être réglée à 0xffffffff.

Lorsque une cible reçoit le NOP-Out avec une étiquette de tâche d'initiateur valide, il DOIT répondre par une réponse NOP-In (voir le paragraphe 4.6.3.6).

Si l'étiquette de tâche d'initiateur contient 0xffffffff, le bit I DOIT être réglé à 1, et le numéro de séquence de commande n'est pas augmenté après l'envoi de cette PDU.

11.18.2 Étiquette de transfert de cible

L'étiquette de transfert de cible est un identifiant alloué par la cible pour l'opération.

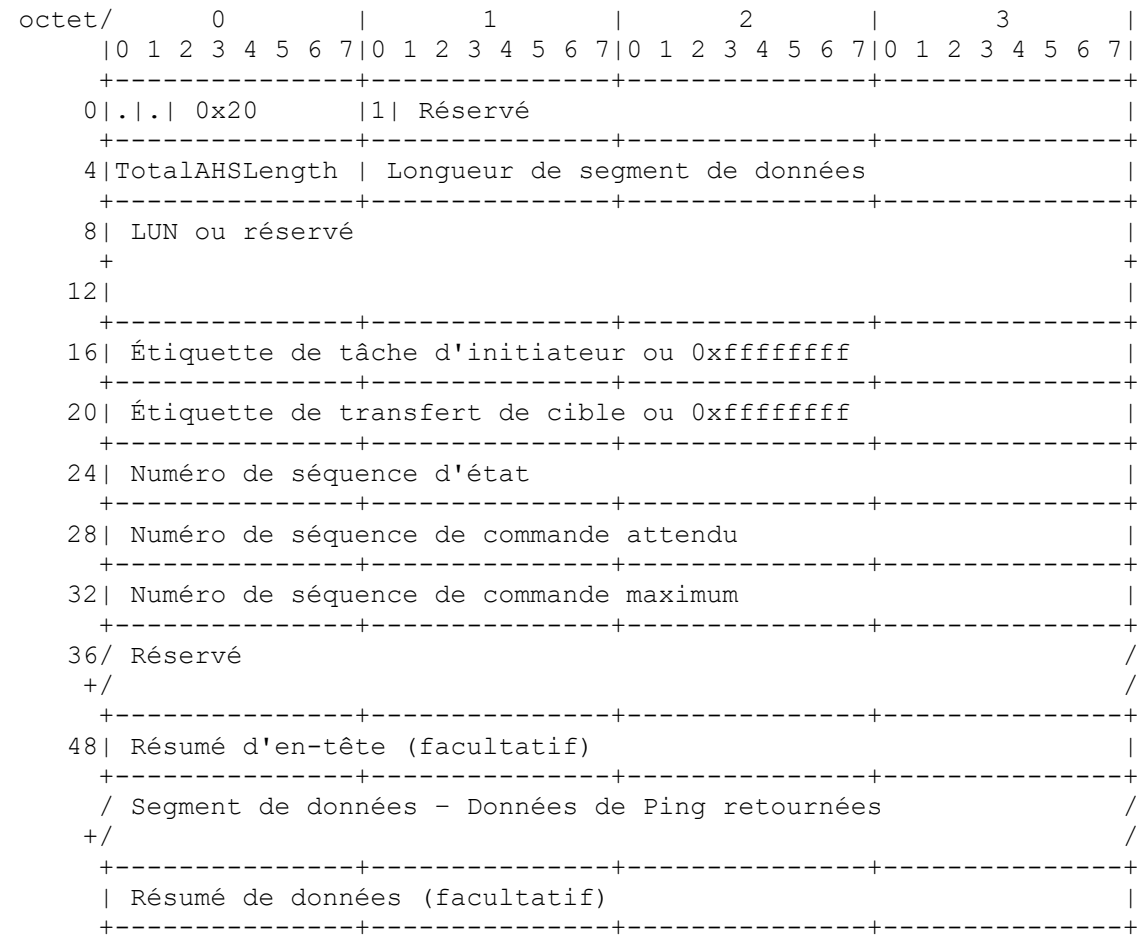
Le NOP-Out NE DOIT avoir l'étiquette de transfert de cible établi QUE si il est produit en réponse à un NOP-In avec une étiquette de transfert de cible valide. Dans ce cas, elle copie l'étiquette de transfert de cible de la PDU NOP-In. Autrement, l'étiquette de transfert de cible DOIT être réglée à 0xffffffff.

Lorsque l'étiquette de transfert de cible est réglée à une valeur autre que 0xffffffff, le champ LUN DOIT aussi être copié du NOP-In.

11.18.3 Données de Ping

Les données de Ping sont reflétées dans la réponse NOP-In. La longueur des données reflétées est limitée à la longueur maximale du segment de données reçues (MaxRecvDataSegmentLength). La longueur des données de ping est indiquée par la longueur du segment de données. 0 est une valeur valide pour Longueur de segment de données et indique l'absence de données de ping.

11.19 NOP-In



NOP-In est envoyé par une cible soit comme réponse à un NOP-Out, un "ping" à un initiateur, soit comme moyen de porter un numéro de séquence de commande attendu changé et/ou un numéro de séquence de commande maximum si une autre PDU n'est pas disponible pendant longtemps (comme déterminé par la cible).

Lorsque une cible reçoit le NOP-Out avec une étiquette de tâche d'initiateur valide (pas la valeur réservée 0xffffffff) elle DOIT répondre avec un NOP-In avec la même étiquette de tâche d'initiateur que fournie dans la demande NOP-Out. Elle DOIT aussi dupliquer jusqu'aux MaxRecvDataSegmentLength premiers octets de données de Ping fournies par l'initiateur.

Pour une telle réponse, l'étiquette de transfert de cible DOIT être 0xffffffff. La cible NE DEVRAIT PAS envoyer un NOP-In en réponse à tout autre NOP-Out reçu afin d'éviter de longues séquences de PDU NOP-In et NOP-Out envoyées en réponse l'une à l'autre.

Autrement, quand une cible envoie un NOP-In qui n'est pas une réponse à un NOP-Out reçu de l'initiateur, l'étiquette de tâche d'initiateur DOIT être réglée à 0xffffffff, et le segment de données NE DOIT PAS contenir de données (Longueur de segment de données DOIT être 0).

11.19.1 Étiquette de transfert de cible

Si la cible répond à un NOP-Out, ce champ est réglé à la valeur réservée 0xffffffff.

Si la cible envoie un NOP-In comme ping (avec l'intention de recevoir un NOP-Out correspondant) ce champ est réglé à une valeur valide (pas la valeur réservée 0xffffffff).

Si la cible initie un NOP-In sans vouloir recevoir un NOP-Out correspondant, ce champ DOIT contenir la valeur réservée 0xffffffff.

11.19.2 Numéro de séquence d'état

Le champ Numéro de séquence d'état va toujours contenir le prochain numéro de séquence d'état. Cependant, quand l'étiquette de tâche d'initiateur est réglée à 0xffffffff, le numéro de séquence d'état pour la connexion n'est pas augmenté après l'envoi de cette PDU.

11.19.3 LUN

Un LUN DOIT être réglé à une valeur correcte quand l'étiquette de transfert de cible est valide (pas la valeur réservée 0xffffffff).

12. Clés textuelles de sécurité et méthodes d'authentification iSCSI

Seules les clés suivantes sont utilisées durant l'étape de négociation de sécurité de la phase d'établissement :
SessionType *(type de session)*
InitiatorName *(nom de l'initiateur)*
TargetName *(nom de la cible)*
TargetAddress *(adresse de la cible)*
InitiatorAlias *(alias de l'initiateur)*
TargetAlias *(alias de la cible)*
TargetPortalGroupTag *(étiquette de groupe de portail cible)*
AuthMethod *(méthode d'authentification)* et les clés utilisées par les méthodes d'authentification spécifiée au paragraphe 12.1, ainsi que toutes leurs clés associées, ainsi que les méthodes d'authentification spécifiques de fabricant.
D'autres clés NE DOIVENT PAS être utilisées.

SessionType, InitiatorName, TargetName, InitiatorAlias, TargetAlias, et TargetPortalGroupTag sont décrites à la Section 13 car elles peuvent être utilisées aussi dans l'étape de négociation opérationnelle.

Toutes les clés de sécurité ont une applicabilité sur l'ensemble de la connexion.

12.1 AuthMethod

Utilisation : Durant l'établissement – négociation de la sécurité
Envoyeurs : Initiateur et cible
Portée : connexion

AuthMethod = <liste-de-valeurs>

Le principal élément de la négociation de la sécurité est la méthode d'authentification (AuthMethod).

Les méthodes d'authentification qui peuvent être utilisées (apparaître dans la liste-des-valeurs) sont soit des méthodes spécifiques du fabricant, soit celles qui figurent dans le tableau suivant :

Nom	**Description**
KRB5	Kerberos V5 - défini dans la [RFC4120]
SRP	Secure Remote Password - défini dans la [RFC2945]
CHAP	Challenge Handshake Authentication Protocol - défini dans la [RFC1994]
None	Pas d'authentification

Le choix de AuthMethod est suivi par un "échange d'authentification" spécifique de la méthode d'authentification choisie.

La proposition de méthode d'authentification peut être faite par l'initiateur ou par la cible. Cependant, l'initiateur DOIT faire le premier pas spécifique de la méthode d'authentification choisie aussitôt qu'elle est choisie. Il s'ensuit que si la cible fait la proposition de méthode d'authentification, l'initiateur envoie la ou les premières clés de l'échange avec son choix de méthode d'authentification.

L'échange d'authentification authentifie l'initiateur à la cible et, facultativement, la cible à l'initiateur. L'authentification est d'utilisation FACULTATIVE mais DOIT être prise en charge par la cible et l'initiateur.

Initiateur et cible DOIVENT mettre en œuvre CHAP. Toutes les autres méthodes d'authentification sont FACULTATIVES.

Des algorithmes d'extension privés ou publics PEUVENT aussi être négociés pour les méthodes d'authentification. Chaque fois qu'un algorithme d'extension privé ou public fait partie de l'offre par défaut (l'offre faite en l'absence d'une action administrative explicite) la mise en œuvre DOIT s'assurer que CHAP figure sur la liste comme solution de remplacement dans l'offre par défaut et que "None" ne fait pas partie de l'offre par défaut.

Les méthodes d'authentification d'extension DOIVENT être nommées en utilisant un des deux formats suivants :
1) Z-nom_dns.inversé.de.fabricant.faire_quelque chose=
2) Nouvelle clé publique sans contrainte de préfixe de nom

Les méthodes d'authentification nommées en utilisant le format Z- sont utilisées comme extensions privées. Les nouvelles clés publiques doivent être enregistrées auprès de l'IANA en utilisant le processus de revue par l'IETF ([RFC5226]). Les nouvelles extensions publiques pour les méthodes d'authentification NE DOIVENT PAS utiliser le préfixe de nom Z#.

Pour toutes les méthodes d'authentification d'extension publiques ou privées, les clés spécifiques de méthode DOIVENT se conformer au format spécifié au paragraphe 6.1 pour étiquette-standard.

Pour identifier le fabricant pour les méthodes d'authentification d'extension privées, on suggère d'utiliser le nom DNS inversé comme préfixe aux noms de résumé appropriés.

La partie de nom de résumé qui suit Z- DOIT se conformer au format de étiquette-standard spécifié au paragraphe 6.1.

La prise en charge des méthodes d'authentification d'extension publiques ou privées est FACULTATIVE.

Les paragraphes qui suivent définissent les échanges spécifiques pour chacune des méthodes d'authentification normalisées. Comme mentionné plus haut, la première étape est toujours faite par l'initiateur.

12.1.1 Kerberos

Pour KRB5 (Kerberos V5) [RFC4120], [RFC1964], l'initiateur DOIT utiliser :

KRB_AP_REQ=<KRB_AP_REQ>

où KRB_AP_REQ est le message client comme défini dans la [RFC4120].

Le nom de principal par défaut supposé par un initiateur ou cible iSCSI (avant toute action de configuration administrative) DOIT être, respectivement, le nom d'initiateur iSCSI ou le nom de cible iSCSI, préfixé de la chaîne "iscsi/".

Si l'authentification de l'initiateur échoue, la cible DOIT répondre avec un rejet d'établissement avec l'état "Échec d'authentifiion". Autrement, si l'initiateur a choisi l'option d'authentification mutuelle (en réglant à MUTUEL-EXIGÉ dans le champ ap-options de la KRB_AP_REQ) ; la cible DOIT répondre avec :

KRB_AP_REP=<KRB_AP_REP>

où KRB_AP_REP est le message de réponse du serveur, comme défini dans la [RFC4120].

Si l'authentification mutuelle a été choisie et si l'authentification de la cible échoue, l'initiateur DOIT clore la connexion.

KRB_AP_REQ et KRB_AP_REP sont des valeurs binaires, et leur longueur binaire (pas la longueur de la chaîne de caractères qui les représente en forme codée) NE DOIT PAS excéder 65536 octets. Les codages Hex ou Base64 peuvent être utilisés pour KRB_AP_REQ et KRB_AP_REP ; voir le paragraphe 6.1.

12.1.2 Secure Remote Password (SRP)

Pour le mot de passe distant sécurisé (SRP) [RFC2945], l'initiateur DOIT utiliser :

SRP_U=<U> TargetAuth=Oui /* ou TargetAuth=Non */

La cible DOIT répondre avec un rejet d'établissement et l'état "Échec d'autorisation" ou répondre par :

SRP_GROUP=<G1,G2...> SRP_s=<s>

où G1,G2... sont les groupes proposés, en ordre de préférence.

L'initiateur DOIT soit clore la connexion, soit continuer avec :

SRP_A=<A> SRP_GROUP=<G>

où G est un des G1, G2... qui étaient proposés par la cible.

La cible DOIT répondre avec un rejet d'établissement et l'état "Éche d'authentication" ou répondre avec :

SRP_B=<B>

L'initiateur DOIT clore la connexion ou continuer avec :

SRP_M=<M>

Si l'authentification de l'initiateur échoue, la cible DOIT répondre avec un rejet d'établissement et l'état "Échec d'authentification". Autrement, si l'initiateur envoie TargetAuth=Oui dans le premier message (demandant l'authentification de la cible) la cible DOIT répondre avec :

SRP_HM=<H(A | M | K)>

Si l'authentification de la cible échoue, l'initiateur DOIT clore la connexion.

U, s, A, B, M, et H(A | M | K) sont définis dans la [RFC2945] (en utilisant la fonction de hachage SHA1, comme SRP-SHA1) et G,Gn ("Gn" pour G1,G2...) sont des identifiants des groupes SRP spécifiés dans la [RFC3723].

G, Gn, et U sont des chaînes de texte ; s, A, B, M, et H(A | M | K) sont des valeurs binaires. La longueur de s ,A, B, M et H(A | M | K) en forme binaire (non pas la longueur de la chaîne de caractères qui les représente en forme codée) NE DOIT PAS excéder 1024 octets. Les codages Hex ou Base64 peuvent être utilisés pour s, A, B, M et H(A | M | K) ; voir le paragraphe 6.1.

Voir à l'Appendice B l'exemple d'établissement qui s'y rapporte.

Pour SRP_GROUP, tous les groupes spécifiés dans la [RFC3723] jusqu'à 1536 bits (c'est-à-dire, SRP-768, SRP-1024, SRP-1280, SRP-1536) doivent être pris en charge par les initiateurs et les cibles. Pour garantir l'interopérabilité, les cibles DOIVENT toujours offrir "SRP-1536" comme un des groupes proposés.

12.1.3 Challenge Handshake Authentication Protocol (CHAP)

Pour CHAP [RFC1994], l'initiateur DOIT utiliser :

CHAP_A=<A1,A2...>

où A1,A2... sont les algorithmes proposés, en ordre de préférence.

La cible DOIT répondre avec un rejet d'établissement et l'état "Échec d'authentification" ou répondre avec :

CHAP_A=<A> CHAP_I=<I> CHAP_C=<C>

où A est un des A1,A2... qui étaient proposés par l'initiateur.

L'initiateur DOIT continuer avec :

CHAP_N=<N> CHAP_R=<R>

ou, si il demande l'authentification de la cible, avec :

CHAP_N=<N> CHAP_R=<R> CHAP_I=<I> CHAP_C=<C>

Si l'authentification de l'initiateur échoue, la cible DOIT répondre avec un rejet d'établissement et l'état "Échec d'authentification". Autrement, si l'initiateur a demandé l'authentification de la cible, la cible DOIT soit répondre par un rejet d'établissement "Échec d'authentification", soit répondre avec :

CHAP_N=<N> CHAP_R=<R>

Si l'authentification de la cible échoue, l'initiateur DOIT clore la connexion.

N, (A, A1, A2), I, C, et R sont (respectivement) le nom, l'algorithme, l'identifiant, le défi , et la réponse comme défini dans la [RFC1994].

N est une chaîne de texte ; A, A1, A2, et I sont des nombres ; C et R sont des valeurs binaires. Leur longueur binaire (pas la longueur de la chaîne de caractères qui les représente en forme codée) NE DOIT PAS excéder 1024 octets. Les codages Hex ou Base64 peuvent être utilisés pour C et R ; voir le paragraphe 6.1.

Voir à l'Appendice B l'exemple d'établissement qui s'y rapporte.

Pour l'algorithme, comme déclaré dans la [RFC1994], il est exigé qu'une valeur soit mise en œuvre :

5 (CHAP avec MD5)

Pour garantir l'interopérabilité, les initiateurs DOIVENT toujours l'offrir comme un des algorithmes proposés.

13. Clés textuelles de fonctionnement d'établissement et de texte

Certains paramètres spécifiques de session NE DOIVENT être portés QUE sur la connexion de tête et ne peuvent pas être changés après l'établissement de la connexion de tête (par exemple, MaxConnections -- le nombre maximum de connexions). Cela tient pour une session d'une seule connexion à l'égard du redémarrage de connexion. Les clés qui entrent dans cette catégorie ont le "use: LO" (Seulement de tête).

Les clés qui peuvent seulement être utilisées durant l'établissement ont le "use: IO" (Seulement à l'initialisation) tandis que celles qui peuvent être utilisées dans les deux phases Établissement et Pleines caractéristiques ont "use: ALL" *(utilisation : TOUTES)*

Les clé qui ne peuvent être utilisées que durant la phase de pleines caractéristiques utilisent “Seulement phase de pleines caractéristiques” : FFPO (Full Feature Phase Only).

Les clés marquées comme “toute étape” (Any-Stage) peuvent aussi apparaître dans l'étape de négociation de sécurité, alors que toutes les autres clés décrites dans cette section sont des clés de fonctionnement.

Les clés qui n'exigent pas de répondre sont marquées comme Déclarative.

La portée de la clé est indiquée comme de session (SW, *session-wide*) ou seulement de connexion (CO, *connexion-only*).

"Fonction résultante", chaque fois qu'elle est mentionnée, déclare la fonction qui peut s'appliquer pour vérifier la validité du choix de celui qui répond. "Minimum" signifie que la valeur choisie ne peut pas excéder la valeur offerte. "Maximum" signifie que la valeur choisie ne peut pas être inférieure à la valeur offerte. "ET" *("AND")* signifie que la valeur choisie doit être un résultat possible de la fonction Booléenne "et" avec une valeur booléenne arbitraire (par exemple, si la valeur offerte est le Non, la valeur choisie doit être Non). "OU" signifie que la valeur choisie doit être un résultat possible d'une fonction booléenne "ou" avec une valeur booléenne arbitraire (par exemple, si la valeur offerte est Oui, la valeur choisie doit être Oui).

13.1 Résumé d'en-tête et Résumé de données

Utilisation : IO
Envoyeurs : Initiateur et cible
Portée : CO
Résumé d'en-tête = <liste-de-valeurs>
Résumé de données = <liste-de-valeurs>

Par défaut, aucun aussi bien pour HeaderDigest que DataDigest.

Les résumés permettent la vérification de bout en bout de l'intégrité de données non chiffrées au delà des vérifications d'intégrité fournies par les couches de liaison et la couverture du chemin de communication entier, incluant tous les éléments qui peuvent changer les PDU de niveau réseau, comme les routeurs, les commutateurs, et les mandataires.

Le tableau suivant fait la liste des sommes de contrôle d'intégrité qui peuvent être négociées pour les résumés et DOIVENT être mises en œuvre par chaque initiateur et cible iSCSI. Ces options de résumé n'ont de signification que pour la détection d'erreur.

Nom	**Description**	**Générateur**
CRC32C	CRC de 32 bits	0x11edc6f41
None	pas de résumé	

Le polynome générateur G(x) pour ce résumé est donné en notation hexadécimale (par exemple, "0x3b" pour 0011 1011, et le polynome est x**5 + x**4 + x**3 + x + 1).

Lorsque l'initiateur et la cible s'accordent sur un résumé, celui-ci DOIT être utilisé pour chaque PDU dans la phase de pleines caractéristiques.

Les octets de bourrage, lorsque présents dans un segment couvert par un CRC, DEVRAIENT être réglés à 0 et être inclus dans le CRC.

Le CRC DOIT être calculé par une méthode qui produise le même résultat que le processus suivant :
- Les bits de PDU sont considérés comme les coefficients d'un polynome M(x) de degré n – 1 ; le bit 7 de l'octet de plus faible numéro est considéré comme le bit de poids fort (x**n - 1), suivi par le bit 6 de l'octet de plus faible numéro jusqu'au bit 0 de l'octet du plus fort numéro (x**0).
- Les 32 bits de plus fort poids sont complétés.
- Le polynome est multiplié par x**32, puis divisé par G(x). Le polynome générateur produit un reste R(x) de degré ≤ 31.
- Les coefficients de R(x) sont formés en séquence de 32 bits.
- La séquence de bits est complétée, et le résultat est le CRC.
- Les bits du CRC sont transposés en mot de résumé. Le coefficient x**31 est transposé en le bit 7 de l'octet de plus faible numéro du résumé, et la transposition continue avec les coefficients et bits successifs de sorte que le coefficient ix**24 soit transposé en bit 0 de l'octet de plus faible numéro. La transposition continue ensuite avec le coefficient x**23 transposé en bit 7 de l'octet suivant dans le résumé jusqu'à ce que le coefficient x**0 soit transposé en le bit 0 de l'octet du numéro le plus élevé du résumé.
- Le calcul du CRC sur tout segment (de données ou d'en-tête) étendu pour inclure le CRC construit en utilisant le générateur 0x11edc6f41 va toujours donner la valeur 0x1c2d19ed comme reste final (R(x)). Cette valeur est donnée ici sous sa forme polynomiale (c'est-à-dire, non transposée en mot de résumé).

Pour une discussion sur les critères de choix pour le CRC, voir la [RFC3385]. Pour une analyse détaillée du polynome iSCSI, voir [Castagnoli93].

Les algorithmes d'extension privés ou publics PEUVENT aussi être négociés pour les résumés. Chaque fois qu'un algorithme d'extension de résumé privé ou public fait partie de l'offre par défaut (l'offre faite en l'absence d'une action administrative explicite) la mise en œuvre DOIT s'assurer que CRC32C figure dans la liste comme solution de remplacement dans l'offre par défaut et que "None" ne fait pas partie de l'offre par défaut.

Les algorithmes d'extension de résumé DOIVENT être désignés en utilisant un des deux formats suivants :
1) Y-nom_dns.inversé.de.fabricant.faire_quelque chose=
2) Nouvelle clé publique sans contrainte de préfixe de nom

Les résumés désignés en utilisant le format Y- sont utilisés pour des besoins privés (non enregistrés). De nouvelles clés publiques doivent être enregistrés auprès de l'IANA en utilisant le processus de revue par l'IETF [RFC5226]. Les nouvelles extensions publiques pour des résumés NE DOIVENT PAS utiliser le préfixe de nom Y#.

Pour les résumés d'extension privée, pour identifier le fabricant, on suggère d'utiliser le nom DNS inversé comme préfixe du propre nom de résumé.

La partie du nom de résumé qui suit Y- DOIT se conformer au format pour étiquette-standard spécifié au paragraphe 6.1.

La prise en charge des résumés d'extension publique ou privée est FACULTATIVE.

13.2 MaxConnections

Utilisation : LO
Envoyeurs : Initiateur et cible
Portée : SW
Non pertinent quand Type de session = Discovery
MaxConnections = <valeur numérique de 1 à 65535>
Défaut : 1.
Fonction résultante : Minimum.
Initiateur et cible négocient le nombre maximum de connexions demandées/acceptables.

13.3 SendTargets

Utilisation : FFPO
Envoyeurs : Initiateur
Portée : SW

Voir la description complète à l'Appendice C.

13.4 TargetName

Utilisation : IO par initiateur, FFPO par cible -- seulement comme réponse à SendTargets, Declarative, ou Any-Stage
Envoyeurs : Initiateur et cible
Portée : SW
TargetName = <valeur de nom iSCSI>

Exemples :
TargetName=iqn.1993-11.com.disk-vendor:diskarrays.sn.45678
TargetName=eui.020000023B040506
TargetName=naa.62004567BA64678D0123456789ABCDEF

L'initiateur de la connexion TCP DOIT fournir cette clé au point d'extrémité distant dans la première demande d'établissement si l'initiateur n'est pas en train d'établir une session Discovery. Le nom de cible iSCSI spécifie le nom unique au monde de la cible.

La clé TargetName peut aussi être retournée par la demande Text SendTargets (qui est sa seule utilisation lorsque elle est produite par une cible).

Le TargetName NE DOIT PAS être redéclaré au sein de la phase d'établissement.

13.5 InitiatorName

Utilisation : IO, Declarative, Any-Stage
Envoyeurs : Initiateur
Portée : SW
InitiatorName = <valeur de nom iSCSI>

Exemples :
InitiatorName=iqn.1992-04.com.os-vendor.plan9:cdrom.12345
InitiatorName=iqn.2001-02.com.ssp.users:customer235.host90
InitiatorName=naa.52004567BA64678D

L'initiateur de la connexion TCP DOIT fournir cette clé au point d'extrémité distant dans le premier login de la phase d'établissement pour chaque connexion. La clé InitiatorName permet à l'initiateur de s'identifier au point d'extrémité distant.

InitiatorName NE DOIT PAS être redéclaré au sein de la phase d'établissement.

13.6 TargetAlias

Utilisation : ALL, Declarative, Any-Stage
Envoyeurs : cible
Portée : SW
TargetAlias = <valeur de nom local iSCSI>

Exemples :
TargetAlias=Disque_de_Claude
TargetAlias=Disque d'enregistrement du serveur 1 de la base de données
TargetAlias=Disque 20 du serveur 3 de la Toile

Si une cible a été configurée avec un nom ou description lisible par l'homme, ce nom DEVRAIT être communiqué à l'initiateur durant une PDU Réponse d'établissement si Type de session=Normal (voir paragraphe 13.21). Cette chaîne n'est pas utilisée comme identifiant, ni n'est destinée à être utilisée pour des décisions d'authentification ou d'autorisation. Elle peut être affichée par l'interface d'utilisateur de l'initiateur dans une liste des cibles auxquelles elle est connectée.

13.7 InitiatorAlias

Utilisation : ALL, Declarative, Any-Stage
Envoyeurs : Initiateur
Portée : SW
InitiatorAlias=<valeur de nom local iSCSI>

Exemples :
InitiatorAlias=Serveur Web 4
InitiatorAlias=spyalley.nsa.gov
InitiatorAlias=Serveur d'échange

Si un initiateur a été configuré avec un nom ou description lisible par l'homme, il DEVRAIT être communiqué à la cible durant une PDU Demande d'établissement. Sinon, le nom d'hôte peut être utilisé à la place. Cette chaîne n'est pas utilisée comme identifiant, ni n'est destinée à être utilisée pour des décisions d'authentification ou d'autorisation. Elle peut être affichée par l'interface d'utilisateur de la cible dans une liste des initiateurs auxquels elle est connectée.

13.8 TargetAddress

Utilisation : ALL, Declarative, Any-Stage
Envoyeurs : Cible
Portée : SW
TargetAddress=nom de domaine[:accès][,étiquette de groupe portail]

Le nom de domaine peut être spécifié comme nom d'hôte DNS, adresse IPv4 en décimal séparé par des points, ou une adresse IPv6 entre crochets comme spécifié dans la [RFC3986].

Si l'accès TCP n'est pas spécifié, on suppose que c'est l'accès alloué par défaut par l'IANA pour iSCSI (Section 14).
Si l'adresse cible est retournée par suite d'un état “redirect” dans une réponse Login, la virgule et l'étiquette de groupe portail DOIVENT être omises.

Si l'adresse de cible est retournée au sein d'une réponse SendTargets, l'étiquette de groupe portail DOIT être incluse.

Exemples :
TargetAddress=10.0.0.1:5003,1
TargetAddress=[1080:0:0:0:8:800:200C:417A],65
TargetAddress=[1080::8:800:200C:417A]:5003,1
TargetAddress=centredecalcul.exemple.com,23

L'utilisation de l'étiquette de groupe portail est décrite dans l'Appendice C. Les formats pour l'accès et l'étiquette de groupe portail sont les mêmes que ceux spécifiés dans TargetPortalGroupTag.

13.9 TargetPortalGroupTag

Utilisation : IO par cible, Declarative, Any-Stage
Envoyeurs : Cible
Portée : SW
TargetPortalGroupTag=<valeur binaire de 16 bits>

Exemple : TargetPortalGroupTag=1

La clé TargetPortalGroupTag est une valeur binaire de 16 bits qui identifie de façon univoque un groupe portail au sein d'un nœud iSCSI cible. Cette clé porte la valeur de l'étiquette du groupe portail qui dessert la demande d'établissement. La cible iSCSI retourne cette clé à l'initiateur dans la PDU Réponse d'établissement à la première PDU Demande d'établissement qui a le bit C réglé à 0 quand TargetName est donné par l'initiateur.

[SAM2] note dans un texte d'information que la valeur de TPGT ne devrait pas être zéro ; cette note est incorrecte. Une valeur de zéro est permise comme valeur légale pour le TPGT. Cette discordance est actuellement corrigée dans [SAM4].

Pour l'explication complète de l'usage de cette clé, voir le paragraphe 6.3.

13.10 InitialR2T

Utilisation : LO
Envoyeurs : Initiateur et cible
Portée : SW
Non pertinent quand SessionType=Discovery
R2T initial = <valeur booléenne>

Exemples :
I->InitialR2T=No
T->InitialR2T=No

La valeur par défaut est Oui.
Fonction résultante : OU.

La clé InitialR2T est utilisée pour désactiver l'utilisation par défaut de R2T pour les opérations unidirectionnelles et la partie sortante des commandes bidirectionnelles, permettant donc à un initiateur de commencer à envoyer des données à une cible comme si il avait reçu un R2T initial avec Décalage de mémoire tampon = Longueur de données immédiates et Longueur de transfert de données désirée = (minimum de (Longueur de première salve, Longueur attendue de transfert de données) – Longueur de données immédiates reçues).

L'action par défaut est que le R2T est exigé, sauf si l'initiateur et la cible envoient tous deux cet attribut de paire de clés en spécifiant InitialR2T=Non. Seule la première salve de données sortantes (PDU Données immédiates et/ou séparées) peut être envoyée non sollicitée (c'est-à-dire, sans exiger un R2T explicite).

13.11 ImmediateData

Utilisation : LO
Envoyeurs : Initiateur et cible
Portée : SW
Non pertinent quand SessionType=Discovery
ImmediateData=<valeur booléenne>
Par défaut : Oui.
Fonction résultante : ET.

L'initiateur et la cible négocient la prise en charge des données immédiates. Pour désactiver les données immédiates, l'initiateur ou la cible doit déclarer son désir de le faire. ImmediateData peut être activé si l'initiateur et la cible ont tous deux ImmediateData=Oui.

Si ImmediateData est réglé à Oui et si InitialR2T est réglé à Oui (par défaut) seules les données immédiates sont alors acceptées dans la première salve.

Si ImmediateData est réglé à Non et InitialR2T réglé à Oui, l'initiateur NE DOIT alors PAS envoyer de données non sollicitées et la cible DOIT rejeter les données non sollicitées avec le code de réponse correspondant.

Si ImmediateData est réglé à Non et InitialR2T à Non, l'initiateur NE DOIT alors PAS envoyer de données immédiates non sollicitées mais PEUT envoyer une salve non sollicitée de PDU Data-OUT.

Si ImmediateData est réglé à Oui et InitialR2T à Non, l'initiateur PEUT alors envoyer des données immédiates non sollicitées et/ou une salve non sollicitée de PDU Data-OUT.

Le tableau suivant résume les options de données non sollicitées :

InitialR2T	ImmediateData	PDU Data-Out non sollicitées	ImmediateData
Non	Non	Oui	Non
Non	Oui	Oui	Oui
Oui	Non	Non	Non
Oui	Oui	Non	Oui

13.12 MaxRecvDataSegmentLength

Utilisation : ALL, Declarative
Envoyeurs : Initiateur et cible
Portée : CO
MaxRecvDataSegmentLength=<valeur numérique de 512 à (2**24 - 1)>
Par défaut : 8192 octets.

L'initiateur ou la cible déclare le segment de données maximum en octets qu'il peut recevoir dans une PDU iSCSI.

L'émetteur (initiateur ou cible) est obligé d'envoyer des PDU avec un segment de données qui n'excède pas le MaxRecvDataSegmentLength du receveur.

Un receveur cible est de plus limité par MaxBurstLength pour les données sollicitées et par FirstBurstLength pour les données non sollicitées. Un initiateur NE DOIT PAS envoyer de PDU sollicitées excédant MaxBurstLength ni de PDU non sollicitées excédant FirstBurstLength (ou FirstBurstLength-ImmediateDataLength si des données immédiates ont été envoyées).

13.13 MaxBurstLength

Utilisation : LO
Envoyeurs : Initiateur et cible
Portée : SW
Non pertinent quand SessionType=Discovery
MaxBurstLength=<valeur numérique de 512 à (2**24 - 1)>
Par défaut : 262144 (256 koctets).
Fonction résultante : Minimum.

L'initiateur et la cible négocient la charge utile maximum de données SCSI en octets dans une séquence iSCSI Data-In ou Data-Out sollicitée. Une séquence consiste en une ou plusieurs PDU Data-In ou Data-Out consécutives, qui se termine par une PDU Data-In ou Data-Out avec le bit F réglé à 1.

13.14 FirstBurstLength

Utilisation : LO
Envoyeurs : Initiateur et cible
Portée : SW
Non pertinent quand SessionType=Discovery
Non pertinent quand (InitialR2T=Oui et ImmediateData=Non)
FirstBurstLength=<valeur numérique de 512 à (2**24 - 1)>
Par défaut : 65536 (64 koctets).
Fonction résultante : Minimum.

Initiateur et cible négocient la quantité maximum en octets de données non sollicitées qu'un initiateur iSCSI peut envoyer à la cible durant l'exécution d'une seule commande SCSI. Cela couvre les données immédiates (s'il en est) et la séquence de PDU Données non sollicitées Data-Out PDU (s'il en est) qui suivent la commande.

FirstBurstLength NE DOIT PAS excéder MaxBurstLength.

13.15 DefaultTime2Wait

Utilisation : LO
Envoyeurs : Initiateur et cible
Portée : SW
DefaultTime2Wait=<valeur numérique de 0 à 3600>
Par défaut : 2.
Fonction résultante : Maximum.

L'initiateur et la cible négocient la durée minimum, en secondes, d'attente avant de tenter un désétablissement explicite/implicite ou une réallocation de tâche active après une terminaison ou une réinitialisation inattendue de connexion.

Une valeur de 0 indique que le désétablissement ou la réallocation de tâche active ne peut être tenté immédiatement.

13.16 DefaultTime2Retain

Utilisation : LO
Envoyeurs : Initiateur et cible
Portée : SW
DefaultTime2Retain=<valeur numérique de 0 à 3600>
Par défaut : 20.
Fonction résultante : Minimum.

L'initiateur et la cible négocient la durée maximum, en secondes, après une attente initiale (Time2Wait) avant laquelle une réallocation de tâche active est encore possible après une terminaison ou une réinitialisation de connexion inattendue .

Cette valeur est aussi la temporisation d'état de session si la connexion en question est la dernière connexion LOGGED_IN dans la session.

Une valeur de 0 indique que l'état de connexion/tâche est immédiatement éliminé par la cible.

13.17 MaxOutstandingR2T

Utilisation : LO
Envoyeurs : Initiateur et cible
Portée : SW
MaxOutstandingR2T=<valeur numérique de 1 à 65535>
Non pertinent quand SessionType=Discovery
Par défaut : 1.
Fonction résultante : Minimum.

L'initiateur et la cible négocient le nombre maximum de R2T en instance par tâche, à l'exclusion de tout R2T implicite initial qui pourrait faire partie de cette tâche. Un R2T est considéré comme en instance jusqu'à ce que la dernière PDU de données (avec le bit F réglé à 1) soit transférée, ou qu'une fin de temporisation de réception de séquence (paragraphe 7.1.4.1) soit rencontrée pour cette séquence de données.

13.18 DataPDUInOrder

Utilisation : LO
Envoyeurs : Initiateur et cible
Portée : SW
Non pertinent quand SessionType=Discovery
DataPDUInOrder=<valeur booléenne>

Par défaut : Oui.
Fonction résultante : OU.

"Non" est utilisé par iSCSI pour indiquer que les PDU de données dans les séquences peuvent être dans n'importe quel ordre. "Oui" est utilisé pour indiquer que les PDU de données dans les séquences doivent être à des adresses de croissance continue et que les chevauchements sont interdits.

13.19 DataSequenceInOrder

Utilisation : LO
Envoyeurs : Initiateur et cible
Portée : SW
Non pertinent quand SessionType=Discovery
DataSequenceInOrder=<valeur booléenne>
Par défaut : Oui.
Fonction résultante : OU.

Une séquence de données est une séquence de PDU Data-In ou Data-Out qui se termine par une PDU Data-In ou Data-Out avec le bit F réglé à 1. Une séquence Data-Out est envoyée soit non sollicitée, soit en réponse à un R2T. Les séquences couvrent une gamme de décalages.

Si DataSequenceInOrder est réglé à Non, des séquences de PDU de données peuvent être transférées dans n'importe quel ordre.

Si DataSequenceInOrder est réglé à Oui, les séquences de données DOIVENT être transférées en utilisant des décalages de séquence continus non décroissants (décalage de mémoire tampon de R2T pour les écritures, ou plus petit décalage de mémoire tampon de Data-In SCSI au sein d'une séquence de données de lecture).

Si DataSequenceInOrder est réglé à Oui, une cible peut réessayer au plus le dernier R2T, et un initiateur peut au plus demander la retransmission de la dernière séquence de données de lecture. Pour cette raison, si le niveau de récupération d'erreur n'est pas 0 et si DataSequenceInOrder est réglé à Oui, alors MaxOutstandingR2T DOIT être réglé à 1.

13.20 ErrorRecoveryLevel

Utilisation : LO
Envoyeurs : Initiateur et cible
Portée : SW
Niveau de récupération d'erreur=<valeur numérique de 0 à 2>
Par défaut : 0.
Fonction résultante : Minimum.

L'initiateur et la cible négocient le niveau de récupération à prendre en charge.

Les niveaux de récupération représentent une combinaison de capacités de récupération. Chaque niveau de récupération inclut toutes les capacités des niveaux de récupération inférieurs et leur en ajoute quelques nouvelles.

Dans la description des mécanismes de récupération, certaines classes de récupération sont spécifiées. Le paragraphe 7.1.5 décrit les transpositions entre les classes et les niveaux.

13.21 SessionType

Utilisation : LO, Declarative, Any-Stage
Envoyeurs : Initiateur
Portée : SW
SessionType=<Discovery|Normal>
Par défaut : Normal.

L'initiateur indique le type de session qu'il veut créer. La cible peut l'accepter ou le rejeter.

Une session Discovery indique à la cible que le seul objet de cette session est la découverte. Les seules demandes qu'une cible accepte dans ce type de session sont une demande Text avec une clé SendTargets et une demande de désétablissement avec comme raison "clore la session".

La session Discovery implique que MaxConnections = 1 et outrepasse les réglages par défaut et explicites. Comme le déclare le paragraphe 7.4.1, le niveau de récupération d'erreur DOIT être 0 (zéro) pour les sessions de découverte.

Selon le type de session, une cible peut décider des ressources à allouer, de la sécurité à appliquer, etc., pour la session. Si la clé SessionType va donc être offerte comme "Discovery", elle DEVRAIT être offerte dans la demande d'établissement initiale par l'initiateur.

13.22 Format de clé d'extension Private

Utilisation : ALL
Envoyeurs : Initiateur et cible
Portée : selon la clé spécifique
X-nom.dns.inversé.de.fabricants. faire_quelque_chose=

Les clés dans ce format sont utilisées pour des extensions privées. Ces clés commencent toujours par X- si elles ne sont pas enregistrées auprès de l'IANA (private). Les nouvelles clés publiques (si elles sont enregistrées auprès de l'IANA via une revue de l'IETF [RFC5226]) ne sont plus obligées d'avoir un préfixe de nom X# ; les développeurs peuvent proposer tout nom intuitif univoque.

Pour les clés non enregistrées, pour identifier le fabricant, on suggère d'utiliser le nom DNS inversé comme préfixe à la clé appropriée.

La partie du nom clé qui suit le X- DOIT se conformer au format pour nom-clé spécifié au paragraphe 6.1.

Les clés spécifiques de fabricant DOIVENT SEULEMENT être utilisées dans les sessions Normal.

La prise en charge des clés d'extension publiques ou privées est FACULTATIVE.

13.23 TaskReporting

Utilisation : LO
Envoyeurs : Initiateur et cible
Portée : SW
Non pertinent quand SessionType=Discovery
TaskReporting=<liste-de-valeurs>
Par défaut : RFC3720.

Cette clé est utilisée pour négocier la sémantique de rapport d'achèvement de tâche par la cible SCSI. Le tableau suivant décrit la sémantique qu'une cible iSCSI DOIT prendre en charge pour les valeurs de clé négociées respectives. Chaque fois que cette clé est négociée, au moins les valeurs RFC3720 et ResponseFence DOIVENT être offertes comme options par l'origine de la négociation.

Nom	Description
RFC3720	Sémantique conforme à la RFC 3720. Une réponse close n'est pas garantie, et l'achèvement rapide d'interruption de multi tâches n'est pas prise en charge.
ResponseFence	La sémantique de réponse close (paragraphe 4.2.2.3.3) DOIT être prise en charge dans les rapports d'achèvement de tâche.
FastAbort	La sémantique d'interruption rapide de multi tâches mise à jour définie au paragraphe 4.2.3.4 DOIT être prise en charge. La prise en charge de la réponse close est implicite -- c'est-à-dire, la sémantique décrite au paragraphe 4.2.2.3.3 DOIT être aussi prise en charge.

Lorsque TaskReporting n'est pas négocié à FastAbort, la sémantique standard d'interruption multi tâches du paragraphe 4.2.3.3 DOIT être utilisée.

13.24 Négociation de iSCSIProtocolLevel

Le niveau de protocole iSCSI associé au présent document est "1". Que ce soit comme répondant ou comme origine de la négociation de cette clé, une mise en œuvre iSCSI conforme à ce seul document, sans autre extension future de ce protocole, DOIT utiliser cette valeur comme défini par la [RFC7144].

13.25 Clés rendues obsolètes

Le présent document rend obsolètes les clés suivantes définies dans la [RFC3720] : IFMarker, OFMarker, OFMarkInt, et IFMarkInt. Cependant, les mises en œuvre iSCSI conformes au présent document peuvent encore recevoir ces clés obsolètes -- c'est-à-dire, dans un rôle de répondant –dans une négociation de texte.

Lorsque une clé IFMarker ou OFMarker est reçue, une mise en œuvre iSCSI conforme DEVRAIT répondre par la valeur constante "Rejet". La mise en œuvre PEUT autrement répondre avec une valeur "Non".

Cependant, la mise en œuvre NE DOIT PAS répondre avec une valeur "NonCompris" pour l'une ou l'autre de ces clés.

Lorsque une clé IFMarkInt ou OFMarkInt est reçue, une mise en œuvre iSCSI conforme DOIT répondre avec la valeur constante "Rejet". La mise en œuvre NE DOIT PAS répondre avec une valeur "NonCompris" pour l'une ou l'autre de ces clés.

13.26 X#NodeArchitecture

13.26.1 Définition

Utilisation : LO, Declarative
Envoyeurs : Initiateur et cible
Portée : SW
X#NodeArchitecture=<liste de valeurs>
Par défaut : aucun.

Exemples :
X#NodeArchitecture=ExempleOS/v1234,ExempleInc_SW_Initiateur/1.05a
X#NodeArchitecture=ExempleInc_HW_Initiateur/4010,Firmware/2.0.0.5
X#NodeArchitecture=ExempleInc_SW_Initiateur/2.1,CPU_Arch/i686

Le présent document ne définit pas la structure ni le contenu de la liste des valeurs.

L'initiateur ou la cible déclare les détails de son architecture de nœud iSCSI au point d'extrémité distant . Ces détails peuvent inclure, sans s'y limiter, les versions de logiciel ou matériel de fabricant iSCSI, la version du système d'exploitation, ou l'architecture du matériel. Cette clé peut être déclarée sur une session Discovery ou une session Normal.

La longueur de la valeur de clé (longueur totale de la liste des valeurs) NE DOIT PAS être supérieure à 255 octets.

X#NodeArchitecture NE DOIT PAS être redéclaré durant la phase d'établissement.

13.26.2 Exigences de mise en œuvre

Le comportement fonctionnel du nœud iSCSI (cela inclut la logique du protocole iSCSI – les protocoles SCSI, iSCSI, et TCP/IP) NE DOIT PAS dépendre de la présence, absence, ou contenu de la clé X#NodeArchitecture. La clé NE DOIT PAS être utilisée par des nœuds iSCSI pour l'interopérabilité ou pour l'exclusion d'autres nœuds. Pour assurer une utilisation appropriée, les valeurs de clé DEVRAIENT être réglées par le nœud lui-même, et il NE DEVRAIT PAS y avoir de dispositions pour que les valeurs de clé contiennent du texte défini par l'utilisateur.

Les nœuds qui mettent en œuvre cette clé DOIVENT choisir une des options de mise en œuvre suivantes :
- seulement transmettre la clé,
- seulement enregistrer les valeurs de clé reçues d'autres nœuds, ou
- transmettre et enregistrer les valeurs de clé.

Chaque nœud qui choisit de mettre en œuvre la transmission des valeurs de clé DOIT être prêt à traiter la réponse des nœuds iSCSI qui ne comprennent pas la clé.

Les nœuds qui mettent en œuvre la transmission et/ou l'enregistrement des valeurs de clé peuvent aussi mettre en œuvre des mécanismes administratifs qui désactivent et/ou changent les détails de l'enregistrement et de transmission de clés (voir le paragraphe 9.4). Donc, un comportement valide pour cette clé peut être qu'un nœud est complètement silencieux (le nœud ne transmet aucune valeur de clé et élimine simplement toutes les valeurs de clé qu'il reçoit sans produire une réponse NonCompris).

14. Raisons de la révision des considérations relatives à l'IANA

Le présent document a fait des changement assez significatifs dans ce domaine, et cette section souligne les raisons de ces changements. Comme spécifié prédédemment dans la [RFC3720], iSCSI avait utilisé des préfixes de chaînes de texte, comme X- et X#, pour distinguer les extensions de clés d'établissement/texte, les algorithmes de résumé, et les méthodes d'authentification de leur contrepartie normalisée. Sur la base de l'expérience d'autres protocoles, la [RFC6648] recommande cependant fortement de ne pas suivre cette pratique, en grande partie parce que les extensions qui utilisent de tels préfixes peuvent être normalisées au fil du temps, et à ce moment il peut devenir impossible de changer leur nom de chaîne de texte à cause de l'usage largement répandu du nom de la chaîne de texte existante.

L'expérience de iSCSI des extensions publiques prend en charge les recommandations de la [RFC6648], car le seul élément d'extension jamais enregistré auprès de l'IANA, la clé X#NodeArchitecture, a été spécifiée comme clé standard dans la [RFC4850] sur la voie de la normalisation et n'exige pas le préfixe X#. De plus, cette clé est la seule extension iSCSI publiqe qui ait été enregistrée auprès de l'IANA depuis que la RFC 3720 a été publiée, de sorte qu'il n'y a pas eu d'utilisation effective des formats d'extension publics X#, Y#, et Z#.

Donc, le présent document fait les changements suivants aux procédures d'enregistrement IANA pour iSCSI :

1) Les registres séparés pour les extensions publiques X#, Y#, et Z# sont supprimés. La seule entrée dans le registre pour les clés X# d'établissement/texte (X#NodeArchitecture) est transférée dans le registre pricnipal "iSCSI Login/Text Keys". L'IANA n'a jamais créé les deux derniers registres parce que il n'y a pas eu de demande d'enregistrement pour eux. Ces formats d'extension publiques (X#, Y#, Z#) NE DOIVENT PAS être utilisés, à l'exception de la clé existante X#NodeArchitecture.
2) Les procédures d'enregistrement pour les principaux registres de l'IANA "iSCSI Login/Text Keys", "iSCSI digests", et "iSCSI authentification methods" sont changées en “revue par l'IETF” [RFC5226] pour de possibles futures extensions à iSCSI. Ce changement inclut une décision délibérée de supprimer la possibilité de spécifier une extension iSCSI enregistrée par l'IANA dans une RFC publiée via une soumision indépendante de l'éditeur des RFC, car le niveau de revue dans ce procès est insuffisant pour les extensions iSCSI.
3) La restriction sur les éléments enregistrés en utilisant les formats d'extension privée (X-, Y-, Z-) dans les registres pricipaux de l'IANA est supprimée. Les extensions qui utilisent ces formats PEUVENT être enregistrées sous les procédures d'enregistrement de revue de l'IETF, mais chaque format se restreint au type d'extension pour lequel il est spécifié dans la présente RFC et NE DOIT PAS être utilisé pour d'autres types. Par exemple, le format d'extension X- pour les clés d'extension d'établissement/text NE DOIT PAS être utilisé pour les algorithmes de résumé ou les méthodes d'authentification.

15. Considérations relatives à l'IANA

Le numéro d'accès TCP bien connu pour les connexions iSCSI alloué par l'IANA est 3260, et c'est l'accès iSCSI par défaut. Les mises en œuvre qui ont besoin d'un numéro d'accès de système TCP peuvent utiliser l'accès 860, l'accès alloué par l'IANA comme accès système iSCSI ; cependant, pour utiliser l'accès 860, il DOIT être explicitement spécifié – les mises en œuvre NE DOIVENT PAS utiliser par défaut l'accès 860, car 3260 est le seul accès par défaut permis.

L'IANA a remplacé les références des accès 860 et 3260, pour TCP et UDP, par les références au présent document. Voir http://www.iana.org/assignments/service-names-port-numbers .

L'IANA a mis à jour toutes les références aux RFC 3720, RFC 4850, et RFC 5048 pour référencer à la place la présente RFC dans tous les registres iSCSI qui font partie de l'ensemble de registre de "Internet Small Computer System Interface (iSCSI) Parameters". Ce changement reflète le fait que ces trois RFC sont rendues obsolètes par la présente RFC. Les références aux autres RFC qui n'ont pas été rendues obsolètes (par exemple, les RFC 3723, RFC 5046) ne devraient pas changer.

L'IANA a effectué les actions suivantes sur le registre "iSCSI Login/Text Keys" :
- Changé la procédure d'enregistrement de revue par l'IETF en norme exigée.
- Changé la référence à la RFC 5048 en référence à la présente RFC pour ce registre.
- Ajouté la clé X#NodeArchitecture du registre "iSCSI extended key", et changé sa référence en la présente RFC.
- Changé toutes les références aux RFC 3720 et RFC 5048 en référence à la présente RFC.

L'IANA a changé la procédure d'enregistrement des registres "iSCSI authentification methods" et "iSCSI digests" de revue par l'IETF en RFC exigée.

L'IANA a supprimé le registre "iSCSI extended key", car sa seule entrée a été ajoutée au registre "iSCSI Login/Text Keys".

L'IANA a marquées comme obsolètes les valeurs 4 et 5 pour respectivement SPKM1 et SPKM2,, dans le sous registre "iSCSI authentification methods" de l'ensemble de registres des "Internet Small Computer System Interface (iSCSI) Parameters".

L'IANA a ajouté le présent document au registre "iSCSI Protocol Level" avec la valeur 1, comme mentionné au paragraphe 13.24.

Toutes les autres considérations relatives à l'IANA mentionnées dans les [RFC3720] et [RFC5048] restent inchangées. Les allocations contenues dans les sous registres suivants ne sont pas répétées dans le présent document :
- méthodes d'authentification iSCSI (de la paragraphe 13 de la [RFC3720])
- résumés iSCSI (de la paragraphe 13 de la [RFC3720]).

Le présent document rend obsolètes les valeurs de clé SPKM1 et SPKM2 pour la clé textuelle AuthMethod. Par conséquent, le préfixe de clé SPKM_text DOIT être traité comme obsolète et non réutilisé.

16. Références

16.1 Références normatives

[EUI] "Guidelines for 64-bit Global Identifier (EUI-64(TM))", <http://standards.ieee.org/regauth/oui/tutorials/EUI64.html>.

[FC-FS3] INCITS Technical Committee T11, "Fibre Channel - Framing et Signaling - 3 (FC-FS-3)", ANSI INCITS 470-2011, 2011.

[OUI] "IEEE OUI and "company_id" Assignments", < http://standards.ieee.org/regauth/oui >.

[RFC1122] R. Braden, "Exigences pour les hôtes Internet – couches de communication", STD 3, octobre 1989. *(MàJ par la* RFC6633)

[RFC1964] J. Linn, "Mécanisme GSS-API de Kerberos version 5", juin 1996. (*MàJ par* RFC4121 *et* RFC6649)

[RFC1982] R. Elz, R. Bush, "Arithmétique des numéros de série", août 1996. (MàJ RFC1034, RFC1035) *(P.S.)*

[RFC1994] W. Simpson, "Protocole d'authentification par mise en cause de la prise de contact en PPP (CHAP)", août 1996.

[RFC2119] S. Bradner, "Mots clés à utiliser dans les RFC pour indiquer les niveaux d'exigence", BCP 14, mars 1997.

[RFC2404] C. Madson, R. Glenn, "Utilisation de HMAC-SHA-1-96 au sein d'ESP et d'AH", novembre 1998. *(P.S.)*

[RFC2406] S. Kent et R. Atkinson, "Encapsulation de charge utile de sécurité IP (ESP)", novembre 1998. *(Obsolète, voir* RFC4303*)*

[RFC2451] R. Pereira, R. Adams, "Algorithmes de chiffrement ESP en mode CBC", novembre 1998. *(P.S.)*

[RFC2945] T. Wu, "Système SRP d'authentification et d'échange de clés", septembre 2000. *(P.S.)*

[RFC3454] P. Hoffman et M. Blanchet, "Préparation de chaînes internationalisées ("stringprep")", décembre 2002. *(P.S.)*

[RFC3566] S. Frankel, H. Herbert, "L'algorithme AES-XCBC-MAC-96 et son utilisation avec IPsec", septembre 2003. *(P.S.)*

[RFC3629] F. Yergeau, "UTF-8, un format de transformation de la norme ISO 10646", STD 63, novembre 2003.

[RFC3686] R. Housley, "Utilisation du mode Compteur de la norme de chiffrement évolué (AES) avec l'encapsulation de la charge utile de sécurité (ESP) dans IPsec", janvier 2004. *(P.S.)*

[RFC3722] M. Bakke, "Profil de chaîne pour les noms d'interface Internet de systèmes de petits ordinateurs (iSCSI)", avril 2004. *(P.S.)*

[RFC3723] B. Aboba et autres, "Protocoles de sécurisation de mémorisation de blocs sur IP", avril 2004. *(P.S.)*

[RFC3986] T. Berners-Lee, R. Fielding et L. Masinter, "Identifiant de ressource uniforme (URI) : Syntaxe générique", STD 66, janvier 2005.

[RFC4106] J. Viega, D. McGrew, "Utilisation du mode Galois/Compteur (GCM) dans une charge utile de sécurité par encapsulation (ESP) IPsec", juin 2005. *(P.S.)*

[RFC4120] C. Neuman et autres, "Service Kerberos d'authentification de réseau (V5)", juillet 2005. (*MàJ par* RFC4537, RFC5021, RFC6649)

[RFC4171] J. Tseng et autres, "Service de noms de mémorisation sur Internet (iSNS)", septembre 2005. *(P.S.)*

[RFC4291] R. Hinden, S. Deering, "Architecture d'adressage IP version 6", février 2006. (*MàJ par* 5952 et 6052) *(D.S.)*

[RFC4301] S. Kent et K. Seo, "Architecture de sécurité pour le protocole Internet", décembre 2005. *(P.S.) (Remplace la* RFC2401)

[RFC4303] S. Kent, "Encapsulation de charge utile de sécurité dans IP (ESP)", décembre 2005. (*Remplace* RFC2406) *(P.S.)*

[RFC4304] S. Kent, "Addendum du numéro de séquence étendu (ESN) au domaine d'interprétation IPsec (DOI) pour le protocole d'associations de sécurité et de gestion de clé Internet (ISAKMP)", décembre 2005. *(P.S.)*

[RFC4543] D. McGrew, J. Viega, "Utilisation du code d'authentification de message de Galois (GMAC) dans les ESP et AH d'IPsec", mai 2006. *(P.S.)*

[RFC4648] S. Josefsson, "Codages de données Base16, Base32 et Base64", octobre 2006. (*Remplace* RFC3548) *(P.S.)*

[RFC5226] T. Narten et H. Alvestrand, "Lignes directrices pour la rédaction d'une section Considérations relatives à l'IANA dans les RFC", BCP 26, mai 2008. (*Remplace* RFC2434)

[RFC5996] C. Kaufman, P. Hoffman, Y. Nir, P. Eronen, "Protocole d'échange de clés sur Internet, version 2 (IKEv2)", septembre 2010 (*Remplace* RFC4306, RFC4718) (*MàJ par* RFC5998) *(P.S.)*

[RFC6960] S. Santesson et autres, "Protocole d'état de certificat en ligne (OCSP) pour l'infrastructure de clé publique Internet X.509", juin 2013. (*Remplace* RFC2560, RFC6277) (*MàJ* RFC5912) *(P.S.)*

[RFC7144] F. Knight, M. Chadalapaka, "Mise à jour des caractéristiques SCSI d'interface Internet de petit système d'ordinateur (iSCSI)", avril 2014. *(P.S.)*

[RFC7145] M. Ko, A. Nezhinsky, "Extensions d'interface Internet de petit système d'ordinateur (iSCSI) pour la spécification d'accès direct à une mémoire distante (RDMA)", avril 2014. (*Remplace* RFC5046) *(P.S.)*

[RFC7146] D. Black, P. Koning, "Sécurisation des protocoles de mémorisation de blocs sur IP : Mise à jour des exigences de la RFC3723 pour IPsec v3", avril 2014. (*MàJ* RFC3720, RFC3723, RFC3821, RFC3822, RFC4018, RFC4172, RFC4173, RFC4174, RFC5040, RFC5041, RFC5042, RFC5043, RFC5044, RFC5045, RFC5046, RFC5047, RFC5048) *(P.S.)*

[SAM2] INCITS Technical Committee T10, "SCSI Architecture Model - 2 (SAM-2)", ANSI INCITS 366-2003, ISO/IEC 14776-412, 2003.

[SAM4] INCITS Technical Committee T10, "SCSI Architecture Model - 4 (SAM-4)", ANSI INCITS 447-2008, ISO/IEC 14776-414, 2008.

[SPC2] INCITS Technical Committee T10, "SCSI Primary Commands - 2", ANSI INCITS 351-2001, ISO/IEC 14776-452, 2001.

[SPC3] INCITS Technical Committee T10, "SCSI Primary Commands - 3", ANSI INCITS 408-2005, ISO/IEC 14776-453, 2005.

[UML] ISO, "Unified Modeling Language (UML) Version 1.4.2", ISO/IEC 19501:2005.

[UNICODE] The Unicode Consortium, "Unicode Standard Annex #15: Unicode Normalization Forms", 2013, <http://www.unicode.org/unicode/reports/tr15>.

16.2 Références pour information

[Castagnoli93] Castagnoli, G., Brauer, S., et M. Herrmann, "Optimization of Cyclic Redundancy-Check Codes with 24 and 32 Parity Bits", IEEE Transact. on Communications, Vol. 41,n° 6, juin 1993.

[FC-SP-2] INCITS Technical Committee T11, "Fibre Channel Security Protocols 2", ANSI INCITS 496-2012, 2012.

[IB] InfiniBand, "InfiniBand(TM) Architecture Specification", Vol. 1, Rel. 1.2.1, InfiniBand Trade Association, <http://www.infinibandta.org>.

[RFC1737] K. Sollins et L. Masinter, "Exigences fonctionnelles pour les noms de ressource uniformes", décembre 1994.

[RFC2401] S. Kent et R. Atkinson, "Architecture de sécurité pour le protocole Internet", novembre 1998. (*Obsolète, voir* RFC4301)

[RFC2407] D. Piper, "Le domaine d'interprétation de sécurité IP de l'Internet pour ISAKMP", novembre 1998. *(Obsolète, voir* 4306)

[RFC2409] D. Harkins et D. Carrel, "L'échange de clés Internet (IKE)", novembre 1998. *(Obsolète, voir la* RFC4306)

[RFC2608] E. Guttman et autres, "Protocole de localisation de service, version 2", juin 1999. (*MàJ par* RFC3224) *(P.S.)*

[RFC2743] J. Linn, "Interface générique de programme d'application de service de sécurité, version 2, mise à jour 1", janvier 2000. (*MàJ par* RFC5554)

[RFC2865] C. Rigney et autres, "Service d'authentification à distance de l'utilisateur appelant (RADIUS)", juin 2000. (*MàJ par* RFC2868, RFC3575, RFC5080) *(D.S.)*

[RFC3385] D. Sheinwald et autres, "Considérations sur la somme de contrôle du contrôle de redondance cyclique (CRC) de l'interface de système de petit ordinateur au protocole Internet (iSCSI)", septembre 2002. *(Information)*

[RFC3602] S. Frankel, R. Glenn, S. Kelly, "Algorithme de chiffrement AES-CBC et utilisation avec IPsec", septembre 2003. *(P.S.)*

[RFC3720] J. Satran et autres, "Interface Internet des systèmes de petits ordinateurs (iSCSI)", avril 2004. *(Remplacée par la présente* RFC)

[RFC3721] M. Bakke et autres, "Interface Internet des systèmes de petits ordinateurs (iSCSI) : dénomination et découverte", avril 2004. *(Information) (MàJ par* RFC7143)

[RFC3783] M. Chadalapaka, R. Elliott, "Considérations sur l'ordre des commandes d'interface de système de petit ordinateur (SCSI) avec iSCSI", mai 2004. *(Information)*

[RFC4121] L. Zhu et autres, "Version 2 du mécanisme d'interface de programme d'application de service de sécurité générique (GSS-API) de Kerberos version 5", juillet 2005. (*MàJ* RFC1964) (*MàJ par les* RFC6542, RFC6649)) *(P.S.)*

[RFC4297] A. Romanow et autres, "Position du problème de l'accès direct en mémoire distante (RDMA) sur IP", décembre 2005. *(Info.)*

[RFC4806] M. Myers, H. Tschofenig, "Extensions du protocole d'état de certificat en ligne (OCSP) à IKEv2", février 2007. *(P.S.)*

[RFC4850] D. Wysochanski, "Déclaration de clé d'extension publique pour l'architecture de nœud d'interface Internet de petit ordinateur (iSCSI)", avril 2007. (*MàJ* RFC3720) *(P.S.) (Remplacée par* RFC7143)

[RFC5046] M. Ko et autres, "Extensions pour l'accès direct à une mémoire distante (RDMA) à l'interface système de petit ordinateur à l'Internet (iSCSI)", octobre 2007. *(P.S.) (Remplacée par* RFC7145)

[RFC5048] M. Chadalapaka, éd. "Corrections et précisions à l'interface système de petit ordinateur à l'Internet (iSCSI)", octobre 2007. (*MàJ* RFC3720) *(P.S.) (Remplacée par* RFC7143)

[RFC5433] T. Clancy, H. Tschofenig, "Protocole d'authentification extensible – méthode des clés pré-partagées généralisée (EAP-GPSK)", février 2009. *(P.S.)*

[RFC6648] P. Saint-Andre, D. Crocker, M. Nottingham, "Le préfixe "X-" et les constructions similaires sont déconseillés dans les protocoles d'application", juin 2012. (BCP0178)

[SAS] INCITS Technical Committee T10, "Serial Attached SCSI - 2.1 (SAS-2.1)", ANSI INCITS 457-2010, 2010.

[SBC2] INCITS Technical Committee T10, "SCSI Block Commands - 2 (SBC-2)", ANSI INCITS 405-2005, ISO/IEC 14776-322, 2005.

[SPC4] INCITS Technical Committee T10, "SCSI Primary Commands - 4", ANSI INCITS 513-201x.

[SPL] INCITS Technical Committee T10, "SAS Protocol Layer - 2 (SPL-2)", ANSI INCITS 505-2013, ISO/IEC 14776-262, 2013.

Appendice A. Exemples

A.1 Exemple d'opération Read

Fonction initiatrice	**Type de PDU**	**Fonction cible**
Demande de commande (read)	Commande SCSI (read)>>>	Prépare le transfert de données
Reçoit les données	<<< SCSI Data-In	Envoie les données
Reçoit les données	<<< SCSI Data-In	Envoie les données
Reçoit les données	<<< SCSI Data-In	Envoie les données
	<<< SCSI réponse	Envoie l'état et le sens
Commande terminée		

A.2 Exemple d'opération Write

Fonction initiatrice	**Type de PDU**	**Fonction cible**
Demande de commande (write)	Commande SCSI (write)>>>	Reçoit la commade et la met en file d'attente*
		Traite les vieilles commandes
	<<< R2T	Prêt à traiter la commande write
Envoie les données	SCSI Data-Out >>>	Reçoit les données
	<<< R2T	Prêt pour les données
	<<< R2T	Prêt pour les données
Envoie les données	SCSI Data-Out >>>	Reçoit les données
	SCSI Data-Out >>>	Reçoit les données
	<<< Réponse SCSI	Envoie l'état et le sens
Commande terminée		

A.3. Exemples d'utilisation de R2TSN/DataSN

A.3.1 Exemple de données en sortie (écriture) de DataSN/R2TSN

Fonction d'initiateur	**Type et contenu de PDU**	**Fonction cible**
Demande de commande (write)	Commande SCSI (write)>>>	Reçoit la commande et la met en file d'attente
		Traite les vieilles commandes
	<<< R2T ; R2TSN = 0	Prêt pour les données
	<<< R2T ; R2TSN = 1	Prêt pour plus de données
Envoie les données pour R2TSN 0	SCSI Data-Out >>>	Reçoit les données
	DataSN = 0, F = 0	
Envoie les données pour R2TSN 0	SCSI Data-Out >>>	Reçoit les données

	DataSN = 1, F = 1	
Envoie les données pour R2TSN 1	SCSI Data >>>	Reçoit les données
	DataSN = 0, F = 1	
	<<< SCSI réponse	Envoie l'état et le sens
	ExpDataSN = 0	
Commande terminée		

A.3.2 Exemple d'entrée (lecture) de données DataSN

Fonction d'initiateur	Type de PDU	Fonction cible
Demande de commande (lire)	Commande SCSI (lire)>>>	
		Prépare le transfert de données
Reçoit les données	<<< SCSI Data-In ; DataSN = 0, F = 0	Envoie les données
Reçoit les données	<<< SCSI Data-In ; DataSN = 1, F = 0	Envoie les données
Reçoit les données	<<< SCSI Data-In ; DataSN = 2, F = 1	Envoie les données
	<<< SCSI réponse ; ExpDataSN = 3	Envoie l'état et le sens
Commande terminée		

A.3.3 Exemple de DataSN bidirectionnel

Fonction d'initiateur	Type de PDU	Fonction cible
Demande de commande (lire-écrire)	Commande SCSI (lire-écrire) >>>	
		Traite les vieilles commandes
	<<< R2T ; R2TSN = 0	Prêt à traiter la commande write
* Reçoit les données	<<< SCSI Data-In ; DataSN = 0, F = 0	Envoie les données
* Reçoit les données	<<< SCSI Data-In ; DataSN = 1, F = 1	Envoie les données
* Envoie les données pour R2TSN 0	SCSI Data-Out >>> ; DataSN = 0, F = 1	Reçoit les données
	<<< SCSI réponse ; ExpDataSN = 2	Envoie l'état et le sens
Commande terminée		

* Envoi des données et Reçoit les données peuvent être transférés simultanément comme dans un Read-Old-Write-New atomique ou séquentiellement comme dans un Read-Update-Write atomique (dans le dernier cas, le R2T peut suivre les données reçues).

A.3.4 Exempe de données non sollicitées et immédiates en sortie (écrire) avec DataSN

Fonction d'initiateur	Type et contenu de PDU	Fonction cible
Demande de commande (écriture) + données immédiates	Commande SCSI (écriture)>>> F = 0	Reçoit la commande et les données et les met en file d'attente
Envoi de donnés non sollicitées	Données d'écriture SCSI ; DataSN = 0, F = 1 >>>	Reçoit plus de données
		Traite les vieilles commandes
	<<< R2T ; R2TSN = 0	Prêt pour plus de données
Envoi de données for R2TSN 0	Données d'écriture SCSI ; DataSN = 0, F = 1 >>>	Reçoit les données
	<<< SCSI réponse	Envoie l'état et le sens
Commande terminée		

A.4 Exemples de CRC

Note : Toutes les valeurs sont en hexadécimal.

32 octets de zéros :

```
octet :    0  1  2  3
  0:      00 00 00 00
  ...
 28:      00 00 00 00
CRC :     aa 36 91 8a
```

32 octets de uns :

```
octet :     0  1  2  3
     0 :    ff ff ff ff
    ...
    28:     ff ff ff ff
CRC :       43 ab a8 62
```

32 octets s'incrémentant de 00 à 1f :

```
octet :      0  1    2   3
     0 :    00 01 02 03
    ...
    28 :     1c 1d 1e 1f
CRC :        4e 79 dd 46
```

32 octets qui décrémentent de 1f à 00 :

```
octet :      0  1  2  3
     0 :     1f 1e 1d 1c
    ...
   28 :     03 02 01 00
CRC :       5c db 3f 11
```

PDU de commande iSCSI - SCSI Lecture (10) :

```
octet:      0    1    2    3
   0 :     01   c0   00   00
   4 :     00   00   00   00
   8 :     00   00   00   00
  12 :     00   00   00   00
  16 :     14   00   00   00
  20 :     00   00   04   00
  24 :     00   00   00   14
  28 :     00   00   00   18
  32 :     28   00   00   00
  36 :     00   00   00   00
  40 :     02   00   00   00
  44 :     00   00   00   00
CRC :      56   3a   96   d9
```

Appendice B. Exemples de phase d'établissement

Dans le premier exemple, l'initiateur (I) et la cible (T) s'authentifient l'un l'autre via Kerberos :

```
I-> Login (CSG,NSG=0,1 T=1)
    InitiatorName=iqn.1999-07.com.os:hostid.77
    TargetName=iqn.1999-07.com.exemple:diskarray.sn.88
    AuthMethod=KRB5,SRP,None

T-> Login (CSG,NSG=0,0 T=0)
        AuthMethod=KRB5

I-> Login (CSG,NSG=0,1 T=1)
        KRB_AP_REQ=<krb_ap_req>
```

(krb_ap_req contient le ticket Kerberos V5 et l'authentifiant avec MUTUEL-EXIGÉ réglé dans le champ ap-options)

Si l'authentification réussit, la cible poursuit avec :

```
T-> Login (CSG,NSG=0,1 T=1)
        KRB_AP_REP=<krb_ap_rep>
```

(krb_ap_rep est la réponse d'authentification mutuelle Kerberos V5)

Si l'authentification réussit, l'initiateur peut poursuive avec :

I-> Login (CSG,NSG=1,0 T=0) FirstBurstLength=8192

T-> Login (CSG,NSG=1,0 T=0) FirstBurstLength=4096
MaxBurstLength=8192

I-> Login (CSG,NSG=1,0 T=0) MaxBurstLength=8192
... plus de paramètres de fonctionnement iSCSI

T-> Login (CSG,NSG=1,0 T=0)
... plus de paramètres de fonctionnement iSCSI

Et à la fin :

I-> Login (CSG,NSG=1,3 T=1)
paramètres iSCSI facultatifs

T-> Login (CSG,NSG=1,3 T=1) "login accept"

Si l'authentification de l'initiateur par la cible ne réussit pas, la cible répond par :

T-> Login "login reject"

au lieu du message d'établissement KRB_AP_REP, et elle termine la connexion.

Si l'authentification de la cible par l'initiateur ne réussit pas, l'initiateur termine la connexion (sans répondre au message d'établissemenh KRB_AP_REP).

Dans le prochain exemple, seul l'initiateur est authentifié par the cible via Kerberos :

I-> Login (CSG,NSG=0,1 T=1)
InitiatorName=iqn.1999-07.com.os:hostid.77
TargetName=iqn.1999-07.com.exemple:diskarray.sn.88
AuthMethod=SRP,KRB5,None

T-> Login-PR (CSG,NSG=0,0 T=0)
AuthMethod=KRB5

I-> Login (CSG,NSG=0,1 T=1)
KRB_AP_REQ=krb_ap_req

(MUTUEL-EXIGÉ n'est pas établi dans le champ ap-options de krb_ap_req)

Si l'authentification réussit, la cible poursuit avec :

T-> Login (CSG,NSG=0,1 T=1)

I-> Login (CSG,NSG=1,0 T=0)
... paramètres iSCSI

T-> Login (CSG,NSG=1,0 T=0)
... paramètres iSCSI

. . .

T-> Login (CSG,NSG=1,3 T=1)"login accept"

Dans le prochain exemple, l'initiateur et la cible s'authentifient l'un l'autre via SRP :

I-> Login (CSG,NSG=0,1 T=1)
InitiatorName=iqn.1999-07.com.os:hostid.77
TargetName=iqn.1999-07.com.exemple:diskarray.sn.88
AuthMethod=KRB5,SRP,None

T-> Login-PR (CSG,NSG=0,0 T=0)
AuthMethod=SRP

I-> Login (CSG,NSG=0,0 T=0)
SRP_U=<user>
TargetAuth=Yes

T-> Login (CSG,NSG=0,0 T=0)
SRP_N=<N>
SRP_g=<g>
SRP_s=<s>

I-> Login (CSG,NSG=0,0 T=0)
SRP_A=<A>

T-> Login (CSG,NSG=0,0 T=0)
SRP_B=<B>

I-> Login (CSG,NSG=0,1 T=1)
SRP_M=<M>

Si l'authentification de l'initiateur réussit, la cible poursuit avec :

T-> Login (CSG,NSG=0,1 T=1)
SRP_HM=<H(A | M | K)>

où N, g, s, A, B, M, et H(A | M | K) sont définis dans la [RFC2945].

Si l'authentification de la cible ne réussit pas , l'initiateur termine la connexion ; autrement, il poursuit .

I-> Login (CSG,NSG=1,0 T=0)
... paramètres iSCSI

T-> Login (CSG,NSG=1,0 T=0)
... paramètres iSCSI

Et à la fin :

I-> Login (CSG,NSG=1,3 T=1)
paramètres iSCSI facultatifs

T-> Login (CSG,NSG=1,3 T=1) "login accept"

Si l'authentification de l'initiateur ne réussit pas, la cible répond par :

T-> Login "login reject"

au lieu du message T-> Login SRP_HM=<H(A | M | K)>, et il termine la connexion.

Dans le prochain exemple, seul l'initiateur est authentifié par la cible via SRP :

I-> Login (CSG,NSG=0,1 T=1)
InitiatorName=iqn.1999-07.com.os:hostid.77
TargetName=iqn.1999-07.com.exemple:diskarray.sn.88
AuthMethod=KRB5,SRP,None

T-> Login-PR (CSG,NSG=0,0 T=0)
AuthMethod=SRP

I-> Login (CSG,NSG=0,0 T=0)
SRP_U=<user>
TargetAuth=No

T-> Login (CSG,NSG=0,0 T=0)
SRP_N=<N>
SRP_g=<g>
SRP_s=<s>

I-> Login (CSG,NSG=0,0 T=0)
SRP_A=<A>

T-> Login (CSG,NSG=0,0 T=0)
SRP_B=<B>

I-> Login (CSG,NSG=0,1 T=1)
SRP_M=<M>

Si l'authentification de l'initiateur réussit , la cible poursuit avec :

T-> Login (CSG,NSG=0,1 T=1)

I-> Login (CSG,NSG=1,0 T=0)
... paramètres iSCSI

T-> Login (CSG,NSG=1,0 T=0)
... paramètres iSCSI

Et à la fin :

I-> Login (CSG,NSG=1,3 T=1)
paramètres iSCSI facultatifs

T-> Login (CSG,NSG=1,3 T=1) "login accept"

Dans le prochain exemple, l'initiateur et la cible s'authentifient l'un l'autre via CHAP :

I-> Login (CSG,NSG=0,0 T=0)
InitiatorName=iqn.1999-07.com.os:hostid.77
TargetName=iqn.1999-07.com.exemple:diskarray.sn.88
AuthMethod=KRB5,CHAP,None

T-> Login-PR (CSG,NSG=0,0 T=0)
AuthMethod=CHAP

I-> Login (CSG,NSG=0,0 T=0)
CHAP_A=<A1,A2>

T-> Login (CSG,NSG=0,0 T=0)
CHAP_A=<A1>
CHAP_I=<I>
CHAP_C=<C>

I-> Login (CSG,NSG=0,1 T=1)
CHAP_N=<N>
CHAP_R=<R>
CHAP_I=<I>
CHAP_C=<C>

Si l'authentification de l'initiateur réussit, la cible poursuit avec :

T-> Login (CSG,NSG=0,1 T=1)

CHAP_N=<N>
CHAP_R=<R>

Si l'authentification de la cible ne réussit pas, l'initiateur interrompt la connexion ; autrement, il poursuit :

I-> Login (CSG,NSG=1,0 T=0)
... paramètres iSCSI

T-> Login (CSG,NSG=1,0 T=0)
... paramètres iSCSI

Et à la fin :

I-> Login (CSG,NSG=1,3 T=1)
paramètres iSCSI facultatifs

T-> Login (CSG,NSG=1,3 T=1) "login accept"

Si l'authentification de l'initiateur ne réussit pas, la cible répond avec :

T-> Login "login reject"

au lieu du message d'établissement CHAP_R=<response> "poursuivre et changer d'état", et il termine la connexion.

Dans le prochain exemple, seul l'initiateur est authentifié par la cible via CHAP :

I-> Login (CSG,NSG=0,1 T=0)
InitiatorName=iqn.1999-07.com.os:hostid.77
TargetName=iqn.1999-07.com.exemple:diskarray.sn.88
AuthMethod=KRB5,CHAP,None

T-> Login-PR (CSG,NSG=0,0 T=0)
AuthMethod=CHAP

I-> Login (CSG,NSG=0,0 T=0)
CHAP_A=<A1,A2>

T-> Login (CSG,NSG=0,0 T=0)
CHAP_A=<A1>
CHAP_I=<I>
CHAP_C=<C>

I-> Login (CSG,NSG=0,1 T=1)
CHAP_N=<N>
CHAP_R=<R>

Si l'authentification de l'initiateur réussit, la cible poursuit avec :

T-> Login (CSG,NSG=0,1 T=1)

I-> Login (CSG,NSG=1,0 T=0)
... paramètres iSCSI

T-> Login (CSG,NSG=1,0 T=0)
... paramètres iSCSI

Et à la fin :

I-> Login (CSG,NSG=1,3 T=1)
paramètres iSCSI facultatifs

T-> Login (CSG,NSG=1,3 T=1) "login accept"

Dans le prochain exemple, l'initiateur n'offre aucun paramètre de sécurité. Il peut donc offrir des paramètres iSCSI sur la PDU d'établissement avec le bit T réglé à 1, et la cible peut répondre immédiatement avec une PDU de réponse finale d'établissement :

```
I-> Login (CSG,NSG=1,3 T=1)
    InitiatorName=iqn.1999-07.com.os:hostid.77
    TargetName=iqn.1999-07.com.exemple:diskarray.sn.88
    ... paramètres iSCSI

T-> Login (CSG,NSG=1,3 T=1) "login accept"
    ... paramètres iSCSI
```

Dans l'exemple suivant, l'initiateur offre des paramètres de sécurité sur la PDU d'établissement, mais la cible n'en choisit aucun (c'est-à-dire, choisit les valeurs "None") :

```
I-> Login (CSG,NSG=0,1 T=1)
    InitiatorName=iqn.1999-07.com.os:hostid.77
    TargetName=iqn.1999-07.com.exemple:diskarray.sn.88
    AuthMethod=KRB5,SRP,None

T-> Login-PR (CSG,NSG=0,1 T=1)
    AuthMethod=None

I-> Login (CSG,NSG=1,0 T=0)
     ... paramètres iSCSI

T-> Login (CSG,NSG=1,0 T=0)
     ... paramètres iSCSI
```

Et à la fin :

```
I-> Login (CSG,NSG=1,3 T=1)
       paramètres iSCSI facultatifs

T-> Login (CSG,NSG=1,3 T=1) "login accept"
```

Appendice C. Fonctionnement de SendTargets

Le texte de cet appendice est une partie normative du présent document.

Pour réduire la quantité de configuration requise pour un initiateur, iSCSI fournit la demande Text SendTargets. L'initiateur utilise la demande SendTargets pour obtenir une liste des cibles auxquelles il peut avoir accès, ainsi que la liste des adresses (adresse IP et accès TCP) sur lesquelles on peut accéder à ces cibles.

Pour utiliser SendTargets, un initiateur doit d'abord établir un de ces deux types de sessions. Si l'initiateur établit la session avec la clé "SessionType=Discovery", la session est une session Discovery, et un nom de cible n'a pas besoin d'être spécifié. Autrement, la session est une session de fonctionnement normale. La commande SendTargets NE DOIT être envoyée QUE durant la phase de pleines caractéristiques d'une session Normal ou Discovery.

Un système qui contient les cibles DOIT prendre en charge les sessions de découverte sur chacune de ses paires d'adresse/accès IP iSCSI et DOIT prendre en charge la commande SendTargets sur la session Discovery. Dans une session Discovery, une cible DOIT retourner toutes les informations de chemin (paires adresse/accès IP et étiquettes de groupe portails cible) pour les cibles sur l'entité réseau cible auquel l'initiateur demandeur est autorisé à accéder.

Une cible DOIT prendre en charge la commande SendTargets sur les sessions de fonctionnement ; celle-ci va seulement retourner des informations de chemin sur la cible à laquelle la session est connectée et n'a pas besoin de retourner des informations sur les noms d'autres cibles qui peuvent être définis dans le système qui répond.

Un initiateur PEUT faire usage de la commande SendTargets à son gré.

Une commande SendTargets consiste en une seule PDU Demande Text. Cette PDU contient exactement une clé et valeur text. La clé text DOIT être SendTargets. Les réponses attendues dépendent de la valeur, ainsi que de si la session est de découverte ou opérationnelle.

La valeur doit être une de :

All : L'initiateur demande que les informations sur toutes les cibles pertinentes connues de la mise en œuvre soient retournées. Cette valeur DOIT être prise en charge sur une session Discovery et NE DOIT PAS être prise en charge sur une session opérationnelle.

<iSCSI-target-name> : Si un nom de cible iSCSI est spécifié, la session devrait répondre avec des adresses pour la seule cible nommée, si possible. Cette valeur DOIT être prise en charge sur les sessions de découverte. Une session Discovery DOIT être capable de retourner des adresses pour les cibles qui auraient été retournées si leur valeur=All avait été désignée.

<nothing> : La session devrait seulement répondre avec l'adresse des cibles auxquelles la session est connectée. Ceci DOIT être pris en charge sur les sessions opérationnelles et NE DOIT PAS retourner des cibles autres que celle à laquelle la session est enregistrée.

La réponse à cette commande est une réponse Text qui contient une liste de zéro, une ou plusieurs cibles et, facultativement, leur adresse. Chaque cible est retournée comme enregistrement de cible. Un enregistrement de cible commence par la clé de texte TargetName, suivie par une liste de clés de texte TargetAddress, et bordée par la fin de la réponse Text ou la prochaine clé TargetName, qui commence un nouvel enregistrement. Aucune autre clé que TargetName et TargetAddress n'est permise dans une réponse SendTargets.

Le format de TargetName est décrit au paragraphe 13.4.

Une session Discovery PEUT répondre à une demande SendTargets avec sa liste complète des cibles, ou avec une liste des cibles qui se fonde sur le nom de l'initiateur enregistré à la session.

Une réponse SendTargets NE DOIT PAS contenir des noms de cible si il n'y a plus de cible offertes à l'accès de l'initiateur demandeur.

Chaque enregistrement de cible retourné inclut zéro, un ou plusieurs champs TargetAddress.

Chaque enregistrement de cible commence par une clé Text de forme :

TargetName=<nom de cible>

suivi par zéro, une ou plusieurs clés d'adresse de forme :

TargetAddress=<nom d'hôte ou adresse IP>[:<accès TCP>], <étiquette de groupe portail>

<nom d'hôte ou adresse IP> contient un nom de domaine, une adresse IPv4, ou une adresse IPv6 ([RFC4291]) comme spécifié pour la clé TargetAddress.

Un <nom d'hôte ou adresse IP> dupliqué dans les réponses TargetAddress pour un certain nœud (l'accès est absent ou égal) indiquerait probablement que plusieurs familles d'adresses sont utilisées en une seule fois (IPv6 et IPv4).

Chaque TargetAddress appartient à un groupe portail, identifié par son étiquette numérique de groupe portail cible (voir le paragraphe 13.9). Le nom de cible iSCSI, avec cette étiquette, constitue l'identifiant d'accès SCSI ; l'étiquette a seulement besoin d'être unique au sein d'une certaine liste d'adresses de noms de cible.

Des sessions à connexions multiples peuvent s'étendre sur des adresses iSCSI qui appartiennent au même groupe portail.

Des session à connexions multiples ne peuvent pas s'étendre sur des adresses iSCSI qui appartiennent à des groupes portail différents.

Si une réponse SendTargets rapporte une adresse iSCSI pour une cible, elle DEVRAIT aussi rapporter toutes les autres adresses de ce groupe portail dans la même réponse.

Une réponse Text SendTargets peut être plus longue qu'une seule PDU de réponse Text et faire usage comme spécifié de la réponse de long texte.

Après l'obtention d'une liste des cibles de la session Discovery, un initiateur iSCSI peut initier de nouvelles sessions pour se connecter aux cibles découvertes pour un fonctionnement complet. L'initiateur PEUT garder la session Discovery ouverte et PEUT envoyer des commandes SendTargets ultérieures pour découvrir de nouvelles cibles.

Exemples. Cet exemple est la réponse SendTargets provenant d'une seule cible qui n'a pas d'autre accès d'interface.

L'initiateur envoie une demande Text qui contient :

SendTargets=All

La cible envoie une réponse Text qui contient :

TargetName=iqn.1993-11.com.exemple:diskarray.sn.8675309

Tout ce que la cible avait à retourner dans ce cas simple était le nom de la cible. L'initiateur suppose que l'adresse IP et accès TCP pour cette cible sont les mêmes que ceux utilisés sur la connexion actuelle à la cible iSCSI par défaut.

Le prochain exemple a deux cibles internes iSCSI, chacune accessible via deux accès différents avec des adresses IP différentes. Voici la réponse Text :
TargetName=iqn.1993-11.com.exemple:diskarray.sn.8675309
TargetAddress=10.1.0.45:3000,1
TargetAddress=10.1.1.45:3000,2
TargetName=iqn.1993-11.com.exemple:diskarray.sn.1234567
TargetAddress=10.1.0.45:3000,1
TargetAddress=10.1.1.45:3000,2

Les deux cibles partagent les deux adresses ; les adresses multiples sont probablement utilisées pour assurer la prise en charge de chemins multiples. L'initiateur peut se connecter au nom de cible sur l'une ou l'autre adresse. Chacune des adresses a sa propre étiquette de groupe portail cible ; elles ne prennent pas en charge l'extension de la session sur des connexions multiples l'une avec l'autre. On se souvient que les étiquettes de groupe portail cible pour les deux cibles désignées sont indépendantes l'une de l'autre ; le groupe portail "1" sur la première cible n'est pas nécessairement le même que le groupe portail "1" sur la seconde cible.

Dans l'exemple ci-dessus, un nom d'hôte DNS ou une adresse IPv6 aurait pu être retourné au lieu d'une adresse IPv4.

La prochaine réponse Text montre une cible qui prend en charge des sessions qui s'étendent sur plusieurs adresses et illustre de plus l'usage des étiquettes de groupe portail cible :
TargetName=iqn.1993-11.com.exemple:diskarray.sn.8675309
TargetAddress=10.1.0.45:3000,1
TargetAddress=10.1.1.46:3000,1
TargetAddress=10.1.0.47:3000,2
TargetAddress=10.1.1.48:3000,2
TargetAddress=10.1.1.49:3000,3

Dans cet exemple, toute adreese de cible peut être utilisée pour atteindre la même cible. Une session à une seule connexion peut être établie à n'importe laquelle de ces adresses TCP. Une session multi connexions pourrait s'étendre sur les adresses . 45 et .46 ou .47 et .48 mais ne peut pas s'étendre sur une autre combinaison. Une TargetAddress avec sa propre étiquette (.49) ne peut pas être combinée avec une autre adresse dans la même session.

Cette réponse SendTargets n'indique pas si .49 prend en charge plusieurs connexions par session ; c'est communiqué via la clé text MaxConnections au moment de l'établissement sur la cible.

Appendice D. Présentation algorithmique des classes de récupération d'erreur

Cet appendice illustre les classes de récupération d'erreur en utilisant un pseudo langage de programmation. Les noms de procédure sont choisis pour être évidents à la plupart des développeurs. Chaque classe de récupération décrite a des procédures d'initiateur ainsi que des procédures de cible. Ces algorithmes se concentrent sur la mise en évidence des mécaniques de classe de récupération d'erreur et ne décrivent pas de façon exhaustive tous les autres aspects/cas. Les exemples de cette approche sont les suivants :

- On ne montre le traitement que pour certains types d'Opcode.
- On ne mentionne que certains codes de cause (par exemple, Recovery dans une commande Logout).

- Les cas résultants, comme la récupération de la synchronisation sur une erreur de résumé d'en-tête, sont considérés comme sortant du domaine d'application de ces algorithmes. Dans cet exemple particulier, une erreur de résumé d'en-tête peut conduire à la récupération de connexion si un certain type de couche de synchronisation et pilotage n'est pas mis en œuvre.

Ces algorithmes s'efforcent de véhiculer les concepts de récupération d'erreur iSCSI dans les termes les plus simples et ne sont pas conçus comme une optimisation.

D.1 Structure générale des données et description des procédures

Ce paragraphe définit les procédures et structures de données qui sont couramment utilisées par tous les algorithmes de récupération d'erreur. Les structures peuvent n'être pas la représentation exhaustive de ce qui est requis d'une mise en œuvre normale.

Définition des structures de données :

```
struct TransferContext {
      int TargetTransferTag;
      int ExpectedDataSN;
  };

struct TCB {                              /* bloc de contrôle de tâche */
      Boolean SoFarInOrder;
      int ExpectedDataSN;                 /* utilisé pour les R2T et les données */
      int MissingDataSNList[MaxMissingDPDU];
      Boolean FbitReceived;
      Boolean StatusXferd;
      Boolean CurrentlyAllegiant;
      int ActiveR2Ts;
      int response;
      char *Reason;
      struct TransferContext TransferContextList[MaxOutstandingR2T];
      int InitiatorTaskTag;
      int CmdSN;
      int SNACK_Tag;
};

struct Connection {
      struct Session SessionReference;
      Boolean SoFarInOrder;
      int CID;
      int State;
      int CurrentTimeout;
      int ExpectedStatSN;
      int MissingStatSNList[MaxMissingSPDU];
      Boolean PerformConnectionCleanup;
};

struct Session {
      int NumConnections;
      int CmdSN;
      int Maxconnexions;
      int ErrorRecoveryLevel;
      struct iSCSIEndpoint OtherEndInfo;
      struct Connection ConnectionList[Maxprendre en chargeedConns];
};
```

Descriptions de procédures :
Receive-an-In-PDU(connexion de transport, PDU entrante); check-basic-validity(PDU entrante);
Start-Timer(gestionnaire de temporisation, argument, valeur de temporisation);
Build-And-Send-Reject(connexion de transport, mauvaise PDU, code de cause);

D.2 Algorithmes de récupération d'erreur au sein de la commande

D.2.1 Descriptions de procédures

Recover-Data-if-Possible(dernier numéro de séquence de données requis, bloc de contrôle de tâche);
Build-And-Send-DSnack(bloc de contrôle de tâche);
Build-And-Send-RDSnack(bloc de contrôle de tâche);
Build-And-Send-Abort(bloc de contrôle de tâche);
SCSI-Task-Completion(bloc de contrôle de tâche);
Build-And-Send-A-Data-Burst(connexion de transport, descripteur de données, bloc de contrôle de tâche);
Build-And-Send-R2T(connexion de transport, descripteur de données, bloc de contrôle de tâche);
Build-And-Send-Status(connexion de transport, bloc de contrôle de tâche);
Transfer-Context-Timeout-Handler(contexte de transfert);

Notes :
- Une procédure utilisée dans ce paragraphe, Handle-Status-SNACK-request est définie dans l'Appendice D.3.
- Le pseudocode de traitement de réponse montré dans les algorithmes de cible s'applique à toutes les PDU sollicitées qui portent le numéro de séquence d'état – réponse SCSI, réponse Text, etc.

D.2.2 Algorithmes d'initiateur

```
Recover-Data-if-Possible(LastRequiredDataSN, TCB)
{
   si (operational ErrorRecoveryLevel > 0) {
       si (les numéros de PDU manquantes sont traçables) { Noter les DataSN manquants dans TCB.
              si (la tâche s'étend sur un changement dans MaxRecvDataSegmentLength) {
                  si (TCB.StatusXferd est VRAI)
                     abandonner la PDU d'état ;
                  Build-And-Send-RDSnack(TCB);
              } autrement {
                  Build-And-Send-DSnack(TCB);
              }
         } autrement {
            TCB.Reason = "Erreur de CRC de service de protocole";
               }
     } autrement {
          TCB.Reason = "Erreur de CRC de service de protocole";
     }
     si (TCB.Reason == "Erreur de CRC de service de protocole") {
          Nettoyer la liste des PDU manquantes dans le TCB.
          si (TCB.StatusXferd n'est pas VRAI)
            Build-And-Send-Abort(TCB);
     }
  }

  Receive-an-In-PDU(Connection, CurrentPDU)
  {
  check-basic-validity(CurrentPDU);
  si (Header-Digest-Bad) éliminer, retour;
  Restituer le TCB pour CurrentPDU.InitiatorTaskTag.
  si ((CurrentPDU.type == Data)
           ou (CurrentPDU.type = R2T)) {
    si (Data-Digest-Bad pour Data) {
            send-data-SNACK = VRAI;
      LastRequiredDataSN = CurrentPDU.DataSN;
          } autrement {
          si (TCB.SoFarInOrder = VRAI) {
             si (DataSN courant est attendu) {
                 Incrémenter TCB.ExpectedDataSN.
             } autrement {
                   TCB.SoFarInOrder = FAUX;
                   send-data-SNACK = VRAI;
                  }
```

```
        } autrement {
              si (DataSN courant était considéré comme manquant) {
                 retirer DataSN courant de la liste des PDU manquantes.
              } autrement si (DataSN courant est supérieur à celui attendu) {
                    send-data-SNACK = VRAI;
                 } autrement {
                    éliminer, retour;
                 }
                 Ajuster TCB.ExpectedDataSN si approprié.
          }
          LastRequiredDataSN = CurrentPDU.DataSN - 1;
           }
           si (send-data-SNACK est VRAI et
             la tâche n'est pas déjà considérée comme échouée) {
          Recover-Data-if-Possible(LastRequiredDataSN, TCB);
    }
         si (la liste des PDU Données manquantes est vide) {
           TCB.SoFarInOrder = VRAI;
         }
     si (CurrentPDU.type == R2T) {
       Incrémenter les ActiveR2T pour cette tâche.
       Créer un descripteur de données pour la salve de données.
       Build-And-Send-A-Data-Burst(Connexion, descripteur de données, TCB);
     }
   } autrement si (CurrentPDU.type == réponse) {
     si (Data-Digest-Bad) {
            send-status-SNACK = VRAI;
          } autrement {
       TCB.StatusXferd = VRAI;
       Mémoriser les informations d'état dans TCB.
       si (ExpDataSN ne correspond pas) {
          TCB.SoFarInOrder = FAUX;
          Recover-Data-if-Possible(DataSN courant, TCB);
       }
            si (liste des PDU de données manquantes est vide) {
               TCB.SoFarInOrder = VRAI;
            }
     }
   } autrement { /*Le reste est sans relation avec la récupération de commande et n'est pas montré */
   }
   si ((TCB.SoFarInOrder == VRAI) et
                 (TCB.StatusXferd == VRAI)) { SCSI-Task-Completion(TCB);
    }
  }
```

D.2.3 Algorithmes de cible

```
  Receive-an-In-PDU(Connection, CurrentPDU)
  {
   check-basic-validity(CurrentPDU);
   si (Header-Digest-Bad) éliminer, retour;
   Restituer le TCB pour CurrentPDU.InitiatorTaskTag.
   si (CurrentPDU.type == Data) {
      Restituer TContext à partir de CurrentPDU.TargetTransferTag;
      si (Data-Digest-Bad) {
            Build-And-Send-Reject(Connection, CurrentPDU, Payload-Digest-Error);
        Noter les PDU Données manquantes dans MissingDataRange[].
            send-recovery-R2T = VRAI;
          } autrement {
        si (DataSN courant n'est pas attendu) {
             Noter les PDU Données manquantes dans MissingDataRange[].
            send-recovery-R2T = VRAI;
```

```
            }
        si (CurrentPDU.Fbit == VRAI) {
          si (PDU courante est sollicitée) {
              Décrémenter TCB.ActiveR2Ts.
          }
          si ((PDU courante n'est pas sollicitée et les données reçues sont moins que longueur I/O et
                les données reçues sont moins que FirstBurstLength)
             ou (PDU courante est sollicitée et la longueur de cette salve est moins que ce qui est attendu)) {
             send-recovery-R2T = VRAI;
             Noter les données manquantes dans MissingDataRange[].
          }
             }
           }
           Incrémenter TContext.ExpectedDataSN.
      si (send-recovery-R2T est VRAI et la tâche n'est pas déjà considérée comme échouée) {
        si (ErrorRecoveryLevel opérationnel > 0) {
          Incrémenter les TCB.ActiveR2T.
          Créer un descripteur de données pour la salve de données à partir de MissingDataRange.
          Build-And-Send-R2T(Connexion, descripteur de données, TCB);
        } autrement {
           si (PDU courante est la dernière non sollicitée)
             TCB.Reason = "Pas assez de données non sollicitées";
           autrement
             TCB.Reason = "Erreur de CRC de service de protocole";
        }
      }
      si (TCB.ActiveR2Ts == 0) {
        Build-And-Send-Status(Connexion, TCB);
      }
    } autrement si (CurrentPDU.type == SNACK) {
      snack-failure = FAUX;
      si (ErrorRecoveryLevel opérationnel > 0) {
        si (CurrentPDU.type == Data/R2T) {
          si (la demande peut être satisfaite) {
            si (demande de données) {
              Créer un descripteur de données pour la salve de données à partir de BegRun et RunLength.
              Build-And-Send-A-Data-Burst(Connexion, descripteur de données, TCB);
            } autrement { /* R2T */
              Créer un descripteur de données pour la salve de données à partir de BegRun et RunLength.
              Build-And-Send-R2T(Connexion, descripteur de données, TCB);
             }
          } autrement {
              snack-failure = VRAI;
          }
        } autrement si (CurrentPDU.type == status) {
            Handle-Status-SNACK-request(Connexion, CurrentPDU);
        } autrement si (CurrentPDU.type == DataACK) {
            Considérer toutes les données jusqu'à CurrentPDU.BegRun comme acquittées.
            Libérer les ressources de retransmission pour ces données.
         } autrement si (CurrentPDU.type == R-Data SNACK) {
                Créer un descripteur de données pour une salve de données couvrant toutes les données non acquittées.
            Build-And-Send-A-Data-Burst(Connexion, descripteur de données, TCB);
            TCB.SNACK_Tag = CurrentPDU.SNACK_Tag;
            si (il n'y a plus de données à envoyer) {
              Build-And-Send-Status(Connexion, TCB);
            }
        }
      } autrement { /* ErrorRecoveryLevel opérationnel = 0 */
            snack-failure = VRAI;
      }
      si (snack-failure == VRAI) {
         Build-And-Send-Reject(Connexion, CurrentPDU, SNACK-Reject);
         si (TCB.StatusXferd != VRAI) {
```

```
            TCB.Reason = "SNACK rejeté";
            Build-And-Send-Status(Connection, TCB);
          }
        }
      } autrement { /* Le reste est sans relation avec la récupération au sein de la commande et n'est pas montré */
      }
    }

    Transfer-Context-Timeout-Handler(TContext)
    {
     Restituer le TCB et la connexion à partir de TContext.
     Décrémenter les TCB.ActiveR2T.
     si (ErrorRecoveryLevel opérationnel > 0 et la tâche n'est pas encore considérée comme échouée) {
        Noter les PDU de données manquantes dans MissingDataRange[].
        Créer un descripteur de données pour la salve de données à partir de MissingDataRange[].
        Build-And-Send-R2T(Connexion, descripteur de données, TCB);
      } autrement {
         TCB.Reason = "Erreur de CRC de service de protocole";
         si (TCB.ActiveR2Ts = 0) {
           Build-And-Send-Status(Connexion, TCB);
         }
      }
    }
```

D.3 Algorithmes de récupération au sein de la connexion

D.3.1 Descriptions des procédures

Descriptions de procédures :
Recover-Status-if-Possible(connexion de transport, PDU actuellement reçue);
Evaluate-a-Numéro de séquence d'état(connexion de transport, PDU actuellement reçue);
Retransmit-Command-if-Possible(connexion de transport, CmdSN);
Build-And-Send-SSnack(connexion de transport);
Build-And-Send-Command(connexion de transport, bloc de contrôle de tâche);
Command-Acknowledge-Timeout-Handler(bloc de contrôle de tâche);
Status-Expect-Timeout-Handler(connexion de transport);
Build-And-Send-NOP-Out(connexion de transport);
Handle-Status-SNACK-request(connexion de transport, PDU SNACK d'état);
Retransmit-Status-Burst(SNACK d'état, bloc de contrôle de tâche);
Is-Acknowledged(commencer numéro de séquence d'état, longueur de cours);
Paramètre spécifiques de mise en œuvre qui sont réglables:
InitiatorProactiveSNACKEnabled

Notes :

- Les algorithmes d'initiateur ne traitent que des PDU NOP-In non sollicitées pour générer des SNACK d'état. Une PDU NOP-In non sollicitée a un numéro de séquence d'état alloué qui, quand il n'est pas à son ordre, pourrait déclencher le traitement de numéro de séquence d'état déclassé dans des algorithmes au sein de la commande, conduisant à un Recover-Status-if-Possible.
- Le pseudocode montré peut résulter en la retransmission de commandes non acquittées dans plus de cas que nécessaire. Cela ne va cependant pas affecter la justesse de l'opération parce que la cible doit éliminer les CmdSN dupliqués.
- La procédure Build-And-Send-Async est définie dans les algorihmes de récupération de connexion.
- La procédure Status-Expect-Timeout-Handler décrit comment les initiateurs peuvent proactivement tenter de restituer l'état si ils le veulent. Cette procédure est supposée être déclenchée avant la fin de la temporisation ULP standard.

D.3.2 Algorithmes d'initiateur

```
Recover-Status-if-Possible(Connexion, PDU courante)
{
  si ((Connection.state == LOGGED_IN) et la connexion n'est pas encore considérée comme en échec) {
    si (operational ErrorRecoveryLevel > 0) {
      si (numéro des PDU manquanres est traçable) {
        Noter les StatSN manquants dans la connexion qui n'étaient pas déjà demandés avec SNACK;
```

```
          Build-And-Send-SSnack(Connexion);
              } autrement { Connection.PerformConnectionCleanup = VRAI; }
      } autrement { Connection.PerformConnectionCleanup = VRAI; }
      si (Connection.PerformConnectionCleanup == VRAI) {
        Start-Timer(Connection-Cleanup-Handler, Connexion, 0);
         }
    }
}

Retransmit-Command-if-Possible(Connexion, CmdSN)
{
   si (ErrorRecoveryLevel opérationnel > 0) {
     Restituer l'étiquette de tâche d'initiateur, et donc le TCB pour le CmdSN.
     Build-And-Send-Command(Connexion, TCB);
   }
}

    Evaluate-a-StatusSN(Connexion, PDU courante)
    {
       send-status-SNACK = FAUX;
       si (Connection.SoFarInOrder == VRAI) {
         si (Numéro de séquence d'état en cours est celui attendu) {
             Incrémenter Connection.ExpectedStatSN.
         } autrement {
                   Connection.SoFarInOrder = FAUX;
                   send-status-SNACK = VRAI;
              }
       } autrement {
         si (Numéro de séquence d'état en cours était considéré comme manquant) {
             retirer le numéro de séquence d'état en cours de la liste des manquants.
         } autrement {
                   si (Numéro de séquence d'état courant est supérieur à ce qui est attendu){
                      send-status-SNACK = VRAI;
                   } autrement {
                      send-status-SNACK = FAUX;
                éliminer la PDU;
             }
         }
         Ajuster Connection.ExpectedStatSN si approprié.
         si (la liste des numéros de séquence d'état manquants est vide) {
             Connection.SoFarInOrder = VRAI;
       }
    }
    retourner send-status-SNACK;
}

Receive-an-In-PDU(Connexion, PDU courante)
  {
     check-basic-validity(PDU courante);
       si (Header-Digest-Bad) éliminer, retour;
       Restituer le TCB pour CurrentPDU.InitiatorTaskTag.
       si (CurrentPDU.type == NOP-In) {
           si (la PDU est non sollicitée) {
               si (Numéro de séquence d'état en cours n'est pas attendu) {
                   Recover-Status-if-Possible(Connexion, PDU courante);
            }
            si (Numéro de séquence de commande attendu en cours n'est pas le Session.CmdSN) {
                   Retransmit-Command-if-Possible(Connexion, PDU courante, Numéro de séquence de commande attendu);
               }
           }
       } autrement si (CurrentPDU.type == Reject) {
           si (c'est une erreur de résumé de données sur des données immédiates) {
               Retransmit-Command-if-Possible(Connexion, CurrentPDU.BadPDUHeader.CmdSN);
```

```
            }
        } autrement si (CurrentPDU.type == réponse) {
            send-status-SNACK = Evaluate-a-StatSN(Connexion, PDU courante);
            si (send-status-SNACK == VRAI)
                Recover-Status-if-Possible(Connexion, PDU courante);
        } autrement { /* le reste est sans rapport avec la récupération au sein de la connexion et n'est pas montré */
    }
}

Command-Acknowledge-Timeout-Handler(TCB)
{
    Restituer la connexion pour le TCB.
    Retransmit-Command-if-Possible(Connexion, TCB.CmdSN);
}

Status-Expect-Timeout-Handler(Connexion)
{
    si (ErrorRecoveryLevel opérationnel > 0) {
        Build-And-Send-NOP-Out(Connexion);
    } autrement si (InitiatorProactiveSNACKEnabled){
        si ((Connection.state == LOGGED_IN) et la connexion n'est pas encore considérée comme en échec) {
            Build-And-Send-SSnack(Connexion);
        }
    }
}
```

D.3.3 Algorithmes de cible

```
Handle-Status-SNACK-request(Connexion, PDU courante)
{
    si (ErrorRecoveryLevel operationnel > 0) {
        si (demande pour un cours acquitté) {
            Build-And-Send-Reject(Connection, PDU courante, Erreur de protocole);
        } autrement si (demande pour un cours non transmis) {
            éliminer, retour;
        } autrement { Retransmit-Status-Burst( PDU courante, TCB); }
    } autrement {
        Build-And-Send-Async(Connection, DroppedConnection, DefaultTime2Wait, DefaultTime2Retain);
    }
}
```

D.4 Algorithmes de récupération de connexion

D.4.1 Descriptions de procédure

```
Build-And-Send-Async(connexion de transport, code de cause, temps minimum, temps maximum);
Pick-A-Logged-In-Connection(session);
Build-And-Send-Logout(connexion de transport, identifiant de désétablissement de connexion, code de cause);
PerformImplicitLogout(connexion de transport, identifiant de désétablissement de connexion, informations de cible);
PerformLogin(connexion de transport, informations de cible);
CreateNewTransportConnection(informations de cible);
Build-And-Send-Command(connexion de transport, bloc de contrôle de tâche);
Connection-Cleanup-Handler(connexion de transport);
Connection-Resource-Timeout-Handler(connexion de transport);
Quiesce-And-Prepare-for-New-Allegiance(session, bloc de contrôle de tâche);
Build-And-Send-Logout-response(connexion de transport, CID de connexion en récupération, code de cause);
Build-And-Send-TaskMgmt-response(connexion de transport, PDU de commande de gestion de tâche, code de réponse);
Establish-New-Allegiance(bloc de contrôle de tâche, connexion de transport);
Schedule-Command-To-Continue(bloc de contrôle de tâche);
```

Note :

- Des conditions de transport d'exception comme une terminaison de connexion inattendue, une réinitialisation de connexion, et une connexion suspendue alors qu'elle est dans la phase de pleines caractéristiques, sont toutes supposées être signalées en asynchrone à la couche iSCSI en utilisant la procédure de Transport_Exception_Handler.

D.4.2 Algorithmes d'initiateur

```
Receive-an-In-PDU(Connexion, PDU courante)
{
   check-basic-validity( PDU courante);
   si (Header-Digest-Bad) éliminer, retour;
   Restituer le TCB à partir de CurrentPDU.InitiatorTaskTag.
   si (CurrentPDU.type == Async) {
      si (CurrentPDU.AsyncEvent == ConnectionDropped) {
         Restituer la connexion affectée pour CurrentPDU.Parameter1.
         AffectedConnection.CurrentTimeout = CurrentPDU.Parameter3;
        AffectedConnection.State = CLEANUP_WAIT;
        Start-Timer(Connection-Cleanup-Handler, AffectedConnection, CurrentPDU.Parameter2);
      } autrement si (CurrentPDU.AsyncEvent == LogoutRequest)) {
        AffectedConnection = Connexion;
        AffectedConnection.State = LOGOUT_REQUESTED;
        AffectedConnection.PerformConnectionCleanup = VRAI;
              AffectedConnection.CurrentTimeout = CurrentPDU.Parameter3;
        Start-Timer(Connection-Cleanup-Handler, AffectedConnection, 0);
      } autrement si (CurrentPDU.AsyncEvent == SessionDropped)) {
        pour (chaque connexion) {
           Connection.State = CLEANUP_WAIT;
           Connection.CurrentTimeout = CurrentPDU.Parameter3;
           Start-Timer(Connection-Cleanup-Handler, Connection, CurrentPDU.Parameter2);
        }
        Session.state = FAILED;
      }
   } autrement si (CurrentPDU.type == LogoutResponse) {
      Restituer le CleanupConnection pour CurrentPDU.CID.
      si (CurrentPDU.response = échec) {
        CleanupConnection.State = CLEANUP_WAIT;
      } autrement {
        CleanupConnection.State = FREE;
      }
   } autrement si (CurrentPDU.type == LoginResponse) {
       si (c'est une réponse à un désétablissement implicite) {
          Restituer le CleanupConnection.
          si (réussite) {
             CleanupConnection.State = FREE;
             Connection.State = LOGGED_IN;
          } autrement {
              CleanupConnection.State = CLEANUP_WAIT;
              DestroyTransportConnection(Connection);
          }
       }
   } autrement { /* le reste est sans relation avec la récupération de connexion et n'est pas montré */
   }
   si (CleanupConnection.State == FREE) {
     pour (chaque commande qui était active sur CleanupConnection) {
     /* Établir une nouvelle allégeance de connexion */
        NewConnection = Pick-A-Logged-In-Connection(Session);
        Build-And-Send-Command(NewConnection, TCB);
     }
   }
}

Connection-Cleanup-Handler(Connection)
{
```

```
   Restituer la session à partir de la connexion.
   si (la connexion peut encore échanger des PDU iSCSI) {
      NewConnection = Connexion;
   } autrement {
      Start-Timer(Connection-Resource-Timeout-Handler, Connection, Connection.CurrentTimeout);
      si (il y a d'autres connexions établies) {
          NewConnection = Pick-A-Logged-In-Connection(Session);
      } autrement {
          NewConnection = CreateTransportConnection(Session.OtherEndInfo);
          Initier un désétablissement implicite sur NewConnection pour Connection.CID.
          retour;
      }
   }
   Build-And-Send-Logout(NewConnection, Connection.CID, RecoveryRemove);
}

Transport_Exception_Handler(Connection)
{
   Connection.PerformConnectionCleanup = VRAI;
   si (l'événement est une déconnexion inattendue du transport) {
      Connection.State = CLEANUP_WAIT;
      Connection.CurrentTimeout = DefaultTime2Retain;
      Start-Timer(Connection-Cleanup-Handler, Connection, DefaultTime2Wait);
   } autrement {
      Connection.State = FREE;
   }
}
```

D.4.3 Algorithmes de cibles

```
Receive-an-In-PDU(Connexion, PDU courante)
{
   check-basic-validity(CurrentPDU);
   si (Header-Digest-Bad) éliminer, retour;
   autrement si (Data-Digest-Bad) {
          Build-And-Send-Reject(Connexion, PDU courante, Payload-Digest-Error);
          éliminer, retour;
   }
   Restituer le TCB et la session.
   si (CurrentPDU.type == Logout) {
     si (CurrentPDU.ReasonCode = RecoveryRemove) {
        Retrieve the CleanupConnection from CurrentPDU.CID).
        pour (chaque commande active sur CleanupConnection) {
            Quiesce-And-Prepare-for-New-Allegiance(Session, TCB);
            TCB.CurrentlyAllegiant = FAUX;
        }
        Cleanup-Connection-State(CleanupConnection);
        si ((mise au repos réussie) et (nettoyage réussi))
{
            Build-And-Send-Logout-response(Connection, CleanupConnection.CID, Succès);
        } autrement {
            Build-And-Send-Logout-réponse(Connection, CleanupConnection.CID, Échec);
        }
     }
   } autrement si ((CurrentPDU.type == Login) et ErrorRecoveryLevel opérationnel == 2) {
         Restituer la CleanupConnection à partire de CurrentPDU.CID).
         pour (chaque commande active sur CleanupConnection) {
             Quiesce-And-Prepare-for-New-Allegiance(Session, TCB);
             TCB.CurrentlyAllegiant = FAUX;
         }
         Cleanup-Connection-State(CleanupConnection);
         si ((mise au repos réussie) et (nettoyage réussi))
```

```
{
            Continuer avec le reste du traitement d'établissement;
        } autrement {
            Build-And-Send-Login-réponse(Connection, CleanupConnection.CID, Erreur de cible);
        }
    }
  } autrement si (CurrentPDU.type == TaskManagement) {
      si (CurrentPDU.function == "TaskReassign") {
          si (Session.ErrorRecoveryLevel < 2) {
              Build-And-Send-TaskMgmt-réponse(Connexion, PDU courante,
                  "Réallocation d'allégeance de tâche non prise en charge");
          } autrement si (tâche non trouvée) {
              Build-And-Send-TaskMgmt-response(Connexion, PDU courante, "Tâche pas dans l'ensemble de tâches");
          } autrement si (tâche actuellement allégeante) {
              Build-And-Send-TaskMgmt-réponse(Connexion, PDU courante, "Tâche toujours allégeante");
          } autrement {
              Establish-New-Allegiance(TCB, Connexion);
              TCB.CurrentlyAllegiant = VRAI;
              Schedule-Command-To-Continue(TCB);
          }
      }
  } autrement { /* Le reste est sans relation avec la récupération de connexion, et n'est pas montré */
  }
}

Transport_Exception_Handler(Connection)
{
  Connection.PerformConnectionCleanup = VRAI;
  si (l'événement est une déconnexion inattendue de transport) {
      Connection.State = CLEANUP_WAIT;
      Start-Timer(Connection-Resource-Timeout-Handler,
          Connection, (DefaultTime2Wait+DefaultTime2Retain));
      si (cette session a encore des connexions en phase de pleines caractéristiques) {
          DifferentConnection = Pick-A-Logged-In-Connection(Session);
          Build-And-Send-Async(DifferentConnection, DroppedConnection, DefaultTime2Wait, DefaultTime2Retain);
      }
  } autrement {
      Connection.State = FREE;
  }
}
```

Appendice E Effets du nettoyage de divers événements sur les cibles

E.1 Effets du nettoyage sur les objets iSCSI

Les tableaux qui suivent décrivent le comportement de la cible à réception des événements spécifiés dans les rangées du tableau. Le second tableau est une extension du premier et définit les actions de nettoyage pour plus d'objets sur les mêmes événements. La légende est :

Oui : (nettoyé/éliminé/réinitialisé sur l'événement spécifié dans la rangée). Sauf noté autrement, l'action de nettoyage n'est applicable que pour l'accès de l'initiateur producteur.

Non : (non affecté sur l'événement spécifié dans la rangée, c'est-à-dire, reste à la valeur précédente).

NA = Non applicable ou non défini.

	T (1)	**IC (2)**	**CT (5)**	**ST (6)**	**PP (7)**
Échec de connexion (8)	Oui	Oui	Non	Non	Oui
Fin de temporisation d'état de connexion (9)	NA	NA	Oui	Non	NA
Fin de temporisation/clôture/réinstallation de session (10)	Oui	Oui	Oui	Oui	Oui (14)

Continuation de session (12)	NA	NA	Non (11)	Non	NA
Désétablissement réusssi de connexion	Oui	Oui	Oui	Non	Oui (13)
Échec de session (18)	Oui	Oui	Non	Non	Oui
Désétablissement de récupération réusssi	Oui	Oui	Non	Non	Oui (13)
Échec de désétablissement	Oui	Oui	Non	Non	Oui
Établissement de connexion (de tête)	NA	NA	NA	Oui (15)	NA
Établissement de connexion (non de tête)	NA	NA	Non (11)	Non	Oui
Réinitialisation de cible à froid (16)	Oui (20)	Oui	Oui	Oui	Oui
Réinitialisation de cible à chaud (16)	Oui (20)	Oui	Oui	Oui	Oui
Réinitialisation de LU (19)	Oui (20)	Oui	Oui	Oui	Oui
Cycle d'alimentation (16)	Oui	Oui	Oui	Oui	Oui

(1) Les TTT incomplètes (IT) sont des étiquettes de transfert de cible sur lesquelles la cible attend encore la réception de PDU. Les exemples incluent des TTT reçues via R2T, NOP-In, etc.
(2 Les commandes immédiates (IC) attendent pour leur exécution sur une cible (par exemple, ABORT TASK SET).
(5) Les tâches de connexion (CT) sont actives sur la connexion iSCSI en question.
(6) Les tâches de session sont actives sur la session iSCSI entière ; une union de "tâches de connexion" sur toutes les connexions participantes.
(7) Les PDU partielles (PP) (s'il en est) sont des PDU qui sont partiellement envoyées et attendent un crédit de fenêtre de transport pour achever la transmission.
(8) L'échec de connexion est une condition d'exception : une des connexions de transport a fermé, des connexions de transport sont réinitialisées, ou des connexions de transport sont arrivées en fin de temporisation, ce qui a terminé de façon abrupte la phase de pleines caractéristiques iSCSI de la connexion. Une défaillance de connexion met toujours l'automate à états de la connexion dans l'état CLEANUP_WAIT *(attente de nettoyage)*.
(9) La fin de temporisation d'état de connexion se produit si une connexion passe plus de temps que prévu durant la négociation d'établissement dans l'état CLEANUP_WAIT, et cela amène la connexion à l'état FREE (transition M1 dans le diagramme d'état de nettoyage de connexion ; paragraphe 8.2).
(10) Fin de temporisation de session, clôture, et réinstallation sont définies au paragraphe 6.3.5.
(11) Cet effet de nettoyage n'est "Oui" que si c'est une réinstallation de connexion et si le niveau de récupération d'erreur opérationnel est inférieur à 2.
(12) La continuation de session est définie au paragraphe 6.3.6.
(13) Cet effet de nettoyage n'est valide que si la connexion est en cours de désétablissement sur une connexion différente et quand la connexion en cours de désétablissement sur la cible peut avoir des PDU partielles en instance d'envoi. Dans tous les autres cas, l'effet est "NA".
(14) Cet effet de nettoyage n'est valide que pour un désétablissement "clore la session" dans une session multi connexions. Dans tous les autres cas, l'effet est "NA".
(15) Applicable seulement si cet établissement de connexion de tête est une réinstallation de session. Si ce n'est pas le cas, c'est "NA".
(16) Cette opération affecte tous les initiateurs établis.
(18) Défaillance de session est défini au paragraphe 6.3.6.
(19) Cette opération affecte tous les initiateurs établis, et les effets du nettoyage ne sont applicables qu'à la LU en cours de réinitialisation,
(20) Avec la sémantique standard d'interruption multi tâches (paragraphe 4.2.3.3), une réinitialisation de cible à chaud ou une réinitialisation de cible à froid ou une réinitialisation de LU va nettoyer les TTT actives à son achèvement. Cependant, la sémantique FastAbort multi tâches définie au paragraphe 4.2.3.4 ne garantit pas que les TTT actives soient nettoyées par la fin des opérations de réinitialisation. En fait, la sémantique de FastAbort a été conçue pour permettre le nettoyage des TTT d'une façon "paresseuse" après la livraison de la réponse de TMF. Donc, quand TaskReporting=FastAbort (paragraphe 13.23) est opérationnel sur une session, les effets de nettoyage des opérations de réinitialisation sur les "TTT incomplètes" est "Non".

	DC (1)	**DD (2)**	**SS (3)**	**CS (4)**	**DS (5)**
Échec de connexion	Non	Oui	Non	Non	Non
Fin de temporisation d'état de connexion	Oui	NA	Oui	Non	NA
Fin de temporisation/clôture/réinstallation de session	Oui	Oui	Oui (7)	Oui	NA
Continuation de session	Non (11)	NA (12)	NA	Non	NA (13)

Désétablissement réusssi de connexion	Oui	Oui	Oui	Non	NA
Échec de session	Non	Oui	Non	Non	Non
Désétablissement de récupération réusssi	Oui	Oui	Oui	Non	Non
Échec de désétablissement	Non	Oui (9)	Non	Non	Non
Établissement de connexion (de tête)	NA	NA	Non (8)	Non (8)	NA
Établissement de connexion (non de tête)	Non (11)	NA (12)	Non (8)	Non	NA (13)
Réinitialisation de cible à froid	Oui	Oui	Oui	Oui (10)	NA
Réinitialisation de cible à chaud	Oui	Oui	Non	Non	NA
Réinitialisation de LU	Non	Oui	Non	Non	Non
Cycle d'alimentation	Oui	Oui	Oui	Oui (10)	NA

(1) Les commandes discontiguës (DC) sont des commandes allégeantes à la connexion en question et qui attendent d'être réordonnées dans la couche iSCSI. Tous les "Oui de cette colonne supposent que la tâche qui cause l'événement (si bien sûr l'événement est le résultat d'une tâche) est produite comme commande immédiate, parce que les discontiguïtés peuvent être en avant de la tâche.
(2) Les données discontiguës (DD) sont des PDU de données reçues pour la tâche en question et qui attendent d'être réordonnées du fait de discontiguïtés antérieures du numéro de séquence de données,
(3) "SS" se réfère au numéro de séquence d'état.
(4) "CS" se réfère au numéro de séquence de commande.
(5) "DS" se réfère au numéro de séquence de données,
(7) Cette action nettoie le numéro de séquence d'état sur toutes les connexions.
(8) Ce numéro de séquence est instancié sur cet événement.
(9) Une défaillance de désétablissement conduit l'automate à états de connexion à l'état CLEANUP_WAIT, comme le fait l'événement de défaillance de connexion. Donc, il a un effet similaire sur cet aspect et plusieurs autres du protocole.
(10) Ceci est nettoyé en vertu du fait que toutes les sessions avec tous les initiateurs sont terminées.
(11) Cet effet de nettoyage est "Oui" si c'est une réinstallation de connexion.
(12) Cet effet de nettoyage n'est "Oui" que si c'est une réinstallation de connexion et si le niveau de récupération d'erreur opérationnel est 2.
(13) Cet effet de nettoyage n'est "Non" que si c'est une réinstallation de connexion et si le niveau de récupération d'erreur opérationnel est 2.

E.2 Effets du nettoyage sur les objets SCSI

La seule action de protocole iSCSI qui puisse effectuer des actions de nettoyage sur les objets SCSI est la notification "Perte de nexus I_T" (paragraphe 6.3.5.1 ("Notification de perte de nexus")). [SPC3] décrit les effets de nettoyage de cette notification sur divers attributs SCSI. De plus, les normes SCSI (comme [SAM2] et [SBC2]) définissent des actions de nettoyage supplémentaires qui peuvent avoir lieu pour plusieurs objets SCSI sur des événements SCSI comme des réinitialisations de LU et des rétablissements d'alimentation.

Comme iSCSI définit une Réinitialisation de cible à froid comme un "équivalent de protocole" d'un cycle d'alimentation de cible, la Réinitialisation de cible à froid iSCSI doit aussi être considérée comme un événement de réinitialisation d'alimentation dans l'interprétation des actions définies dans les normes SCSI.

Lorsque la session iSCSI est reconstruite (entre les mêmes accès SCSI avec le même identifiant de nexus) rétablissant le même nexus I_T, tous les objets SCSI qui sont définis comme n'étant pas nettoyés en présence de l'événement de notification "Perte de nexus I_T", comme des réservations persistentes, sont automatiquement associés à cette nouvelle session.

Remerciements

Plusieurs personnes du groupe de travail IPS d'origine ont apporté des contributions significatives aux RFC d'origine, 3720, 3980, 4850, et 5048. Précisément, les auteurs des RFC d'origine – qui sont consolidées ici dans un seul document – étaient les suivants :
RFC 3720 : Julian Satran, Kalman Meth, Costa Sapuntzakis, Mallikarjun Chadalapaka, Efri Zeidner
RFC 3980 : Marjorie Krueger, Mallikarjun Chadalapaka, Rob Elliott
RFC 4850 : David Wysochanski
RFC 5048 : Mallikarjun Chadalapaka

Tous nos remerciements à Fred Knight pour sa contribution à la notation UML et aux dessins de ce document.

Nous tenons de plus à remercier les personnes suivantes qui ont contribué à ce document révisé : David Harrington, Paul Koning, Mark Edwards, Rob Elliott, et Martin Stiemerling. Merci à Yi Zeng et Nico Williams pour la suggestion et/ou la révision du texte des considérations sur la sécurité relatives à Kerberos.

Les auteurs remercient chaleureusement de leurs retours durant le processus de dernier appel à révision un certain nombre de personnes dont les retours ont significativement amélioré le présent document. Ces personnes sont Stephen Farrell, Brian Haberman, Barry Leiba, Pete Resnick, Sean Turner, Alexey Melnikov, Kathleen Moriarty, Fred Knight, Mike Christie, Qiang Wang, Shiv Rajpal, et Andy Banta.

Finalement, le présent document a aussi bénéficié de contributions et révisions significatives du groupe de travail Storm dans son ensemble.

Les commentaires peuvent être envoyés à Mallikarjun Chadalapaka.

Adresse des auteurs

Mallikarjun Chadalapaka
Microsoft
One Microsoft Way
Redmond, WA 98052
USA
mél : cbm@chadalapaka.com

Kalman Meth
IBM Haifa Research Lab
Haifa University Campus - Mount Carmel
Haifa 31905, Israel
téléphone +972.4.829.6341
mél : meth@il.ibm.com

David L. Black
EMC Corporation
176 South St.
Hopkinton, MA 01748
USA
téléphone +1 (508) 293-7953
mél : david.black@emc.com

Julian Satran
Infinidat Ltd.
mél : julians@infinidat.com , julian@satran.net