Groupe de travail Réseau
Request for Comments : 4975
Catégorie : sur la voie de la normalisation
Traduction Claude Brière de L'Isle

B. Campbell, éd., Estacado Systems
R. Mahy, éd., Plantronics
C. Jennings, éd., Cisco Systems, Inc.
septembre 2007

Protocole de relais de session de message (MSRP)

Résumé
Le présent document décrit le protocole de relais de session de message, un protocole pour transmettre une série de messages instantanés en rapports dans le contexte d'une session. Les sessions de messages sont traitées comme tout autre flux de supports quand ils sont établis via un protocole de rendezvous ou de création de sessions comme le protocole d'intitalisation de session (SIP, *Session Initiation Protocol*).

Table des Matières

1. Introduction....2
2. Conventions....3
3. Applicabilité de MSRP....3
4. Vue d'ensemble du protocole....4
5. Concepts clés....5
5.1 Tramage MSRP et mise en tronçons de message....5
5.2 Adressage MSRP....6
5.3 Transaction et modèle de rapport MSRP....6
5.4 Modèle de connexion MSRP....7
6. URI MSRP....8
6.1 Comparaison d'URI MSRP....9
6.2 Résolution de l'appareil hôte MSRP....9
7. Comportement spécifique de la méthode....10
7.1 Construction des demandes....10
7.2 Construction des réponses....14
7.3 Réception des demandes....14
8. Utilisation de MSRP avec SIP et SDP....16
8.1 Connexion et lignes de supports SDP....16
8.2 Négociations d'URI....17
8.3 Attributs de chemin avec plusieurs URI....17
8.4 Mise à jour des offres SDP....18
8.5 Négociation de connexion....18
8.6 Négociation de type de contenu....18
8.7 Exemple d'échange SDP....20
8.8 Expérience d'utilisateur MSRP avec SIP....20
8.9 Attribut Direction SDP et MSRP....21
9. Syntaxe formelle....21
10. Description des codes de réponse....22
10.1 200....23
10.2 400....23
10.3 403....23
10.4 408....23
10.5 413....23
10.6 415....23

10.7 423....23
10.8 481....23
10.9 501....23
10.10 506....23
11. Exemples....24
11.1 Session IM de base....24
11.2 Message avec contenu XHTML....25
11.3 Message tronçonné....26
11.4 Message tronçonné avec charge utile Message/CPIM....26
11.5 Message System....26
11.6 Rapport positif....27
11.7 IM fourché....27
12. Extensibilité....29
13. Compatibilité CPIM....29
14. Considérations sur la sécurité....29
14.1 Secret de l'URI MSRP....30
14.2 Protection au niveau du transport....30
14.3 S/MIME....31
14.4 Utilisation de TLS en mode d'homologue à homologue....31
14.5 Autres problèmes de sécurité....32
15. Considérations relatives à l'IANA....33
15.1 Nom des méthodes MSRP....33
15.2 Champs d'en-tête MSRP....33
15.3 Codes d'état MSRP....34
15.4 Accès MSRP....34
15.5 Schéma d'URI....34
15.6 Protocole de transport SDP....35
15.7 Noms d'attribut SDP....35
16. Contributeurs et remerciements....36
17. Références....36
17.1 Références normatives....36
17.2 Références pour information....37
Adresse des auteurs....38
Déclaration complète de droits de reproduction....38

1. Introduction

Une série de messages instantanés liés entre deux ou plusieurs parties peut être considérée comme faisant partie d'une "session de messages", c'est-à-dire un échange conversationnel de messages avec un début et une fin définis. Cela contraste avec les messages individuels envoyés indépendamment les uns des autres. Les systèmes de messagerie qui ne suivent que les messages individuels peuvent être décrits comme une messagerie en "mode page", tandis que la messagerie qui fait partie d'une "session" avec un début et une fin définis est appelée messagerie en "mode session".

La messagerie en mode page est activée dans le protocole SIP [RFC3261] via la méthode SIP MESSAGE [RFC3428]. La messagerie en mode session présente toutefois un certain nombre d'avantages par rapport à la messagerie en mode page, tels qu'un rendez-vous explicite, une intégration plus étroite avec d'autres types de supports, un fonctionnement direct de client à client, ainsi qu'une confidentialité et une sécurité accrues.

Le présent document définit un protocole de transport de messages instantanés en mode session, appelé protocole de relais de session de messages (MSRP, *Message Session Relay Protocol*) dont les sessions peuvent être négociées avec une offre ou une réponse [RFC3264] en utilisant le protocole de description de session (SDP, *Session Description Protocol*) [RFC4566]. L'échange est assuré par un protocole de signalisation, tel que SIP [RFC3261]. Cela permet à un agent d'utilisateur de communication de proposer une session de messagerie comme l'un des types de supports possibles dans une session. Par exemple, Alice peut vouloir communiquer avec Bob. Alice ne sait pas pour l'instant si Bob a son téléphone ou son client de messagerie instantanée à portée de main, mais elle est prête à utiliser l'un ou l'autre. Elle envoie une invitation à une session à l'adresse qu'elle a enregistrée pour Bob, sip:bob@exemple.com. Son invitation propose à la fois une session vocale et une session de messagerie instantanée. Les services SIP à exemple.com transmettent l'invitation à Bob par l'intermédiaire de ses clients actuellement enregistrés. Bob accepte l'invitation sur son client de messagerie instantanée et ils entament une conversation suivie.

Lorsqu'un utilisateur utilise un URL de messagerie instantanée (IM, *Instant Messaging*), la [RFC3861] définit la manière dont le DNS peut être utilisé pour transposer cela en un protocole particulier afin d'établir la session comme une session SIP. SIP peut utiliser un modèle offre/réponse pour transporter les URI MSRP pour les supports dans SDP. Le présent document définit comment l'échange offre/réponse fonctionne pour établir des connexions MSRP et comment les messages sont envoyés à travers le MSRP, mais il ne traite pas des questions de transposition d'un URL IM avec un protocole d'établissement de session.

Ce modèle de session permet d'intégrer des sessions de messages dans des applications de communication avancées avec peu ou pas de développement de protocole supplémentaire. Par exemple, au cours de la session de discussion ci-dessus, Bob décide qu'Alice a vraiment besoin de parler à Carol. Bob peut transférer [RFC5589] Alice à Carol, les introduisant ainsi dans leur propre session de messagerie. Les sessions de messagerie peuvent ensuite être facilement intégrées dans les centres d'appel et les environnements de répartition à l'aide d'applications tierces de contrôle d'appel [RFC3725] et de conférence [RFC4579].

Le présent document spécifie le comportement de MSRP uniquement pour les sessions d'homologue à homologue, c'est-à-dire les sessions qui ne franchissent qu'un seul bond. Les appareils de relais MSRP [RFC4976] (appelés ici "relais") sont spécifiés dans un document distinct. Un point d'extrémité qui met en œuvre la présente spécification, mais pas la spécification relative aux relais, ne sera pas en mesure d'introduire des relais dans le chemin du message, mais pourra toujours interopérer avec des homologues qui utilisent des relais.

2. Conventions

Les mots clés "DOIT", "NE DOIT PAS", "EXIGE", "DEVRA", "NE DEVRA PAS", "DEVRAIT", "NE DEVRAIT PAS", "RECOMMANDE", "PEUT", et "FACULTATIF" en majuscules dans ce document sont à interpréter comme décrit dans le BCP 14, [RFC2119].

Le présent document se réfère constamment à un "message" comme à une unité complète de contenu de texte ou MIME. Dans certains cas, un message est partagé et livré en plus d'une demande MSRP. Chacune de ces portions du message complet est appelée un "tronçon".

3. Applicabilité de MSRP

MSRP n'est pas conçu pour être utilisé en tant que protocole autonome. MSRP ne DOIT être utilisé que dans le cadre d'un mécanisme de rendez-vous saisfaisant aux exigences suivantes :

- Le mécanisme de rendez-vous DOIT fournir les deux URI MSRP associés à une session MSRP à chacun des points d'extrémité participants. Le mécanisme de rendez-vous DOIT mettre en œuvre des mécanismes pour protéger la confidentialité de ces URI -- ils NE DOIVENT PAS être mis à la disposition d'un tiers qui n'est pas de confiance ou être facilement découvrables.
- Le mécanisme de rendez-vous DOIT fournir des mécanismes pour la négociation de toutes les extensions MSRP prises en charge qui ne sont pas rétrocompatibles.
- Le mécanisme de rendez-vous DOIT être capable de transporter nativement les URI im: ou de traduire automatiquement les URI im: [RFC3860] en les identifiants d'adressage du protocole de rendez-vous.

Pour utiliser un mécanisme de rendez-vous avec MSRP, une RFC DOIT être préparée pour décrire comment échanger les URI MSRP et répondre aux exigences énumérées ici. Le présent document fournit une telle description pour l'utilisation de MSRP dans le contexte de SIP et SDP.

SIP satisfait à ces exigences pour un mécanisme de rendez-vous. Les URI MSRP sont échangés à l'aide de SDP dans le cadre d'un échange offre/réponse via SIP.

Le SDP échangé peut également être utilisé pour négocier des extensions à MSRP. Ce SDP peut être sécurisé à l'aide de n'importe quel mécanisme disponible dans SIP, y compris le mécanisme sips pour assurer la sécurité du transport à travers les intermédiaires et les extensions de messagerie électronique Internet sécurisée/multi objets (S/MIME, *Secure/Multipurpose Internet Mail Extensions*) pour la protection de bout en bout du corps du SDP. SIP peut porter des URI arbitraires (y compris des URI im:) dans l'URI de demande, et des procédures sont disponibles pour transposer les URI im: en URI sip: ou sips:. Il est prévu que les premiers déploiements de MSRP utilisent SIP comme mécanisme de rendez-vous.

4. Vue d'ensemble du protocole

MSRP est un protocole fondé sur le texte, en mode connexion, pour l'échange de contenu MIME [RFC2045] arbitraire (binaire) en particulier les messages instantanés. Cette section est un aperçu non normatif du fonctionnement de MSRP et de son utilisation avec SIP.

Les sessions MSRP sont normalement organisées à l'aide de SIP de la même manière qu'est établie une session de supports audio ou vidéo. Un agent d'utilisateur SIP (Alice) envoie à l'autre (Bob) une invitation SIP contenant une description de session offerte qui inclut une session MSRP. L'agent d'utilisateur SIP receveur peut accepter l'invitation et inclure une description de session de réponse qui accepte le choix du support. La description de session d'Alice contient un URI MSRP qui décrit où elle est prête à recevoir des demandes MSRP de Bob, et vice versa. (Note : certaines lignes des exemples ont été supprimées pour des raisons de clarté et de concision).

Alice envoie à Bob :

```
INVITE sip:bob@biloxi.exemple.com SIP/2.0
To: <sip:bob@biloxi.exemple.com>
From: <sip:alice@atlanta.exemple.com>;tag=786
Call-ID: 3413an89KU
Content-Type: application/sdp

c=IN IP4 atlanta.exemple.com
m=message 7654 TCP/MSRP *
a=accept-types:text/plain
a=path:msrp://atlanta.exemple.com:7654/jshA7weztas;tcp
```

Bob envoie à Alice :

```
SIP/2.0 200 OK
To: <sip:bob@biloxi.exemple.com>;tag=087js
From: <sip:alice@atlanta.exemple.com>;tag=786
Call-ID: 3413an89KU
Content-Type: application/sdp

c=IN IP4 biloxi.exemple.com
m=message 12763 TCP/MSRP *
a=accept-types:text/plain
a=path:msrp://biloxi.exemple.com:12763/kjhd37s2s20w2a;tcp
```

Alice envoie à Bob :

```
ACK sip:bob@biloxi SIP/2.0
To: <sip:bob@biloxi.exemple.com>;tag=087js
From: <sip:alice@atlanta.exemple.com>;tag=786
Call-ID: 3413an89KU
```

Figure 1 : Établissement de session

MSRP définit deux types de demandes, ou méthodes. Les demandes SEND sont utilisées pour livrer un message complet ou un tronçon (une partie d'un message complet) tandis que les demandes REPORT rendent compte de l'état d'un message précédemment envoyé, ou d'une gamme d'octets à l'intérieur d'un message. Lorsqu'Alice reçoit la réponse de Bob, elle vérifie qu'elle dispose déjà d'une connexion avec Bob. Si ce n'est pas le cas, elle ouvre une nouvelle connexion avec Bob en utilisant l'URI qu'il a fourni dans le SDP. Alice envoie ensuite une demande SEND à Bob avec son message initial, et Bob répond en indiquant que la demande d'Alice a été reçue avec succès.

```
MSRP a786hjs2 SEND
To-Path: msrp://biloxi.exemple.com:12763/kjhd37s2s20w2a;tcp
From-Path: msrp://atlanta.exemple.com:7654/jshA7weztas;tcp
Message-ID: 87652491
Byte-Range: 1-25/25
```

```
Content-Type: text/plain

Hé Bob, es-tu là ?
  -------a786hjs2$

MSRP a786hjs2 200 OK
To-Path: msrp://atlanta.exemple.com:7654/jshA7weztas;tcp
From-Path: msrp://biloxi.exemple.com:12763/kjhd37s2s20w2a;tcp
  -------a786hjs2$
```

Figure 2 : Exemple d'échange MSRP

La demande d'Alice commence par la ligne de début du MSRP, qui contient un identifiant de transaction également utilisé pour encadrer la demande. Elle inclut ensuite le chemin des URI vers la destination dans le champ d'en-tête To-Path, et son propre URI dans le champ d'en-tête From-Path. Dans ce cas typique, il n'y a qu'un seul "bond", et donc un seul URI dans chaque champ d'en-tête de chemin. Elle inclut également un identifiant de message, qu'elle peut utiliser pour établir une corrélation entre les rapports d'état et le message original. Ensuite, elle met le contenu réel. Enfin, elle clôt la demande par une ligne de fin composée de sept traits d'union, de l'identifiant de la transaction et d'un "$" pour indiquer que cette requête contient la fin d'un message complet.

Si Alice souhaite transmettre un message très volumineux, elle peut le diviser en tronçons et transmettre chaque tronçon dans une demande SEND distincte. L'identifiant du message correspond à l'ensemble du message, de sorte que le receveur peut également l'utiliser pour réassembler le message et déterminer quels tronçons appartiennent à quel message. Le découpage en tronçons est décrit plus en détail au paragraphe 5.1. Le champ d'en-tête Byte-Range identifie la partie du message transportée dans ce tronçon et la taille totale du message.

Alice peut aussi préciser le type de rapport qu'elle souhaite obtenir en réponse à sa demande. Si Alice demande des accusés de réception positifs, Bob envoie une demande REPORT à Alice confirmant la livraison de son message complet. Cette fonction est particulièrement utile si Alice a envoyé une série de demandes SEND contenant des tronçons d'un seul message. Le paragraphe 5.3 donne plus de détails sur les types de rapports et d'erreurs.

Alice et Bob choisissent leurs URI MSRP de telle sorte qu'il soit difficile de deviner l'URI exact. Alice et Bob peuvent rejeter les demandes adressées à des URI qu'ils ne s'attendent pas à recevoir et peuvent établir une corrélation entre l'URI spécifique et l'envoyeur probable. Alice et Bob peuvent aussi utiliser TLS [RFC4346] pour assurer la sécurité du canal sur ce bond. Pour recevoir des demandes MSRP sur une connexion protégée par TLS, Alice ou Bob peuvent annoncer des URI avec le schéma "msrps" au lieu de "msrp".

MSRP est conçu de manière à pouvoir transporter des URI pour des nœuds situés à l'autre bout des relais. Pour cette raison, un URI avec le schéma "msrps" ne fait aucune hypothèse sur les propriétés de sécurité des autres bonds, mais seulement sur le bond suivant. L'agent d'utilisateur connaît l'URI de chaque bond et peut donc vérifier que chaque URI possède les propriétés de sécurité souhaitées. Les URI MSRP sont examinés plus en détail à la Section 6.

Une paire adjacente de nœuds MSRP occupés (par exemple, deux relais) peut facilement avoir plusieurs sessions et échanger du trafic pour plusieurs utilisateurs simultanément. Les nœuds peuvent utiliser les connexions existantes pour porter du nouveau trafic avec le même hôte de destination, le même accès, le même protocole de transport et le même schéma. Les nœuds MSRP peuvent retracer le nombre de sessions qui utilisent une connexion particulière et fermer ces connexions lorsqu'aucune session ne les a utilisées pendant un certain temps. La gestion des connexions est abordée plus en détail au paragraphe 5.4.

5. Concepts clés

5.1 Tramage MSRP et mise en tronçons de message

Les messages envoyés à l'aide de MSRP peuvent être très volumineux et être transmis en plusieurs demandes SEND, chaque demande SEND contenant un tronçon du message global. Les longs tronçons peuvent être interrompus en cours de transmission afin de garantir l'équité entre les connexions de transport partagées. Pour ce faire, MSRP utilise un mécanisme de tramage fondé sur les limites. La première ligne d'une demande MSRP contient un identifiant unique qui est aussi utilisé pour indiquer la fin de la demande. À la fin de la ligne de fin, un fanion indique s'il s'agit du dernier tronçon de données de ce message ou si le message se poursuivra dans un tronçon ultérieur. La demande contient aussi un champ d'en-tête Byte-Range qui indique la position globale de ce tronçon dans le message complet.

Par exemple, l'extrait suivant de deux demandes SEND montre qu'un message contenant le texte "abcdEFGH" est envoyé en deux tronçons.

```
MSRP dkei38sd SEND
Message-ID: 4564dpWd
Byte-Range: 1-*/8
Content-Type: text/plain

abcd
-------dkei38sd+

MSRP dkei38ia SEND
Message-ID: 4564dpWd
Byte-Range: 5-8/8
Content-Type: text/plain

EFGH
-------dkei38ia$
```

Figure 3 : Coupure d'un message en tronçons

Ce mécanisme de découpage permet à l'envoyeur d'interrompre un tronçon à mi-chemin de son envoi. La possibilité d'interrompre les messages permet à plusieurs sessions de partager une connexion TCP et aux gros messages d'être envoyés efficacement sans bloquer les autres messages qui partagent la même connexion, voire la même session MSRP. Tout tronçon de plus de 2048 octets DOIT pouvoir être interrompu. MSRP serait plus simple à mettre en œuvre si chaque session MSRP utilisait sa propre connexion TCP, mais il existe des raisons impérieuses de conserver les connexions. Par exemple, l'homologue TCP peut être un appareil de relais qui se connecte à de nombreux autres homologues. Un tel dispositif s'adaptera mieux si chaque homologue ne crée pas un grand nombre de connexions. (Noter que dans l'exemple ci-dessus, le tronçon initial était interruptible pour les besoins de l'exemple, même si sa taille est bien inférieure à la limite pour laquelle l'interruptibilité serait nécessaire).

Le mécanisme de découpage ne s'applique qu'à la méthode SEND, car c'est la seule méthode utilisée pour transférer le contenu de message.

5.2 Adressage MSRP

Les entités MSRP sont adressées à l'aide d'URI. Les schémas d'URI MSRP sont définis à la Section 6. La syntaxe des champs d'en-tête To-Path et From-Path autorise à chacun une liste d'URI. Cela a été fait pour permettre au protocole de fonctionner avec des relais, qui sont définis dans un document séparé, afin de fournir un chemin complet jusqu'au receveur final. Lorsque deux nœuds MSRP communiquent directement, ils n'ont besoin que d'un seul URI dans la liste To-Path et d'un seul URI dans la liste From-Path.

5.3 Transaction et modèle de rapport MSRP

Un émetteur envoie des demandes MSRP à un receveur. Le receveur DOIT rapidement accepter ou rejeter la demande. Si le receveur a initialement accepté la demande, il peut encore faire des choses qui prennent beaucoup de temps pour réussir ou échouer. Par exemple, si le receveur est une passerelle MSRP vers le protocoele extensible de messagerie et de présence (XMPP, *Extensible Messaging and Presence Protocol*) [RFC3921], il peut transmettre le message via XMPP. Le côté XMPP peut indiquer ultérieurement que la demande n'a pas fonctionné. À ce stade, le receveur MSRP peut avoir besoin d'indiquer que la demande n'a pas abouti. Il y a ici deux concepts importants : premièrement, la livraison bond par bond de la demande peut réussir ou échouer ; deuxièmement, le résultat final de la demande peut être traité avec succès ou non. Le premier type d'état est appelé "état de transaction" et peut être renvoyé en réponse à une demande. Le second type d'état est appelé "état de livraison" et peut être retourné dans une transaction REPORT.

L'envoyeur original d'une demande peut indiquer s'il souhaite recevoir des rapports pour les demandes qui échouent, et peut indépendamment indiquer s'il souhaite recevoir des rapports pour les demandes qui réussissent. Un receveur n'envoie un RAPPORT de réussite que s'il sait que la demande a été livrée avec succès et que l'envoyeur a demandé un rapport de réussite. Un receveur n'envoie un RAPPORT d'échec que si la livraison de la demande a échoué et que l'expéditeur a demandé des rapports d'échec.

Le présent document décrit le comportement des points d'extrémité MSRP. Les relais MSRP introduiront des conditions

supplémentaires indiquant qu'un rapport d'échec devrait être envoyé, comme l'absence de réponse positive de la part du prochain bond.

Deux champs d'en-tête contrôlent le désir de l'envoyeur de recevoir des rapports. Le champ d'en-tête Success-Report peut avoir une valeur de "oui" ou "non" et le champ d'en-tête Failure-Report peut avoir les valeurs "oui", "non" ou "partiel".

Les combinaisons de rapports sont nécessaires pour répondre aux différents scénarios des systèmes de gestion de l'information actuellement déployés. Le rapport de réussite peut être "non" dans de nombreux systèmes publics afin de réduire la charge, mais il peut être "oui" dans certains systèmes d'entreprise, tels que les systèmes utilisés pour le commerce des valeurs mobilières. Une valeur de rapport d'échec de "non" est utile pour envoyer des messages système tels que "le système va tomber en panne dans 5 minutes" sans provoquer d'explosion de réponses à l'envoyeur. Un rapport d'échec de "oui" est utilisé par de nombreux systèmes qui souhaitent notifier à l'utilisateur l'échec du message. Un rapport d'échec "partiel" permet de signaler des erreurs autres que des dépassements de délai. Le signalement d'une erreur de temporisation nécessite que le bond d'envoi lance un temporisateur et que le bond receveur envoie un accusé de réception pour arrêter le temporisateur. Certains systèmes ne veulent pas de cette surcharge. L'option "partiel" leur permet de choisir de ne pas le faire, tout en autorisant l'envoi de réponses d'erreur dans de nombreux cas.

Le terme "partiel" indique que le mécanisme d'accusé de réception bond par bond qui serait nécessaire avec une valeur de "Failure-Report" de "oui" n'est pas invoqué. Ainsi, chaque appareil n'utilise qu'une "partie" de l'ensemble des outils de détection d'erreurs dont il dispose. Cela permet un compromis entre l'absence de signalement des défaillances et le signalement de toutes les défaillances possibles. Par exemple, avec "partiel", un appareil envoyeur n'a pas à conserver l'état de la transaction dans l'attente d'un accusé de réception positif. Mais cela permet quand même aux appareils de rapporter d'autres types d'erreurs. L'appareil receveur peut toujours rapporter une violation de la politique, comme un type de contenu inacceptable, ou une erreur ICMP en essayant de se connecter à un appareil en aval.

5.4 Modèle de connexion MSRP

Lorsqu'un point d'extrémité MSRP souhaite envoyer une demande à un homologue identifié par un URI MSRP, il a d'abord besoin d'une connexion de transport, avec les propriétés de sécurité appropriées, avec l'hôte spécifié dans l'URI. Si l'envoyeur a déjà une telle connexion, c'est-à-dire une connexion associée au même hôte, au même accès et au même schéma d'URI, il DEVRAIT réutiliser cette connexion.

Lorsqu'une nouvelle session MSRP est créée, le point d'extrémité initiateur DOIT agir en tant que point d'extrémité "actif", ce qui signifie qu'il est responsable de l'ouverture de la connexion de transport avec celui qui répond, si une nouvelle connexion est nécessaire. Cependant, cette exigence PEUT être assouplie si des mécanismes normalisés de négociation de la direction de la connexion sont disponibles et mis en œuvre par les deux parties à la connexion.

De même, le point d'extrémité actif DOIT immédiatement produire une demande SEND. Cette demande SEND initiale PEUT avoir un corps si l'envoyeur a un contenu à envoyer, ou PEUT ne pas avoir de corps du tout.

La première demande SEND sert à lier une connexion à une session MSRP du point de vue du point d'extrémité passif. Si la connexion n'est pas authentifiée par TLS et si le point d'extrémité actif n'a pas envoyé de demande immédiate, le point d'extrémité passif n'aurait aucun moyen de déterminer qui s'est connecté et ne pourrait envoyer en toute sécurité aucune demande à la partie active tant que celle-ci n'aura pas envoyé sa première demande.

Quand un élément a besoin de former une nouvelle connexion, il consulte l'URI pour déterminer le type de connexion (TLS, TCP, etc.) puis se connecte à l'hôte indiqué par l'URI, en suivant les règles de résolution d'URI énoncées au paragraphe 6.2. Les connexions qui utilisent le schéma "msrps" DOIVENT utiliser TLS. Le SubjectAltName dans le certificat reçu DOIT correspondre à la partie nom d'hôte de l'URI et le certificat DOIT être valide conformément à la [RFC3280], y compris d'avoir une date valide et d'être signé par une autorité de certification acceptable. À ce stade, l'appareil qui a initié la connexion peut supposer que cette connexion est avec l'hôte correct.

Les règles de correspondance des noms de certificats et de signature de l'autorité de certification PEUVENT être assouplies quand on utilise TLS d'homologue à homologue. Dans ce cas, un mécanisme permettant de s'assurer que l'homologue a utilisé un certificat correct DOIT être utilisé. Voir plus de détails au paragraphe 14.4.

Si la connexion utilise l'authentification TLS mutuelle et si le client TLS présente un certificat valide, l'élément qui accepte la connexion peut alors vérifier l'identité de l'appareil qui se connecte en comparant la partie Nom d'hôte de l'URI cible dans le SDP fourni par l'appareil homologue au SubjectAltName dans le certificat du client.

Lorsque l'authentification TLS mutuelle n'est pas utilisée, l'appareil qui écoute DOIT attendre de recevoir une demande sur

la connexion, après quoi il déduit l'identité de l'appareil qui se connecte à partir de la description de la session associée.

Lorsque la première demande arrive, son champ d'en-tête To-Path devrait contenir un URI que l'élément qui écoute a fourni dans le SDP pour une session. L'élément qui a accepté la connexion recherche l'URI dans la demande reçue et détermine à quelle session il correspond. S'il existe une correspondance, le nœud DOIT supposer que l'hôte qui a formé la connexion est l'hôte auquel cet URI a été donné. S'il n'y a pas de correspondance, le nœud DOIT rejeter la demande avec une réponse 481. Le nœud DOIT aussi vérifier que la session n'est pas déjà en cours d'utilisation sur une autre connexion. Si la session est déjà utilisée, il DOIT rejeter la demande avec une réponse 506.

Si il était légal d'avoir plusieurs connexions associées à la même session, il y aurait un problème de sécurité. Si la demande SEND initiale n'est pas protégée, un espion pourrait apprendre l'URI et l'utiliser pour insérer des messages dans la session via une autre connexion.

Si une connexion échoue pour une raison quelconque, un point d'extrémité MSRP DOIT considérer que toutes les sessions associées à la connexion ont également échoué. Quand l'un des points d'extrémité constate un tel échec, il PEUT tenter de recréer ces sessions. S'il choisit de le faire, il DOIT utiliser un nouvel échange SDP, par exemple dans un SIP re-INVITE. Si une session de remplacement est créée avec succès, les points d'extrémité PEUVENT tenter de renvoyer tout contenu pour lequel la livraison sur la session originale n'a pas pu être confirmée. Dans ce cas, les valeurs d'identifiant de message des messages renvoyés DOIVENT correspondre à celles utilisées lors des tentatives initiales. Si le point d'extrémité receveur reçoit plus d'un message avec le même identifiant de message, il DEVRAIT supposer que les messages sont des dupliqués. L'action spécifique qu'un point d'extrémité entreprend quand il reçoit un message en double est une affaire de politique locale, sauf qu'il NE DEVRAIT PAS présenter les messages dupliqués à l'utilisateur sans l'avertir de la duplication. Noter que les accusés de réception nécessaires en fonction des paramètres Failure-Report et Success-Report sont toujours nécessaires, même pour les demandes contenant un contenu dupliqué.

Quand les points d'extrémité créent une nouvelle session de cette manière, les tronçons d'un message logique donné PEUVENT être répartis entre les sessions. Cependant, les points d'extrémité NE DEVRAIENT PAS répartir les tronçons entre les sessions dans des circonstances autres que l'échec.

Si un point d'extrémité tente de recréer une session qui a échoué de cette manière, il NE DOIT PAS supposer que les URI MSRP dans le SDP seront les mêmes que les anciens.

Une connexion NE DEVRAIT PAS être fermée tant que des sessions y sont associées.

6. URI MSRP

Les URI avec les schémas "msrp" et "msrps" sont utilisés pour identifier une session de messages instantanés sur un appareil MSRP particulier, ainsi que pour identifier un relais MSRP en général. Le présent document décrit la première utilisation ; la seconde est décrite dans la spécification du relais MSRP [RFC4976]. Les URI MSRP qui identifient les sessions sont éphémères ; un appareil MSRP utilisera un URI MSRP différent pour chaque session distincte. Un URI MSRP qui identifie une session n'a aucune signification en dehors de la portée de cette session.

Un URI MSRP suit un sous-ensemble de la syntaxe d'URI de l'Appendice A de la [RFC3986], avec un schéma "msrp" ou "msrps". La syntaxe est décrite à la section 9.

Les URI MSRP sont principalement destinés à être générés et échangés entre systèmes et ne sont pas destinés à la "consommation humaine". Ils sont donc entièrement codés en US-ASCII.

Les constructions pour "authority", "userinfo" et "unreserved" sont détaillées dans la [RFC3986]. Les URI désignant MSRP sur TCP DOIVENT inclure le paramètre de transport "tcp".

Comme le présent document ne spécifie que MSRP sur TCP, tous les URI MSRP utilisent ici le paramètre de transport "tcp". Les documents qui fournissent des liens sur d'autres transports doivent définir les paramètres respectifs pour ces transports.

Le champ d'autorité de l'URI MSRP identifie un participant à une session MSRP particulière. Si le champ d'autorité contient une adresse IP numérique, il DOIT également contenir un accès. La partie "identifiant de session" identifie une session particulière du participant. L'absence de la partie identifiant de session indique une référence à un appareil d'hôte MSRP, mais ne fait pas référence à une session particulière sur cet appareil. Une valeur particulière de l'identifiant de

session n'a de sens que dans le contexte de l'autorité associée ; on peut donc considérer que le composant "autorité" identifie l'autorité qui régit un espace de noms pour l'identifiant de session.

Un schéma de "msrps" indique que la connexion sous-jacente DOIT être protégée par TLS.

MSRP a un accès recommandé enregistré auprès de l'IANA, défini au paragraphe 15.4. Cette valeur n'est pas une valeur par défaut, car le processus de négociation des URI décrit ici inclura toujours des numéros d'accès explicites. Cependant, les URI DEVRAIENT être configurés de manière à ce que l'accès recommandé soit utilisé chaque fois qu'approprié. Cela facilite la vie des administrateurs de réseau qui doivent gérer la politique de pare-feu pour MSRP.

Le composant d'autorité ne contient normalement pas de composant "userinfo", mais il PEUT le faire pour indiquer un compte d'utilisateur pour lequel la session est valide. Noter que ce n'est pas la même chose que d'identifier la session elle-même. Une partie userinfo NE DOIT PAS contenir d'informations de mot de passe.

Voici un exemple d'URI MSRP typique :

msrp://host.example.com:8493/asfd34;tcp

6.1 Comparaison d'URI MSRP

Dans le contexte du protocole MSRP, les comparaisons d'URI MSRP DOIVENT être effectuées conformément aux règles suivantes :

1. Le schéma DOIT correspondre. La comparaison de schémas est insensible à la casse.

2. Si le composant d'autorité contient une adresse IP explicite et/ou un accès, ceux-ci sont comparés pour l'équivalence d'adresse et d'accès. La normalisation du codage en pourcentage [RFC3986] s'applique, c'est-à-dire que si des caractères non réservés codés en pourcentage existent dans le composant d'autorité, ils doivent être décodés avant la comparaison. Les parties Userinfo ne sont pas prises en compte pour la comparaison des URI. Autrement, le composant d'autorité est comparé comme une chaîne de caractères insensible à la casse.

3. Si l'accès existe explicitement dans l'un ou l'autre URI, il DOIT alors correspondre exactement. Un URI avec un accès explicite n'est jamais équivalent à un autre sans accès spécifié.

4. La comparaison de la partie session-id est sensible à la casse. Un URI sans partie session-id n'est jamais équivalent à un URI qui en contient une.

5. Les URI ayant des paramètres "transport" différents ne sont jamais équivalents. Deux URI identiques à l'exception du paramètre "transport" ne sont pas équivalents. Le paramètre "transport" n'est pas sensible à la casse.

La normalisation des chemins [RFC3986] n'est pas pertinente pour les URI MSRP.

6.2 Résolution de l'appareil hôte MSRP

Un appareil hôte MSRP est identifié par le composant d'autorité d'un URI MSRP.

Si le composant d'autorité contient une adresse IP numérique et un accès, ils DOIVENT être utilisés comme indiqué.

Si le composant d'autorité contient un nom d'hôte et un accès, l'appareil qui se connecte DOIT déterminer une adresse d'hôte en effectuant une requête DNS A ou AAAA et utiliser l'accès comme indiqué.

Si une tentative de connexion échoue, l'appareil DOIT essayer de se connecter aux adresses renvoyées dans les enregistrements A ou AAAA supplémentaires, dans l'ordre dans lequel les enregistrements ont été présentés.

Ce processus suppose que l'accès de connexion est toujours connu avant la résolution. Cela est toujours vrai pour les utilisations de l'URI MSRP décrites dans le présent document, c'est-à-dire les URI échangés dans l'offre et la réponse SDP. L'introduction de relais crée des situations où ce n'est pas le cas. Par exemple, lorsqu'un utilisateur configure son client pour qu'il utilise un relais, il est souhaitable que l'URI MSRP du relais soit facile à mémoriser et à communiquer aux humains. Souvent, ce type de MSRP omettra le numéro d'accès. C'est pourquoi la spécification du relais [RFC4976] décrit des étapes

supplémentaires pour résoudre le numéro d'accès.

Les appareils MSRP PEUVENT utiliser d'autres méthodes pour découvrir d'autres appareils de ce type, quand c'est approprié. Par exemple, les points d'extrémité MSRP peuvent utiliser d'autres mécanismes pour découvrir les relais, qui sortent du domaine d'application du présent document.

7. Comportement spécifique de la méthode

7.1 Construction des demandes

Pour former une nouvelle demande, l'envoyeur crée un identifiant de transaction et l'utilise, ainsi que le nom de la méthode, pour créer une ligne de début de demande MSRP. L'identifiant de transaction NE DOIT PAS entrer en conflit avec celui d'autres transactions existant au même moment. Il DOIT donc contenir au moins 64 bits d'aléa.

Ensuite, l'envoyeur place le chemin cible dans un champ d'en-tête To-Path et l'URI de l'expéditeur dans un champ d'en-tête From-Path. Si plusieurs URI sont présents dans le To-Path, l'URI le plus à gauche est le premier URI visité ; l'URI le plus à droite est le dernier URI visité. Le traitement devient alors spécifique de la méthode. Des champs d'en-tête supplémentaires spécifiques de la méthode sont ajoutés comme décrit dans les paragraphes suivants.

Après l'ajout des champs d'en-tête spécifiques de la méthode, le traitement se poursuit pour traiter le corps de la demande, s'il est présent. Si la demande a un corps, il DOIT contenir un champ d'en-tête Content-Type. Il peut contenir d'autres champs d'en-tête spécifiques de MIME. Le champ d'en-tête Content-Type DOIT être le dernier champ de la section d'en-tête du message. Le corps du message DOIT être séparé des champs d'en-tête par un CRLF supplémentaire.

Les demandes non SEND ne sont pas destinées à porter un contenu de message et ne sont donc pas interruptibles. Les corps de demandes non SEND NE DOIVENT PAS dépasser 10 240 octets.

Bien que le présent document ne traite pas de l'utilisation particulière des corps dans les demandes non SEND, ils pourraient être utiles à l'avenir pour porter des informations sur la sécurité ou l'identité, des informations sur un message en cours, etc. La limite de 10 240 caractères a été choisie pour être suffisamment grande pour la plupart de ces applications, mais suffisamment petite pour éviter les problèmes d'équité causés par l'envoi d'un contenu arbitrairement volumineux dans des corps de méthode non interruptibles.

Une demande sans corps ne DOIT PAS inclure un Content-Type ou tout autre champ d'en-tête spécifique de MIME. Une demande sans corps DOIT contenir une ligne de fin après le dernier champ d'en-tête. Aucun CRLF supplémentaire ne sera présent entre la section d'en-tête et la ligne de fin.

Les demandes sans corps sont utiles quand un client souhaite envoyer du "trafic", mais ne souhaite pas envoyer de contenu à rendre à l'utilisateur homologue. Par exemple, le point d'extrémité actif envoie une demande SEND immédiatement après avoir établi une connexion. Si il n'a rien à dire pour le moment, il peut envoyer une demande sans corps. Les demandes sans corps peuvent également être utilisées dans certaines applications pour maintenir en vie les liens de traduction d'adresse réseau (NAT, *Network Address Translation*), etc.

Les demandes sans corps sont distinctes des demandes avec un corps vide. Une demande avec un corps vide aura une valeur de champ d'en-tête Content-Type et sera généralement rendue au destinataire selon les règles de ce type.

La ligne de fin qui termine la demande DOIT être composée de sept caractères "-" (signe moins), de l'identifiant de transaction utilisé dans la ligne de début et d'un caractère fanion. Si un corps est présent, la ligne de fin DOIT être précédée d'un CRLF qui ne fait pas partie du corps. Si le tronçon représente les données qui forment la fin du message complet, la valeur du fanion DOIT être un "$". Si l'envoyeur interrompt un message incomplet et n'a pas l'intention d'envoyer d'autres tronçons dans ce message, le fanion DOIT être un "#". Dans le cas contraire, le fanion DOIT être un "+".

Si la demande contient un corps, l'envoyeur DOIT s'assurer que la ligne de fin (sept traits d'union, l'identificateur de transaction et un fanion de continuation) n'est pas présente dans le corps. Un receveur qui détecte une ligne de fin présente dans le corps précédée par une séquence non vide autre que CRLF DEVRAIT terminer la session. Si la ligne de fin est présente dans le corps, l'envoyeur DOIT choisir un nouvel identifiant de transaction qui n'est pas présent dans le corps, et ajouter un CRLF si nécessaire, et la ligne de fin, y compris le caractère "$", "#" ou "+".

Certaines mises en œuvre peuvent choisir de rechercher la séquence de cloture lors de l'envoi du corps et, si elle est rencontrée, d'interrompre simplement le tronçon à ce point et de commencer une nouvelle transaction avec un identifiant de

transaction différent pour porter le reste du corps. D'autres mises en œuvre peuvent choisir d'examiner les données et de s'assurer que le corps ne contient pas l'identifiant de transaction avant de commencer l'envoi de la transaction.

Lorsqu'une demande est prête à être transmise, l'envoyeur suit les règles de gestion de connexions (paragraphe 5.4) pour transmettre la demande sur une connexion ouverte existante ou créer une nouvelle connexion.

7.1.1 Envoi des demandes SEND

Lorsqu'un point d'extrémité a un message à livrer, il génère d'abord un nouvel identifiant de message. La valeur DOIT être très peu susceptible d'être répétée à l'avenir par une autre instance de point d'extrémité, ou par la même instance. Si nécessaire, le point d'extrémité divise le message en tronçons. Il génère ensuite une demande SEND pour chaque tronçon, en suivant les procédures de construction des demandes (paragraphe 7.1).

Le champ d'en-tête Message-ID fournit un identifiant de message unique qui se réfère à une version particulière d'un message particulier. Dans ce contexte, le terme "message" désigne une unité de contenu que l'envoyeur souhaite transmettre au destinataire. Bien qu'un tel message puisse être divisé en tronçons, l'identifiant de message se réfère au message entier, et non à un tronçon du message.

L'unicité de l'identifiant de message est assurée par l'hôte qui le génère. Cet identifiant de message est destiné à être lisible par la machine et n'est pas nécessairement signifiant pour l'homme. Un identifiant de message concerne exactement une version d'un message particulier ; les révisions ultérieures du message reçoivent chacune un nouvel identifiant de message. Les points d'extrémité peuvent assurer une unicité suffisante de plusieurs façons, dont la sélection est un choix de mise en œuvre. Par exemple, un point d'extrémité pourrait enchaîner un identifiant d'instance comme une adresse MAC, son idée du nombre de secondes écoulées depuis un certain instant, un identifiant de processus et un nombre entier de 16 bits à croissance monotone, tous codés en base-64. Autrement, un point d'extrémité sans horloge incorporée pourrait simplement utiliser un nombre aléatoire de 64 bits.

Chaque tronçon d'un message DOIT contenir un champ d'en-tête Message-ID contenant l'identifiant du message. Si l'envoyeur souhaite un rapport d'état autrement que par défaut, il DOIT insérer un champ d'en-tête Failure-Report et/ou Success-Report avec une valeur appropriée. Tous les tronçons d'un même message DOIVENT utiliser les mêmes valeurs de Failure-Report et Success-Report dans leurs demandes SEND.

Si des rapports de réussite sont demandés, c'est-à-dire si la valeur du champ d'en-tête Success-Report est "oui", l'appareil envoyeur PEUT souhaiter lancer un temporisateur d'une valeur appropriée pour son application et prendre des mesures si un rapport de réussite n'est pas reçu dans ce laps de temps. Il n'existe pas de valeur universelle pour ce temporisateur. Pour de nombreuses applications de messagerie instantanée, ce peut être de 2 minutes, tandis que pour certains systèmes de négociation, ce peut être de moins d'une seconde. Indépendamment de l'utilisation d'un tel temporisateur, si le rapport de succès n'a pas été reçu au moment où la session est terminée, l'appareil DEVRAIT en informer l'utilisateur.

Si la valeur de "Failure-Report" est fixée à "oui", l'envoyeur de la demande lance un temporisateur. Si une réponse 200 à la transaction n'est pas reçue dans les 30 secondes qui suivent l'envoi du dernier octet de la transaction ou sa soumission au système d'exploitation pour envoi, l'élément DOIT informer l'utilisateur que la demande a probablement échoué. Si la valeur est "partiel", l'élément qui envoie la transaction n'est pas tenu de lancer un temporisateur, mais il DOIT informer l'utilisateur s'il reçoit une réponse d'erreur non récupérable à la transaction. Quelle que soit la valeur du rapport d'échec, il n'est pas nécessaire d'attendre une réponse avant d'envoyer la demande suivante.

Le traitement des temporisateurs pour les rapports de réussite et d'échec est intentionnellement non cohérent. Une valeur explicite de temporisation a un sens pour les rapports d'échec, car ces rapports se réfèrent généralement à un "tronçon" de message dont l'accusé de réception se fait bond par bond. Ce n'est pas le cas pour les rapports de succès, qui sont transmis de bout en bout et peuvent se référer au contenu du message entier, qui peut être arbitrairement volumineux.

Si aucun champ d'en-tête Success-Report n'est présent dans une demande SEND, elle DOIT être traitée de la même manière qu'avec un champ d'en-tête Success-Report ayant la valeur "non". Si aucun champ d'en-tête Failure-Report n'est présent, elle DOIT être traitée de la même manière qu'avec un champ d'en-tête Failure-Report ayant la valeur "oui". Si un point d'extrémité MSRP reçoit un RAPPORT pour un identifiant de message qu'il ne reconnaît pas, il DOIT ignorer le RAPPORT en silence.

La valeur du champ d'en-tête Byte-Range contient une valeur de départ (range-start) suivie d'un "-", une valeur de fin (range-end) suivie d'un "/", et enfin la longueur totale. Le premier octet du message a une position de un, plutôt que zéro.

Le premier tronçon du message DEVRAIT, et tous les tronçons suivants DOIVENT, inclure un champ d'en-tête Byte-Range. Le champ range-start DOIT indiquer la position du premier octet du corps du message dans le message complet (pour le premier tronçon, ce champ aura une valeur de un). Le champ range-end DOIT indiquer la position du dernier octet dans le corps du message, s'il est connu. Il DOIT prendre la valeur "*" si la position est inconnue ou si la demande doit être interruptible. Le champ total DEVRAIT contenir la taille totale du message, si elle est connue. Le champ total PEUT contenir un "*" si la taille totale du message n'est pas connue à l'avance. L'envoyeur DOIT envoyer tous les tronçons dans l'ordre de la plage d'octets. (Cependant, le receveur ne peut pas supposer que les demandes seront livrées dans l'ordre, car des relais intermédiaires peuvent avoir modifié l'ordre).

Dans certaines circonstances, un point d'extrémité peut choisir d'envoyer une demande SEND vide. Par souci de cohérence, un champ d'en-tête Byte-Range se référant à un contenu inexistant ou de longueur nulle DOIT quand même avoir une valeur de début de plage de 1. Par exemple, "1-0/0".

Pour garantir l'équité sur une connexion, les envoyeurs NE DOIVENT PAS envoyer de tronçons dont le corps dépasse 2048 octets, à moins qu'ils ne soient prêts à les interrompre (ce qui signifie que tout tronçon dont le corps dépasse 2048 octets aura un caractère "*" dans le champ de fin de plage). Un envoyeur peut utiliser une des deux stratégies suivantes pour satisfaire à cette exigence. Il est FORTEMENT RECOMMANDÉ à l'envoyeur d'envoyer des messages de plus de 2048 octets en utilisant aussi peu de tronçons que possible, en interrompant les tronçons (longs d'au moins 2048 octets) uniquement lorsque un autre trafic est en attente d'utilisation de la même connexion. Autrement, l'envoyeur PEUT simplement envoyer des tronçons par incréments de 2048 octets jusqu'au dernier tronçon. Noter que la première stratégie permet une utilisation nettement plus efficace de la connexion. Tous les nœuds MSRP DOIVENT être capables de recevoir des tronçons de n'importe quelle taille, de zéro octet au nombre maximum d'octets qu'ils peuvent recevoir pour un message complet. Les envoyeurs NE DEVRAIENT PAS diviser les messages en tronçons de moins de 2048 octets, à l'exception du dernier tronçon d'un message complet.

Une demande SEND est interrompue pendant qu'un corps est en cours d'écriture sur la connexion en notant simplement quelle quantité du message a déjà été écrite sur la connexion, puis en écrivant la ligne de fin pour terminer le tronçon. Il peut ensuite être repris dans un autre tronçon avec le même identifiant de message et un champ de début de gamme de champ d'en-tête Plage d'octets contenant la position du premier octet après l'interruption.

Les demandes SEND de plus de 2048 octets DOIVENT être interrompues si l'envoyeur a besoin d'envoyer des réponses en attente ou des demandes REPORT. Si plusieurs demandes SEND provenant de différentes sessions sont envoyées concurramment sur la même connexion, l'appareil DEVRAIT mettre en œuvre un schéma permettant d'alterner entre elles de manière à ce que chaque demande concurrente ait la possibilité d'envoyer une partie équitable des données à des intervalles réguliers adaptés à l'application.

L'envoyeur NE DOIT PAS supposer qu'un message est reçu par l'homologue avec la même allocation de tronçons que celle envoyée. Un relais intermédiaire pourrait éventuellement diviser les demandes SEND en tronçons plus petits ou regrouper plusieurs tronçons en tronçons plus grands.

La disposition par défaut des messages est d'être rendue à l'utilisateur. Si l'envoyeur veut une disposition différente, il PEUT insérer un champ d'en-tête Content-Disposition [RFC2183]. Les valeurs PEUVENT inclure toutes celles de la [RFC2183] ou du registre IANA qui les définit. Comme MSRP peut porter toute charge utile binaire non codée, le codage de transfert est toujours "binary", et les paramètres de codage de transfert NE DOIVENT PAS être présents.

7.1.2 Envoi des demandes REPORT

Les demandes REPORT sont similaires aux demandes SEND, sauf que les demandes de rapport NE DOIVENT PAS inclure de champs d'en-tête Success-Report ou Failure-Report, et DOIVENT contenir un champ d'en-tête État. Les demandes REPORT DOIVENT contenir le champ d'en-tête Identifiant de message de la demande SEND d'origine.

Si un élément MSRP reçoit un REPORT pour un identifiant de message qu'il ne reconnaît pas, il DEVRAIT ignorer en silence le REPORT.

Un point d'extrémité MSRP DOIT être capable de générer des demandes REPORT de succès.

Les demandes REPORT ne comportent normalement pas de corps, car les champs de l'en-tête de la demande REPORT peuvent contenir suffisamment d'informations dans la plupart des cas. Cependant, les demandes REPORT PEUVENT inclure un corps contenant des informations supplémentaires sur l'état de la demande SEND associée. Un tel corps n'a qu'une valeur d'information et l'envoyeur de la demande REPORT NE DEVRAIT PAS supposer que le receveur y prête

attention. Les demandes REPORT ne sont pas interruptibles.

Les champs d'en-tête Success-Report et Failure-Report NE DOIVENT PAS être présents dans les demandes REPORT. Les nœuds MSRP NE DOIVENT PAS envoyer de demandes REPORT en réponse à des demandes REPORT. Les nœuds MSRP NE DOIVENT PAS envoyer de réponses MSRP aux demandes REPORT.

Les points d'extrémité NE DEVRAIENT PAS envoyer de demandes REPORT s'ils ont des raisons de penser que la demande ne sera pas livrée. Par exemple, ils NE DEVRAIENT PAS envoyer de demande REPORT pour une session qui n'est plus valide.

7.1.3 Générer des rapports de succès

Lorsqu'un point d'extrémité reçoit un message en un ou plusieurs tronçons contenant la valeur "Success-Report" de "oui", il DOIT envoyer un ou plusieurs rapports de réussite couvrant tous les octets reçus avec succès. Les rapports de réussite sont envoyés sous la forme de demandes REPORT, en suivant les procédures normales (paragraphe 7.1) avec quelques exigences supplémentaires.

Le receveur PEUT attendre de recevoir le dernier tronçon d'un message et envoyer un rapport de réussite couvrant l'ensemble du message. Autrement, il peut générer des REPORT de réussite incrémentiels au fur et à mesure de la réception des tronçons. Ceux-ci peuvent être envoyés périodiquement et couvrir tous les octets qui ont été reçus jusqu'à présent, ou ils peuvent être envoyés après l'arrivée d'un tronçon et couvrir uniquement la partie de ce tronçon.

Il est utile de considérer qu'un rapport de réussite fait état d'une plage particulière d'octets, plutôt que d'un tronçon particulier envoyé par un client. Le client envoyeur ne peut pas compter sur le fait que le champ d'en-tête Byte-Range d'un rapport de réussite donné corresponde à celui d'une demande SEND particulière. Par exemple, un relais MSRP intermédiaire peut diviser les tronçons en tronçons plus petits ou regrouper plusieurs tronçons en tronçons plus grands. Un effet collatéral est que, même si aucun relais n'est utilisé, le client receveur peut signaler des plages d'octets qui ne correspondent pas exactement à celles des tronçons originaux envoyés par l'expéditeur. Il peut attendre que tous les octets d'un message soient reçus et faire un rapport sur l'ensemble, il peut faire un rapport au fur et à mesure qu'il reçoit chaque tronçon, ou il peut faire un rapport sur n'importe quelle autre plage reçue. Les rapports sur des plages plus petites que l'ensemble du contenu du message permettent d'améliorer le ressenti de l'utilisateur pour l'envoyeur. Par exemple, un client émetteur pourrait afficher des informations d'état incrémentielles montrant quelles plages d'octets ont été acquittées par le destinataire. Cendant, le choix d'afficher ou non des informations incrémentielles incombe entièrement au client receveur. Il n'y a pas de mécanisme pour que l'envoyeur affirme son désir de recevoir ou non des rapports incrémentiels. Comme la présence d'un relais peut amener le receveur à voir une allocation de tronçons très différente de celle de l'envoyeur, un tel mécanisme serait d'une valeur discutable.

Quand il génère une demande REPORT, le point d'extrémité insère un champ d'en-tête To-Path contenant la valeur From-Path de la demande originale, et un champ d'en-tête From-Path contenant l'URI l'identifiant lui-même dans la session. Le point d'extrémité insère ensuite un champ d'en-tête État avec un espace de noms de "000", un code d'état de "200" et une phrase de commentaire définie par la mise en œuvre. Il insère aussi un champ d'en-tête Identifiant de message contenant la valeur de la demande initiale.

Le champ Espace de noms indique le contexte du champ Code d'état. La valeur d'espace de noms de "000" signifie que le code d'état devrait être interprété de la même manière que le code de réponse de la transaction MSRP correspondante. Si une future spécification utilise le champ "Code d'état" à d'autres fins, elle DOIT définir une nouvelle valeur pour le champ "Espace de noms".

Le point d'extrémité NE DOIT PAS envoyer de rapport de réussite pour une demande SEND qui ne contient pas de champ d'en-tête Success-Report ou qui contient un tel champ avec la valeur "non". C'est-à-dire, si aucun champ d'en-tête Success-Report n'est présent, cela est traité de la même manière qu'un champ ayant la valeur "non".

7.1.4 Générer des rapports d'échec

Si un point d'extrémité MSRP reçoit une demande SEND qu'il ne peut pas traiter pour une raison quelconque et si le champ d'en-tête Failure-Report n'était pas présent dans la demande initiale ou avait la valeur "oui", il DEVRAIT simplement inclure le code d'erreur approprié dans la réponse de la transaction. Cependant, il peut y avoir des situations où l'erreur ne peut pas être déterminée rapidement, comme, par exemple, lorsque le point d'extrémité est une passerelle qui attend qu'un réseau en aval indique une erreur. Dans cette situation, il PEUT envoyer une réponse 200 OK à la demande, puis envoyer

une demande REPORT d'échec lorsque l'erreur est détectée.

Si le point d'extrémité reçoit une demande SEND dont le champ d'en-tête Failure-Report a la valeur "non", il ne DOIT alors PAS envoyer de demande REPORT d'échec ni de réponse de transaction. Si la valeur est "partiel", il NE DOIT PAS envoyer une réponse de transaction 200 à la demande, mais DEVRAIT envoyer une réponse appropriée de classe non 200 si un échec se produit.

Comme indiqué ci-dessus, si aucun champ d'en-tête Failure-Report n'est présent, cela DOIT être traité de la même manière qu'un champ d'en-tête Failure-Report ayant la valeur de "oui".

La construction des demandes REPORT d'échec est identique à celle des demandes REPORT de succès, sauf que le champ de code de l'en-tête État DOIT contenir le code d'erreur approprié. Tout code de réponse d'erreur défini dans la présente spécification PEUT également être utilisé dans les rapports d'échec.

Si une demande REPORT d'échec est envoyée en réponse à une demande SEND contenant un tronçon, elle DOIT inclure un champ d'en-tête Byte-Range indiquant la plage réelle faisant l'objet du rapport. Elle peut reprendre les valeurs de début de plage et de total de la demande SEND d'origine, mais DOIT calculer le champ de fin de plage à partir des données réelles du corps.

Ce paragraphe ne décrit que le comportement de génération de rapports d'échec pour les points d'extrémité MSRP. Le comportement de relais sort du domaine d'application du présent document et sera examiné dans un document distinct [RFC4976]. On s'attend à ce que les rapports d'échec soient générés plus souvent par les relais que par les points d'extrémité.

7.2 Construction des réponses

Si un point d'extrémité MSRP reçoit une demande qui contient un champ d'en-tête Failure-Report de valeur "oui" ou qui ne contient pas de champ d'en-tête Failure-Report du tout, il DOIT immédiatement générer une réponse. De même, si un point d'extrémité MSRP reçoit une demande contenant un champ d'en-tête Failure-Report de valeur "partiel" et que le destinataire est incapable de traiter la demande, il DEVRAIT immédiatement générer une réponse.

Pour construire la réponse, le point d'extrémité crée d'abord la ligne de début de réponse, en insérant le code de réponse approprié et facultativement un commentaire. L'identifiant de transaction dans la ligne de début de réponse DOIT correspondre à l'identifiant de transaction de la demande originale.

Le point d'extrémité insère ensuite un champ d'en-tête To-Path approprié. Si la demande qui déclenche la réponse est une demande SEND, le champ d'en-tête To-Path est formé en copiant le premier URI (le plus à gauche) dans le champ d'en-tête From-Path de la demande. (Les réponses aux demandes SEND ne sont renvoyées qu'au bond précédent.) Pour les réponses à toutes les autres méthodes de demande, le champ d'en-tête To-Path contient le chemin complet de retour vers l'envoyeur d'origine. Ce chemin complet est généré en copiant la liste des URI du champ From-Path de la demande originale dans le champ To-Path de la réponse. (Les demandes REPORT légales n'exigent pas de réponse, donc la présente spécification ne respecte pas le comportement décrit ci-dessus ; cependant, on s'attend à ce que des extensions pour les passerelles et les relais aient besoin d'un tel comportement).

Enfin, le point d'extrémité insère un champ d'en-tête From-Path contenant l'URI qui l'identifie dans le contexte de la session, suivi de la ligne de fin après le dernier champ d'en-tête. Comme une réponse n'est jamais découpée en tronçons, le fanion de continuation dans la ligne de fin contiendra toujours un signe dollar ("$"). La réponse DOIT être retransmise sur la même connexion que celle sur laquelle la demande originale est arrivée.

7.3 Réception des demandes

Le point d'extrémité receveur DOIT d'abord vérifier l'URI dans le champ To-Path pour s'assurer que la demande appartient à une session existante. Quand la demande est reçue, le champ To-Path contient exactement un URI, qui DOIT se transposer en une session existante associée à la connexion sur laquelle la demande est arrivée. Si ce n'est pas le cas, le receveur DOIT générer une erreur 481 et ignorer la demande. Noter que si le champ d'en-tête Failure-Report a la valeur de "non", aucun rapport d'erreur n'est alors envoyé.

Le traitement ultérieur de la demande par le receveur est spécifique de la méthode.

7.3.1 Réception des demandes SEND

Quand le point d'extrémité receveur reçoit une demande SEND, il détermine d'abord si elle contient un message complet ou un tronçon d'un message plus important. Si la demande ne contient pas de champ d'en-tête Byte-Range ou en contient un avec une valeur de début de plage de "1", et si le fanion de continuation de ligne de cloture a une valeur de "$", alors la demande contenait le message entier. Autrement, le receveur regarde la valeur de l'identifiant de message pour associer les tronçons ensemble dans le message d'origine. Le receveur forme une mémoire tampon virtuelle pour recevoir le message, en gardant trace des octets reçus et de ceux qui manquent. Le receveur prend les données de la demande et les place à l'endroit approprié dans la mémoire tampon. Le receveur DEVRAIT déterminer la longueur réelle de chaque tronçon en inspectant la charge utile elle-même ; il est possible que le corps soit plus court que ne l'indique le champ "range-end". Cela peut se produire si l'envoyeur a interrompu une demande SEND de manière inattendue. Il convient de noter que le tronçon dont le caractère de fin est "$" définit la longueur totale du message.

Il est techniquement illégal pour l'envoyeur d'interrompre prématurément une demande dont le dernier octet du champ d'en-tête "Byte-Range" ne contient rien d'autre que "*". Mais le fait que le receveur calcule la longueur d'un tronçon sur la base du contenu réel ajoute de la résilience face aux erreurs de l'envoyeur. Comme cela ne devrait jamais se produire avec des envoyeurs conformes, ce point n'a qu'une valeur de "DEVRAIT".

Les receveurs NE DOIVENT PAS supposer que les tronçons seront livrés dans l'ordre ou qu'ils recevront tous les tronçons avec des fanions "+" avant de recevoir le tronçon avec le fanion "$". Dans certains cas d'échec de la connexion, il est possible que des informations soient dupliquées. En cas de réception d'un tronçon de données qui chevauche des données déjà reçues pour le même message, le dernier tronçon reçu DEVRAIT avoir la priorité (même s'il ne s'agit pas du dernier tronçon transmis). Par exemple, si les octets 1 à 100 ont été reçus et qu'un tronçon arrive contenant les octets 50 à 150, ce deuxième tronçon écrasera les octets 50 à 100 des données déjà reçues. Bien que d'autres schémas fonctionnent, celui-ci est le plus simple pour le receveur et résulte en un comportement cohérent entre les clients.

Il y a des situations où le receveur peut n'être pas capable de donner la priorité au dernier tronçon reçu quand des tronçons se chevauchent. Par exemple, le receveur peut rendre les morceaux de manière incrémentaire au fur et à mesure de leur arrivée. Si un nouveau tronçon arrive et chevauche un tronçon rendu précédemment, il serait trop tard pour "reprendre" les données conflictuelles du premier tronçon. Donc, l'exigence de donner la priorité au tronçon le plus récent est spécifiée au niveau "DEVRAIT". Cette exigence n'a pas pour but d'interdire les applications où ce comportement n'a pas de sens.

Les sept "-" de la ligne de fin sont utilisés pour que le reeveur puisse rechercher la valeur "----", 32 bits à la fois, pour trouver l'emplacement probable de la ligne de fin. Cela permet à la plupart des processeurs de localiser les limites et de copier la mémoire à la même vitesse qu'une copie normale de la mémoire. Cette approche résulte en un système aussi rapide que le tramage fondé sur la spécification de la longueur du corps dans les champs d'en-tête de la demande, mais elle permet aussi d'interrompre les messages.

Ce qui est fait avec le corps du message sort du champ d'application de MSRP et est largement déterminé par le type de contenu MIME et la disposition du contenu (Content-Disposition). Le corps du message PEUT être rendu après réception du message complet, ou partiellement rendu au fur et à mesure de sa réception.

Si la demande SEND contenait un champ d'en-tête Content-Type indiquant un type de supports non pris en charge et si la valeur de Failure-Report n'est pas "non", le receveur DOIT générer une réponse avec un code d'état de 415. Tous les points d'extrémité MSRP DOIVENT être capables de recevoir les types de supports multipart/mixed [RFC2046] et multipart/alternative [RFC2046].

Si le champ d'en-tête Success-Report a la valeur de "oui", le receveur doit construire et envoyer un ou plusieurs rapports de réussite, comme décrit au paragraphe 7.1.3.

7.3.2 Réception des demandes REPORT

Quand un point d'extrémité reçoit une demande REPORT, il fait une corrélation entre le rapport et la demande SEND originale en utilisant l'identifiant de message et la plage d'octets, si elle est présente. Si il a demandé des rapports de succès, il DEVRAIT conserver suffisamment d'état sur chaque message envoyé en instance pour pouvoir établir une corrélation entre les demandes REPORT et les messages originaux.

Un point d'extrémité qui reçoit une demande REPORT contenant un champ d'en-tête État avec un champ d'espace de noms de "000" DOIT interpréter le rapport exactement de la même manière qu'il interpréterait une réponse de transaction MSRP avec un code de réponse correspondant au champ Code d'état.

Il est possible de recevoir un rapport d'échec ou une réponse d'échec de transaction pour un tronçon en cours de livraison. Dans ce cas, l'ensemble du message correspondant à ce tronçon DEVRAIT être interrompu, en incluant le caractère "#" dans le champ de continuation de la ligne de fin.

Il est possible qu'un point d'extrémité reçoive une demande REPORT sur une session qui n'est plus valide. Le comportement du point d'extrémité dans ce cas est une question de politique locale. Le point d'extrémité n'est pas tenu de prendre des mesures pour faciliter une telle livraison tardive ; c'est-à-dire qu'il n'est pas censé maintenir une connexion active au cas où des REPORT tardifs arriveraient.

Quand un point d'extrémité qui a envoyé une demande SEND reçoit un rapport d'échec indiquant qu'une plage d'octets particulière n'a pas été reçue, il DOIT considérer que la session a échoué. S'il souhaite la récupérer, il DOIT d'abord renégocier les URI au niveau de la signalisation, puis renvoyer cette plage d'octets du message dans le cadre de la nouvelle session résultante.

Les nœuds MSRP NE DOIVENT PAS envoyer de demandes REPORT MSRP en réponse à d'autres demandes REPORT.

8. Utilisation de MSRP avec SIP et SDP

Les sessions MSRP vont normalement être initiées en utilisant le protocole de description de session (SDP, *Session Description Protocol*) [RFC4566] via le mécanisme SIP d'offre/réponse [RFC3264].

Le présent document définit quelques nouveaux paramètres SDP pour établir des sessions MSRP. Ces paramètres sont détaillés ci-dessous et dans la section sur les considérations relatives à l'IANA.

Une ligne de supports MSRP (c'est-à-dire, une ligne de supports proposant MSRP) dans la description de session est accompagnée d'un attribut "path" obligatoire. Cet attribut contient une liste séparée par des espaces des URI à visiter pour contacter l'agent utilisateur qui annonce cette description de session. Si plus d'un URI est présent, l'URI le plus à gauche est le premier URI à visiter pour atteindre la ressource cible. (La liste des chemins peut contenir plusieurs URI afin de permettre à l'avenir le déploiement de passerelles ou de relais). Les mises en œuvre de MSRP qui peuvent accepter des connexions entrantes sans avoir besoin de relais ne fourniront généralement ici qu'un seul URI.

Une ligne de supports MSRP est aussi accompagnée d'un attribut "accept-types" et facultativement d'un attribut "accept-wrapped-types". Ces attributs sont utilisés pour spécifier les types de supports acceptables par le point d'extrémité.

8.1 Connexion et lignes de supports SDP

Une ligne de connexion SDP a le format suivant :

c=<type de réseau> <type d'adresse> <adresse de connexion>

Figure 4 : Ligne de connexion SDP standard

Les champs Type de réseau et type d'adresse sont utilisés comme normalement pour SDP. Le champ Adresse de connexion DOIT être réglé à l'adresse IP ou au nom de comaine pleinement qualifié provenant de l'URI MSRP qui identifie le point d'extrémité dans son attribut Path.

Le format général d'une ligne de supports SDP est le suivant :

m=<support> <accès> <protocole> <liste de formats>

Figure 5 : Ligne de supports SDP standard

Une ligne de supports offerte ou acceptée pour MSRP sur TCP DOIT inclure une valeur de champ de protocole de "TCP/MSRP", ou "TCP/TLS/MSRP" pour TLS. La valeur du champ Supports DOIT être "message". Le champ de la liste des formats DOIT être réglée à la valeur "*".

La valeur du champ Accès DOIT correspondre à la valeur de l'accès utilisée dans l'URI MSRP du point d'extrémité dans l'attribut Path, sauf que, comme décrit dans la [RFC3264], un agent d'utilisateur qui souhaite accepter une offre, mais pas une ligne de supports spécifique, DOIT régler le numéro d'accès de cette ligne de supports à zéro (0) dans la réponse.

Comme MSRP permet à plusieurs sessions de partager la même connexion TCP, plusieurs lignes m dans un seul document SDP peuvent partager la même valeur de champ d'accès ; les appareils MSRP NE DOIVENT PAS supposer une relation particulière entre les lignes m sur la seule base du fait qu'elles ont des valeurs de champ d'accès correspondantes.

Les appareils MSRP n'utilisent pas le champ d'adresse Ligne c, ni les champs accès et liste de format de ligne m pour déterminer où se connecter. Ils utilisent plutôt les attributs définis dans la présente spécification. Les informations de connexion sont copiées sur la ligne c et la ligne m à des fins de rétro compatibilité avec les utilisations SDP conventionnelles. Bien que MSRP puisse théoriquement porter n'importe quel type de support, "message" est approprié.

8.2 Négociations d'URI

Chaque point d'extrémité d'une session MSRP est identifié par un URI. Ces URI sont négociés dans l'échange SDP. Chaque offre ou réponse SDP qui propose MSRP DOIT contenir un attribut "path" contenant un ou plusieurs URI MSRP. L'attribut Path est utilisé dans une ligne a SDP et a la syntaxe suivante :

```
path = path-label ":" path-list
path-label = "path"
path-list= MSRP-URI *(SP MSRP-URI)
```

Figure 6 : Attribut Path

où MSRP-URI est un URI "msrp" ou "msrps" tel que défini à la Section 6. Les URI MSRP inclus dans une offre ou une réponse SDP DOIVENT inclure des numéros d'accès explicites.

Un appareil MSRP utilise l'URI pour déterminer l'adresse de l'hôte, l'accès, le transport et le niveau de protection lors de la connexion, et pour identifier la cible lors de l'envoi des demandes et des réponses.

L'offreur et le répondant sélectionnent chacun un URI pour se représenter et envoient cet URI à leur homologue dans le document SDP. Chaque homologue mémorise la valeur du chemin reçue de l'autre homologue et l'utilise comme cible pour les demandes dans la session résultante. Si l'attribut de chemin reçu de l'homologue contient plus d'un URI, l'URI cible est alors le plus à droite, tandis que l'entrée la plus à gauche représente le bond adjacent. Si une seule entrée est présente, il s'agit de l'URI de l'homologue et de l'URI du bond adjacent. Le chemin cible est la valeur entière de l'attribut de chemin reçue de l'homologue.

L'exemple suivant montre une offre SDP avec un URI de session de "msrp://alice.exemple.com:7394/2s93i9ek2a;tcp"

```
v=0
o=alice 2890844526 2890844527 IN IP4 alice.exemple.com
s= -
c=IN IP4 alice.exemple.com
t=0 0
m=message 7394 TCP/MSRP *
a=accept-types:text/plain
a=path:msrp://alice.exemple.com:7394/2s93i9ek2a;tcp
```

Figure 7 : Exemple de SDP avec l'attribut Path

L'URI le plus à droite dans l'attribut path DOIT identifier le point d'extrémité qui a généré le document SDP, ou un autre emplacement où ce point d'extrémité souhaite recevoir les demandes associées à la session. Il DOIT être alloué pour cette session particulière et NE DOIT PAS dupliquer un URI utilisé pour une autre session à laquelle le point d'extrémité participe actuellement. Il DEVRAIT être difficile à deviner et protégé contre les écoutes clandestines. Ce point est abordé plus en détail à la Section 14.

8.3 Attributs de chemin avec plusieurs URI

Comme mentionné précédemment, le présent document décrit MSRP pour des scénarios d'homologue à homologue, c'est-à-dire, quand aucun relais n'est utilisé. L'utilisation de relais est décrite dans un document distinct [RFC4976]. Afin de permettre à un appareil MSRP qui ne met en œuvre que le cœur de cette spécification d'interopérer avec les appareils qui utilisent des relais, le présent document doit inclure quelques hypothèses sur la façon dont fontionnent les relais.

Un point d'extrémité qui utilise un ou plusieurs relais l'indiquera en plaçant un URI pour chaque appareil de la chaîne de

relais dans l'attribut de chemin SDP. La dernière entrée pointera sur le point d'extrémité lui-même. Les autres entrées vont indiquer chaque relais proposé, dans l'ordre. La première entrée va indiquer le premier relais de la chaîne du point de vue de l'homologue, c'est-à-dire le relais auquel l'appareil homologue, ou un relais opérant en son nom, devrait se connecter.

Les points d'extrémité qui ne souhaitent pas insérer de relais, y compris ceux qui ne prennent pas du tout en charge les relais, indiqueront exactement un URI dans l'attribut path. Cet URI représente à la fois le point d'extrémité de la session et le point de connexion.

Même si les points d'extrémité qui ne mettent en œuvre que cette spécification n'introduiront jamais de relais, ils doivent pouvoir interopérer avec d'autres points d'extrémité qui utilisent des relais. Par conséquent, ils DOIVENT être prêts à recevoir plus d'un URI dans l'attribut de chemin SDP. Lorsqu'un point d'extrémité reçoit plus d'un URI dans un attribut de chemin, seule la première entrée est pertinente pour la résolution de l'adresse et de l'accès, et pour l'établissement de la connexion réseau, car elle décrit le premier bond adjacent.

Si un point d'extrémité place plus d'un URI dans un attribut de chemin, le dernier URI dans l'attribut de chemin (l'URI de l'homologue) identifie la session et NE DOIT PAS dupliquer l'URI d'une autre session à laquelle le point d'extrémité participe actuellement. Les exigences d'unicité pour les autres entrées de l'attribut Path sortent du domaine d'application du présent document.

8.4 Mise à jour des offres SDP

Les points d'extrémité MSRP peuvent parfois avoir besoin d'envoyer des échanges SDP supplémentaires pour une session existante. Ils peuvent avoir besoin d'envoyer des échanges périodiques sans modification de l'état de rafraîchissement dans le réseau, par exemple, des temporisateurs de session SIP ou la demande SIP UPDATE [RFC3311]. Ils peuvent avoir besoin de modifier un autre flux dans une session sans affecter le flux MSRP, ou ils peuvent avoir besoin de modifier un flux MSRP sans affecter un autre flux.

L'un ou l'autre des homologues peut initier un échange de mise à jour à tout moment. Le point d'extrémité qui envoie la nouvelle offre assume le rôle d'offreur dans tous les cas. Celui qui répond DOIT le faire avec un attribut de chemin qui représente un chemin valide vers lui-même au moment de l'échange mis à jour. Ce nouveau chemin peut être le même que le précédent, mais il peut aussi être différent. Le nouvel offreur NE DOIT PAS supposer que l'homologue répondra avec le même chemin que celui qu'il a utilisé précédemment.

Si l'une des parties souhaite envoyer un document SDP qui ne change rien du tout, elle DOIT alors utiliser la même ligne o que dans l'échange précédent.

8.5 Négociation de connexion

Les versions précédentes de ce document incluaient un mécanisme pour négocier la direction de toute connexion TCP requise. Ce mécanisme était vaguement fondé sur les travaux du groupe de travail MMUSIC sur les supports en mode connexion (COMEDIA) [RFC4145]. La motivation première était de permettre aux sessions MSRP d'aboutir dans des situations où l'offreur ne pouvait pas accepter de connexions mais où celui qui répond le pouvait. Par exemple, l'offreur pourrait être derrière un NAT, tandis que celui qui répond pourrait avoir une adresse acheminable mondialement.

Le groupe de travail SIMPLE a choisi de supprimer ce mécanisme de MSRP, car il ajoutait beaucoup de complexité à la gestion des connexions. À la place, MSRP spécifie maintenant une direction de connexion par défaut. La partie qui a envoyé l'offre initiale est responsable de la connexion avec son homologue.

8.6 Négociation de type de contenu

Une ligne de supports SDP proposant MSRP DOIT être accompagnée d'un attribut accept-types.

Une entrée de "*" dans l'attribut accept-types indique que l'envoyeur peut tenter d'envoyer du contenu avec des types de supports qui n'ont pas été explicitement mentionnés. De même, une entrée avec un type explicite et un caractère "*" comme sous-type indique que l'envoyeur peut tenter d'envoyer du contenu avec n'importe quel sous-type de ce type. Si le receveur reçoit une demande MSRP et est capable de traiter le type de supports, il le fait. Sinon, il va faire une réponse 415. Noter que toutes les entrées explicites DEVRAIENT être considérées comme préférées à tout type non mentionné. Cette caractéristique est nécessaire car autrement, la liste des formats pour les appareils de messagerie instantanée enrichie pourrait être prohibitive.

La présente spécification exige la prise en charge de certains formats de données. Les formats obligatoires DOIVENT être signalés comme tous les autres, soit explicitement, soit par l'utilisation d'un "*".

L'attribut accept-types peut inclure des types de conteneurs, c'est-à-dire des formats MIME qui contiennent d'autres types en interne. Si des types composés sont utilisés, les types énumérés dans l'attribut accept-types peuvent être utilisés comme charge utile racine ou être enveloppés dans un type de conteneur énuméré. Tout type de conteneur DOIT également être mentionné dans l'attribut accept-types.

À l'occasion, un point d'extrémité va devoir spécifier un type de support MIME qui ne peut être utilisé que si il est enveloppé dans un type de conteneur mentionné.

Les points d'extrémité PEUVENT spécifier des types de supports qui ne sont autorisés que lorsque ils sont enveloppés dans des types composés à l'aide de l'attribut "accept-wrapped-types" dans une ligne a SDP.

La sémantique de l'attribut accept-wrapped-types est identique à celle de l'attribut accept-types, à l'exception du fait que les types spécifiés ne peuvent être utilisés que lorsqu'ils sont enveloppés dans des types de conteneurs énumérés dans l'attribut accept-types. Seuls les types énumérés dans l'attribut accept-types peuvent être utilisés comme type "racine" pour l'ensemble du corps. Comme tout type énuméré dans accept-types peut être à la fois utilisé comme corps racine et enveloppé dans d'autres corps, les entrées de format accept-types NE DOIVENT PAS être répétées dans cet attribut.

Cette approche ne permet pas de spécifier des listes distinctes de types enveloppés acceptables pour différents types de conteneurs. Si un point d'extrémité comprend un type de supports dans le contexte d'un conteneur, il est supposé le comprendre dans le contexte de tous les autres conteneurs acceptables, sous réserve des contraintes définies par les types de conteneurs eux-mêmes.

L'approche consistant à spécifier les types qui ne sont autorisés qu'à l'intérieur des conteneurs, séparément des types de charge utile principaux, permet à un point d'extrémité de forcer l'utilisation de certaines enveloppes. Par exemple, un appareil de passerelle de présence commune et de messagerie instantanée (CPIM, *Common Presence and Instant Messaging*) [RFC3862] peut exiger que tous les messages soient enveloppés dans des corps message/cpim, mais peut autoriser plusieurs types de contenu à l'intérieur de l'enveloppe. Si la passerelle devait spécifier les types enveloppés dans l'attribut accept-types, son homologue pourrait tenter d'utiliser ces types sans l'enveloppe.

Si le receveur d'une offre ne comprend aucun des types de charge utile indiqués dans le SDP offert, il DEVRAIT l'indiquer en utilisant le mécanisme approprié du protocole de rendez-vous. Par exemple, dans SIP, il DEVRAIT retourner une réponse SIP 488.

Un point d'extrémité MSRP NE DOIT PAS envoyer de contenu d'un type non signalé par l'homologue dans un attribut accept-types ou accept-wrapped-types. En outre, il NE DOIT PAS envoyer un document MIME de premier niveau (c'est-à-dire non enveloppé) d'un type non signalé dans l'attribut accept-types. Dans les deux cas, la signalisation peut être explicite ou implicite par l'utilisation du caractère "*".

Un point d'extrémité PEUT indiquer la taille maximale de message qu'il souhaite recevoir à l'aide de l'attribut de ligne a max-size. La taille maximale se réfère au message complet en octets, et non à la taille d'un tronçon particulier. Les envoyeurs NE DEVRAIENT PAS dépasser la limite de taille maximale pour tout message envoyé dans la session résultante. Cependant, le receveur devrait considérer la valeur max-size comme une indication.

Les entrées de format de support peuvent inclure des paramètres. L'interprétation de ces paramètres varie d'un type de support à l'autre. Pour les besoins de la négociation du type de supports, une entrée de format avec un ou plusieurs paramètres est supposée correspondre à la même entrée de format sans paramètres.

La syntaxe formelle de ces attributs est la suivante :

```
accept-types = accept-types-label ":" format-list
accept-types-label = "accept-types"
accept-wrapped-types = wrapped-types-label ":" format-list
wrapped-types-label = "accept-wrapped-types"
format-list = format-entry *( SP format-entry)
format-entry = ( ( (type "/" subtype) / (type "/" "*") ) *( ";" type-param ) ) / ("*")
type = token
subtype = token
```

```
type-param = parm-attribute "=" parm-value
parm-attribute = token
parm-value = token / quoted-string
max-size = max-size-label ":" max-size-value
max-size-label = "max-size"
max-size-value = 1*(DIGIT) ; max size in octets
```

Figure 8 : Syntaxe d'attribut

8.7 Exemple d'échange SDP

Le point d'extrémité A souhaite inviter le point d'extrémité B à une session MSRP. A offre la description de session suivante :

```
v=0
o=usera 2890844526 2890844527 IN IP4 alice.exemple.com
s= -
c=IN IP4 alice.exemple.com
t=0 0
m=message 7394 TCP/MSRP *
a=accept-types:message/cpim text/plain text/html
a=path:msrp://alice.exemple.com:7394/2s93i93idj;tcp
```

Figure 9 : SDP du point d'extrémité A

B répond avec son propre URI :

```
v=0
o=userb 2890844530 2890844532 IN IP4 bob.exemple.com
s= -
c=IN IP4 bob.exemple.com
t=0 0
m=message 8493 TCP/MSRP *
a=accept-types:message/cpim text/plain
a=path:msrp://bob.exemple.com:8493/si438dsaodes;tcp
```

Figure 10 : SDP du point d'extrémité B

8.8 Expérience d'utilisateur MSRP avec SIP

Dans les applications SIP normales, quand un point d'extrémité reçoit une demande INVITE, il alerte l'utilisateur, et attend une entrée de l'utilisateur avant de répondre. Ceci est analogue au ressenti normal de l'utilisateur du téléphone, où l'appelé "répond" à l'appel.

En revanche, dans le ressenti normal de l'utilisateur des applications de messagerie instantanée, le message initial reçu est immédiatement affiché à l'utilisateur, sans attendre qu'il "rejoigne" la conversation. Par conséquent, le principe de moindre surprise suggère que les points d'extrémité MSRP utilisant la signalisation SIP DEVRAIENT permettre un mode dans lequel le point d'extrémité accepte tranquillement la session et commence à afficher les messages.

Cette ligne directrice peut ne pas avoir de sens dans toutes les situations, par exemple dans le cas d'applications de supports mixtes, où des sessions MSRP et audio sont proposées dans le même INVITE. En général, une bonne conception de l'application devrait primer.

Les demandes SIP INVITE peuvent être fourchées par un mandataire SIP, ce qui fait que plus d'un point d'extrémité reçoit le même INVITE. Les techniques SIP de support précoce [RFC3960] peuvent être utilisées pour établir une session préliminaire avec chaque point d'extrémité afin que le ou les messages initiaux soient affichés sur chaque point d'extrémité, et pour annuler la transaction INVITE pour tous les points d'extrémité qui n'envoient pas de trafic MSRP après un certain temps, de sorte qu'ils cessent de recevoir du trafic MSRP de l'invitant.

8.9 Attribut Direction SDP et MSRP

SDP définit un certain nombre d'attributs qui modifient la direction des flux de supports. Il s'agit des attributs "sendonly", "recvonly", "inactive" et "sendrecv".

Si un attribut "sendonly" ou "recvonly" modifie une ligne de description de support MSRP, l'attribut indique la direction des demandes MSRP SEND qui contiennent des charges utiles de messages ordinaires. Sauf indication contraire, ces attributs n'affectent pas la direction des autres types de demandes, comme REPORT. Les demandes SEND qui contiennent un protocole de contrôle ou de rapport plutôt qu'une charge utile de message ordinaire (par exemple, les rapports de notification de livraison de message instantané (IMDN, *Instant Message Delivery Notification*) devraient être générées conformément aux règles du protocole, comme si aucun attribut de direction n'était présent.

9. Syntaxe formelle

MSRP est un protocole de texte qui utilise le format de transformation UTF-8 [RFC3629].

La spécification de syntaxe suivante utilise la forme Backus-Naur (BNF) augmentée décrite dans la [RFC4234].

```
msrp-req-or-resp = msrp-request / msrp-response
msrp-request = req-start headers [content-stuff] end-line
msrp-response = resp-start headers end-line

req-start  = pMSRP SP transact-id SP method CRLF
resp-start = pMSRP SP transact-id SP status-code [SP comment] CRLF
comment = utf8text

pMSRP = %x4D.53.52.50 ; MSRP in caps
transact-id = ident
method = mSEND / mREPORT / other-method
mSEND = %x53.45.4e.44 ; SEND in caps
mREPORT = %x52.45.50.4f.52.54; REPORT in caps

other-method = 1*UPALPHA
status-code = 3DIGIT                    ; tout code défini dans le présent document ou un document d'extension.

MSRP-URI = msrp-scheme "://" authority ["/" session-id] ";" transport *( ";" URI-parameter)
                                        ; authority est défini dans la RFC3986

msrp-scheme = "msrp" / "msrps"
session-id = 1*( unreserved / "+" / "=" / "/" )             ; unreserved est défini dans la RFC3986
transport = "tcp" / 1*ALPHANUM
URI-parameter = token ["=" token]

headers = To-Path CRLF From-Path CRLF 1*( header CRLF )

header  = Message-ID / Success-Report / Failure-Report / Byte-Range / Status / ext-header

To-Path = "To-Path:" SP MSRP-URI *( SP MSRP-URI )
From-Path = "From-Path:" SP MSRP-URI *( SP MSRP-URI )
Message-ID = "Message-ID:" SP ident
Success-Report = "Success-Report:" SP ("oui" / "non" )
Failure-Report = "Failure-Report:" SP ("oui" / "non" / "partiel" )
Byte-Range = "Byte-Range:" SP range-start "-" range-end "/" total
range-start = 1*DIGIT
range-end = 1*DIGIT / "*"
total = 1*DIGIT / "*"

Status = "Status:" SP namespace SP status-code [SP comment]
namespace = 3(DIGIT)                    ; "000" pour tous les codes définis dans le présent document.
```

```
ident = ALPHANUM  3*31ident-char
ident-char = ALPHANUM / "." / "-" / "+" / "%" / "="

content-stuff = *(Other-Mime-header CRLF)
                Content-Type 2CRLF data CRLF

Content-Type = "Content-Type:" SP media-type
media-type = type "/" subtype *( ";" gen-param )
type = token
subtype = token

gen-param = pname [ "=" pval ]
pname = token
pval  = token / quoted-string

token = 1*(%x21 / %x23-27 / %x2A-2B / %x2D-2E / %x30-39 / %x41-5A / %x5E-7E)
                                        ; la comparaison de token est insensible à la casse.

quoted-string = DQUOTE *(qdtext / qd-esc) DQUOTE
qdtext = SP / HTAB / %x21 / %x23-5B / %x5D-7E / UTF8-NONASCII
qd-esc = (BACKSLASH BACKSLASH) / (BACKSLASH DQUOTE)
BACKSLASH = "\"
UPALPHA  = %x41-5A
ALPHANUM = ALPHA / DIGIT

Other-Mime-header = (Content-ID / Content-Description / Content-Disposition / mime-extension-field)
        ; Content-ID, et Content-Description sont définis dans la RFC2045.
        ; Content-Disposition est défini dans la RFC2183
        ; MIME-extension-field indique des champs d'en-tête d'extension MIME supplémentaires décrits dans la RFC2045

data = *OCTET
end-line = "-------" transact-id continuation-flag CRLF
continuation-flag = "+" / "$" / "#"

ext-header = hname ":" SP hval
hname = ALPHA *token
hval = utf8text

utf8text = *(HTAB / %x20-7E / UTF8-NONASCII)

UTF8-NONASCII = %xC0-DF 1UTF8-CONT / %xE0-EF 2UTF8-CONT / %xF0-F7 3UTF8-CONT
                        / %xF8-Fb 4UTF8-CONT / %xFC-FD 5UTF8-CONT
UTF8-CONT     = %x80-BF
```

Figure 11 : ABNF MSRP

10. Description des codes de réponse

Cette section résume la sémantique des différents codes de réponse qui peuvent être utilisés dans les réponses aux transactions MSRP. Ces codes peuvent aussi être utilisés dans le champ d'en-tête État des demandes REPORT.

10.1 200

Le code de réponse 200 indique une transaction réussie.

10.2 400

Une réponse 400 indique qu'une demande est incompréhensible. L'envoyeur peut réessayer la demande après la correction

de l'erreur.

10.3 403

Une réponse 403 indique que l'action tentée n'est pas permise. L'envoyeur ne devrait pas réessayer la demande.

10.4 408

Une réponse 408 indique qu'une transaction en aval ne s'est pas achevée dans le délai imparti. Elle n'est jamais envoyée par des éléments décrits dans la présente spécification. Cependant, 408 est utilisé dans les extensions de relais MRSP ; donc, les points d'extrémité MSRP peuvent la recevoir. Un point d'extrémité DOIT traiter une réponse 408 de la même manière qu'une fin de temporisation locale.

10.5 413

Une réponse 413 indique que le receveur souhaite que l'envoyeur cesse d'envoyer ce message particulier. Normalement, une réponse 413 est envoyée en réponse à un tronçon d'un message non désiré.

Si l'envoyeur d'un message reçoit un 413 dans une réponse ou dans une demande REPORT, il NE DOIT PAS envoyer d'autres tronçons du message, c'est-à-dire d'autres tronçons ayant la même valeur d'identifiant de message. Si l'envoyeur reçoit le 413 alors qu'il est en train d'envoyer un tronçon et que ce tronçon peut être interrompu, il DOIT l'interrompre.

10.6 415

Une réponse 415 indique que la demande SEND contenait un type de support qui n'est pas compris du receveur. L'envoyeur ne devrait pas envoyer d'autres messages avec le même type de contenu pour la durée de la session.

10.7 423

Une réponse 423 indique qu'un des paramètres demandés est hors limites. Elle est utilisée par les extensions de relais au présent document.

10.8 481

Une réponse 481 indique que la session indiquée n'existe pas. L'envoyeur devrait terminer la session.

10.9 501

Une réponse 501 indique que le destinataire ne comprend pas la méthode de demande.

Le code de réponse 501 existe pour permettre un certain degré d'extensibilité des méthodes. Il ne s'agit pas d'une autorisation d'ignorer les méthodes définies dans le présent document, mais plutôt d'un mécanisme permettant de signaler l'absence de prise en charge des méthodes d'extension.

10.10 506

Une réponse 506 indique qu'une demande est arrivée sur une session qui est déjà liée à une autre connexion réseau. L'envoyeur devrait cesser d'envoyer des messages pour cette session sur cette connexion.

11. Exemples

11.1 Session IM de base

Cette Section montre un exemple de flux pour le scénario le plus courant. L'exemple suppose que SIP est utilisé pour transporter l'échange SDP. Les détails des messages SIP et de l'infrastructure de mandataire SIP sont omis par souci de concision. Dans l'exemple, on suppose que l'offreur est sip:alice@exemple.com et que celui qui répond est

sip:bob@exemple.com.

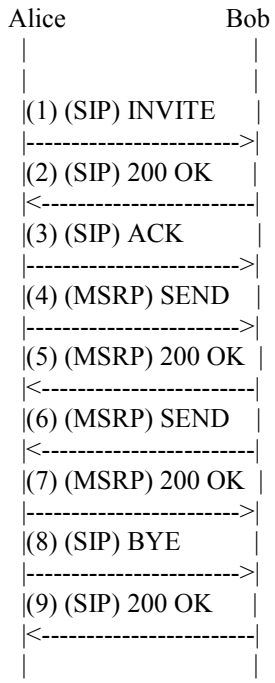

Figure 12 :Exemple de session IM de base

1. Alice construit un URI local de msrp://alicepc.exemple.com:7777/iau39soe2843z;tcp .

Alice->Bob (SIP) : INVITE sip:bob@exemple.com

```
v=0
o=alice 2890844557 2890844559 IN IP4 alicepc.exemple.com
s= -
c=IN IP4 alicepc.exemple.com
t=0 0
m=message 7777 TCP/MSRP *
a=accept-types:text/plain
a=path:msrp://alicepc.exemple.com:7777/iau39soe2843z;tcp
```

2. Bob écoute sur l'accès 8888, et envoie la réponse suivante :

Bob->Alice (SIP) : 200 OK

```
v=0
o=bob 2890844612 2890844616 IN IP4 bob.exemple.com
s= -
c=IN IP4 bob.exemple.com
t=0 0
m=message 8888 TCP/MSRP *
a=accept-types:text/plain
a=path:msrp://bob.exemple.com:8888/9di4eae923wzd;tcp
```

3. Alice->Bob (SIP): ACK sip:bob@exemple.com

4. (Alice ouvre une connexion avec Bob.) Alice->Bob (MSRP):

```
MSRP d93kswow SEND
To-Path: msrp://bob.exemple.com:8888/9di4eae923wzd;tcp
From-Path: msrp://alicepc.exemple.com:7777/iau39soe2843z;tcp
Message-ID: 12339sdqwer
```

Byte-Range: 1-16/16
Content-Type: text/plain

Hé, c'est Alice !
-------d93kswow$

5. Bob->Alice (MSRP):

MSRP d93kswow 200 OK
To-Path: msrp://alicepc.exemple.com:7777/iau39soe2843z;tcp
From-Path: msrp://bob.exemple.com:8888/9di4eae923wzd;tcp
-------d93kswow$

6. Bob->Alice (MSRP):

MSRP dkei38sd SEND
To-Path: msrp://alicepc.exemple.com:7777/iau39soe2843z;tcp
From-Path: msrp://bob.exemple.com:8888/9di4eae923wzd;tcp
Message-ID: 456s9wlk3
Byte-Range: 1-21/21
Content-Type: text/plain

Hé, Alice ! C'est Bob !
-------dkei38sd$

7. Alice->Bob (MSRP):

MSRP dkei38sd 200 OK
To-Path: msrp://bob.exemple.com:8888/9di4eae923wzd;tcp
From-Path: msrp://alicepc.exemple.com:7777/iau39soe2843z;tcp
-------dkei38sd$

8. Alice->Bob (SIP): BYE sip:bob@example.com

Alice invalide l'état de session local.

9. Bob invalide l'état local pour la session.

Bob->Alice (SIP): 200 OK

11.2 Message avec contenu XHTML

```
MSRP dsdfoe38sd SEND
To-Path: msrp://alice.exemple.com:7777/iau39soe2843z;tcp
From-Path: msrp://bob.exemple.com:8888/9di4eae923wzd;tcp
Message-ID: 456so39s
Byte-Range: 1-374/374
Content-Type: application/xhtml+xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html
PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"_http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd_">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
  <title>FY2005 Results</title>
</head>
 <body>
  <p>See the results at <a
href="http://example.org/">example.org</a>.</p>
 </body>
```

```
</html>
-------dsdfoe38sd$
```

Figure 13 : Exemple de message avec XHTML

11.3 Message tronçonné

Pour un exemple de message tronçonné, voir l'exemple du paragraphe 5.1.

11.4 Message tronçonné avec charge utile Message/CPIM

Cet exemple montre un message tronçonné contenant un message CPIM qui enveloppe une charge utile text/plain. Noter que MSRP prend en compte l'intégralité du message CPIM avant de le découper en tronçons ; les en-têtes CPIM ne sont donc inclus que dans le premier tronçon. Les en-têtes MSRP Content-Type et Byte-Range, présents dans les deux tronçons, font référence à l'ensemble du message CPIM.

```
MSRP d93kswow SEND
To-Path: msrp://bobpc.exemple.com:8888/9di4eae923wzd;tcp
From-Path: msrp://alicepc.exemple.com:7654/iau39soe2843z;tcp
Message-ID: 12339sdqwer
Byte-Range: 1-137/148
Content-Type: message/cpim

To: Bob <sip:bob@exemple.com>
From: Alice <sip:alice@exemple.com>
DateTime: 2006-05-15T15:02:31-03:00
Content-Type: text/plain

ABCD
-------d93kswow+
```

Figure 14 : Premier tronçon

Alice envoie le second et dernier tronçon.

```
MSRP op2nc9a SEND
To-Path: msrp://bobpc.exemple.com:8888/9di4eae923wzd;tcp
From-Path: msrp://alicepc.exemple.com:7654/iau39soe2843z;tcp
Message-ID: 12339sdqwer
Byte-Range: 138-148/148
Content-Type: message/cpim

1234567890
-------op2nc9a$
```

Figure 15 : Second tronçon

11.5 Message System

Sysadmin->Alice (MSRP):

```
MSRP d93kswow SEND
To-Path: msrp://alicepc.exemple.com:8888/9di4eae923wzd;tcp
From-Path: msrp://exemple.com:7777/iau39soe2843z;tcp
Message-ID: 12339sdqwer
Byte-Range: 1-38/38
Failure-Report: no
Success-Report: no
Content-Type: text/plain
```

Cette conférence va se terminer dans 5 minutes
-------d93kswow$

11.6 Rapport positif

Alice->Bob (MSRP) :

```
MSRP d93kswow SEND
To-Path: msrp://bob.exemple.com:8888/9di4eae923wzd;tcp
From-Path: msrp://alicepc.exemple.com:7777/iau39soe2843z;tcp
Message-ID: 12339sdqwer
Byte-Range: 1-106/106
Success-Report: oui
Failure-Report: non
Content-Type: text/html

<html><body>
<p>Voici ce lien important ...
<a href="http://www.exemple.com/foobar">foobar</a>
</p>
</body></html>
-------d93kswow$
```

Figure 16 : Demande initiale SEND

Bob->Alice (MSRP) :

```
MSRP dkei38sd REPORT
To-Path: msrp://alicepc.exemple.com:7777/iau39soe2843z;tcp
From-Path: msrp://bob.exemple.com:8888/9di4eae923wzd;tcp
Message-ID: 12339sdqwer
Byte-Range: 1-106/106
Status: 000 200 OK
-------dkei38sd$
```

Figure 17 : Rapport de succès

11.7 IM fourché

Les systèmes de messagerie instantanée traditionnels gèrent généralement mal la présence simultanée de plusieurs clients de messagerie instantanée pour la même personne. Bien que certains d'entre eux soient plus performants que de nombreux systèmes existants, la gestion des clients multiples est assez rudimentaire. Ce problème devient beaucoup plus important lorsque des appareils mobiles toujours actifs sont disponibles, mais il est souhaitable de les utiliser uniquement si un autre client de messagerie instantanée n'est pas disponible.

L'utilisation de SIP rend les décisions de rendez-vous explicites, déterministes et très souples. En revanche, les systèmes de messagerie instantanée en mode "page" utilisent des décisions implicites spécifiques à la mise en œuvre que les clients de messagerie instantanée ne peuvent pas influencer. Avec la messagerie SIP en mode session, les décisions de rendez-vous peuvent être contrôlées par le client d'une manière prévisible et interopérable pour tout hôte qui met en œuvre des capacités d'appelant [RFC3840]. Par conséquent, la politique de rendez-vous est gérée de manière cohérente pour chaque adresse d'enregistrement.

L'exemple suivant montre Juliette avec plusieurs clients de messagerie instantanée auxquels elle peut être jointe. Chacun d'eux possède un contact SIP et une session MSRP uniques. L'exemple tire parti de la capacité de SIP à "fourcher" une invitation à plusieurs contacts en parallèle, en séquence, ou en combinaison. Juliette s'est enregistrée depuis sa chambre, le balcon, son PDA et, en dernier recours, on peut laisser un message à sa nurse. Les contacts de Juliette sont énumérés ci-dessous. Les valeurs q expriment la préférence relative (q=1.0 est la préférence la plus élevée).

Lorsque Roméo ouvre son programme de messagerie instantanée, il sélectionne Juliette et tape le message "es-tu céant ?" (au lieu de "tu es là ?"). Son client envoie une invitation SIP à sip:juliet@thecapulets.exemple.com. Le mandataire essaie d'abord le balcon et la chambre simultanément. Un client s'exécute sur chacun de ces systèmes, qui établissent tous deux des sessions précoces de MSRP avec le client de Roméo. Le client envoie automatiquement le message sur MSRP aux deux URI MSRP concernés. Après un délai de plusieurs secondes sans réponse ni activité de la part de Juliette, le mandataire annule l'invitation à ses deux premiers contacts et transmet l'invitation à l'assistant numérique de Juliette. Comme son père lui parle de son mariage, elle sélectionne l'option "Ne pas déranger" sur son PDA, qui envoie une réponse "Occupé ici". Le mandataire essaie alors de joindre la nurse, qui répond et explique à Roméo ce qui se passe.

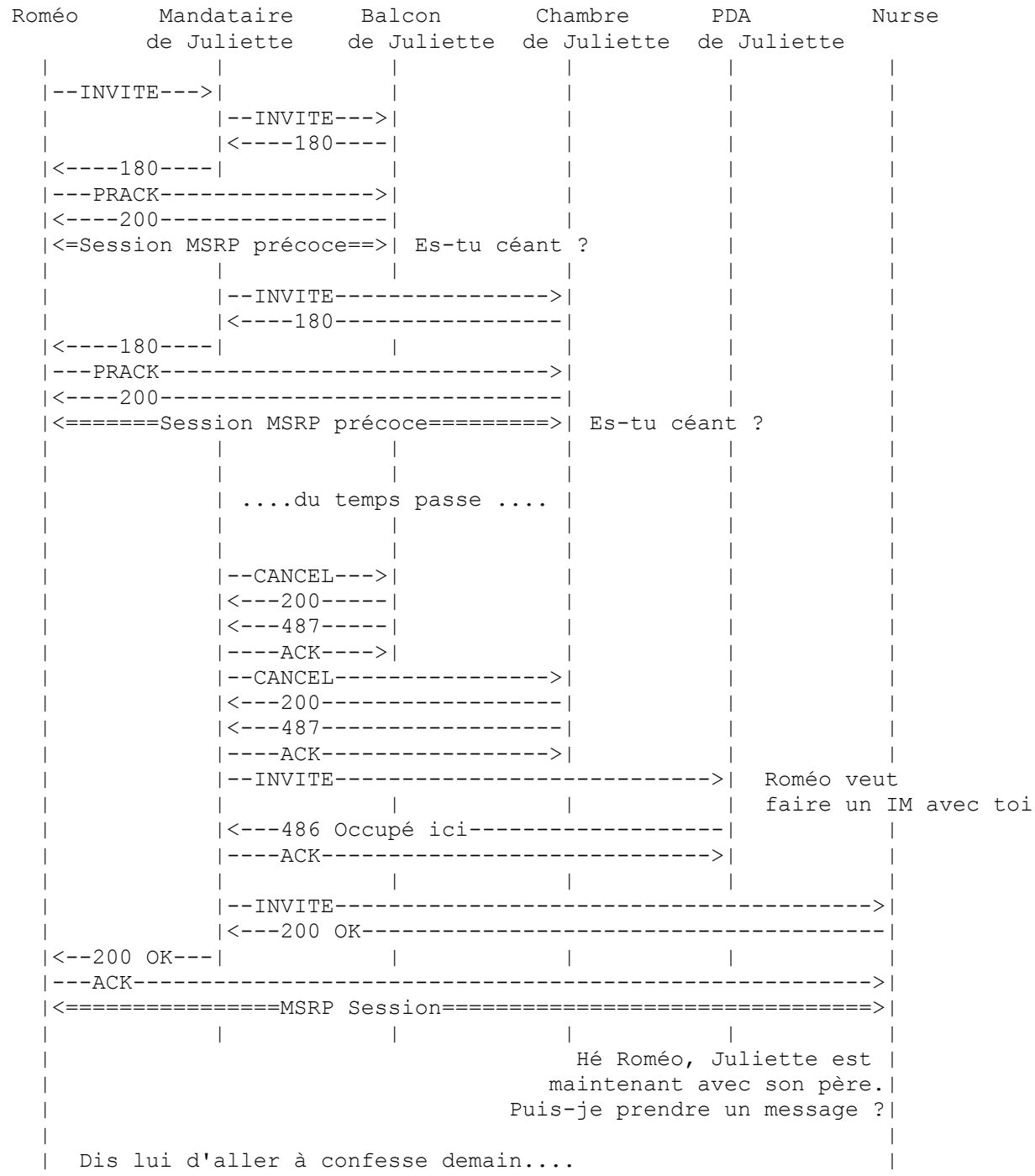

Figure 18 : Exemple de fourchement

12. Extensibilité

MSRP a été conçu pour n'être que très peu extensible. De nouvelles méthodes MSRP, de nouveaux champs d'en-tête et de nouveaux codes d'état peuvent être définis dans des RFC sur la voie de la normalisation. MSRP ne contient pas de numéro de version ni de mécanisme de négociation pour exiger ou découvrir de nouvelles fonctionnalités. Si une extension nécessitant une négociation est spécifiée à l'avenir, la spécification devra décrire comment l'extension doit être négociée dans le protocole de signalisation encapsulant. Si une mise à jour ou une extension non interopérable se produit à l'avenir, elle sera traitée comme un nouveau protocole et devra décrire comment son utilisation sera signalée.

Afin d'autoriser les champs d'en-tête d'extension sans rompre l'interopérabilité, si un appareil MSRP reçoit une demande ou une réponse contenant un champ d'en-tête qu'il ne comprend pas, il DOIT ignorer le champ d'en-tête et traiter la demande ou la réponse comme si le champ d'en-tête n'était pas présent. Si un appareil MSRP reçoit une demande avec une méthode inconnue, il DOIT renvoyer une réponse 501.

MSRP a été conçu pour utiliser des listes d'URI au lieu d'un seul URI dans les champs d'en-tête To-Path et From-Path en prévision de l'ajout d'une fonctionnalité de relais ou de passerelle. En outre, les URI "msrp" et "msrps" peuvent contenir des paramètres qui sont extensibles.

13. Compatibilité CPIM

Les sessions MSRP peuvent passer par une passerelle vers d'autres protocoles compatibles avec le profil commun pour la messagerie instantanée (CPIM, *Common Profile for Instant Messaging*) [RFC3860]. Si cela se produit, la passerelle DOIT conserver l'état de session et traduire la sémantique de la session MSRP en sémantique CPIM, qui ne comprend pas de concept de session. En outre, lorsque l'un des points d'extrémité de la session est une passerelle CPIM, les messages instantanés DEVRAIENT être enveloppés dans des corps "message/cpim" [RFC3862]. Une telle passerelle DOIT inclure "message/cpim" comme première entrée de son attribut SDP accept-types. Les points d'extrémité MSRP qui envoient des messages instantanés à un homologue ayant inclus "message/cpim" comme première entrée de l'attribut accept-types DEVRAIENT encapsuler tous les corps de messages instantanés dans des enveloppes "message/cpim". Tous les points d'extrémité MSRP DOIVENT prendre en charge le type message/cpim et DEVRAIENT prendre en charge les caractéristiques S/MIME [RFC3851] de ce format.

Si un message doit être enveloppé dans une enveloppe message/cpim, l'enveloppement DOIT être effectué avant de diviser le message en tronçons, si nécessaire.

Tous les points d'extrémité MSRP DOIVENT reconnaître les champs d'en-tête From, To, DateTime et Require tels que définis dans la RFC 3862. Ces applications DEVRAIENT reconnaître le champ d'en-tête CC et PEUVENT reconnaître le champ d'en-tête Subject. Toute application MSRP qui reconnaît un champ d'en-tête message/cpim DOIT comprendre le champ d'en-tête NS (espace de noms).

Toutes les parties de corps message/cpim envoyées par un point d'extrémité MSRP DOIVENT inclure les champs d'en-tête From et To. Si la partie de corps message/cpim est protégée par S/MIME, elle DOIT aussi inclure le champ d'en-tête DateTime.

Les champs d'en-tête NS, To et CC peuvent apparaître plusieurs fois. Les autres champs d'en-tête définis dans la RFC 3862 NE DOIVENT PAS apparaître plus d'une fois dans une partie de corps message/cpim d'un message MSRP. Le champ d'en-tête Require PEUT comporter plusieurs valeurs. Le champ d'en-tête NS PEUT apparaître zéro, une ou plusieurs fois, selon le nombre d'espaces de noms référencés.

Les champs d'en-tête d'extension PEUVENT apparaître plus d'une fois, selon la définition de ces champs d'en-tête.

L'utilisation d'enveloppes message/cpim est aussi utile si un appareil MSRP souhaite envoyer un message au nom d'une autre identité. L'appareil peut ajouter une enveloppe message/cpim avec la valeur appropriée de champ d'en-tête From.

14. Considérations sur la sécurité

Les systèmes de messagerie instantanée sont utilisés pour échanger toute une série d'informations sensibles, qu'il s'agisse de

conversations personnelles, d'informations confidentielles d'entreprises, de numéros de compte ou d'autres informations sur les transactions financières. La messagerie instantanée est utilisée par les particuliers, les entreprises et les gouvernements pour communiquer des informations importantes. Les systèmes de messagerie instantanée doivent garantir l'intégrité et la confidentialité des informations échangées, la certitude de communiquer avec le bon interlocuteur et la possibilité de communiquer de manière anonyme. MSRP repousse une grande partie des problèmes difficiles vers SIP lorsque ce dernier établit la session, mais certains problèmes subsistent. Les pourriels et les attaques de déni de service (DoS) sont également très importants pour les systèmes de messagerie instantanée.

MSRP doit assurer la confidentialité et l'intégrité des messages qu'il transfère. Il doit aussi garantir que l'hôte connecté est bien celui auquel il voulait se connecter et que la connexion n'a pas été détournée.

14.1 Secret de l'URI MSRP

Lorsqu'un point d'extrémité envoie un URI MSRP à son homologue dans un protocole de rendez-vous, cet URI est en fait un secret partagé entre les homologues. Si un attaquant apprend ou devine l'URI avant la fin de l'établissement de la session, il peut être en mesure d'usurper l'identité de l'un des homologues.

En supposant que l'échange d'URI dans le protocole de rendez-vous soit suffisamment protégé, il est essentiel que l'URI reste difficile à "deviner" par des méthodes de force brute. La plupart des composants de l'URI, tels que le schéma et les composants d'autorité, sont connus de tous. Le secret est entièrement assuré par le composant d'identifiant de session.

Par conséquent, lorsqu'un appareil MSRP génère un URI MSRP à utiliser lors de l'ouverture d'une session MSRP, le composant "identifiant de session" DOIT contenir au moins 80 bits d'aléa.

14.2 Protection au niveau du transport

Quand on utilise uniquement des connexions TCP, la sécurité MSRP est assez faible. Si l'hôte A contacte l'hôte B, B transmet son nom d'hôte et un secret à A à l'aide d'un protocole de rendez-vous. Bien que MSRP exige l'utilisation d'un protocole de rendez-vous capable de protéger cet échange, il n'y a aucune garantie que la protection soit utilisée en permanence. Si une telle protection n'est pas utilisée, n'importe qui peut voir ce secret. L'hôte A se connecte alors au nom d'hôte fourni et transmet le secret en clair à B à travers la connexion. L'hôte A suppose qu'il parle à B en se fondant sur l'endroit où il a envoyé le paquet SYN et transmet ensuite le secret en clair à travers les connexions. L'hôte B suppose qu'il parle à A parce que l'hôte à l'autre bout de la connexion a livré le secret. Un attaquant capable d'acquitter le paquet SYN pourrait s'insérer en tant qu'interposé dans la connexion.

L'utilisation de connexions TLS améliore considérablement la sécurité. On suppose que l'hôte qui accepte la connexion dispose d'un certificat délivré par une autorité de certification bien connue. En outre, on suppose que la signalisation pour établir la session est protégée par le protocole de rendez-vous. Dans ce cas, quand l'hôte A contacte l'hôte B, le secret est passé par un canal confidentiel à A. A se connecte à B avec TLS. B présente un certificat valide, de sorte que A sait qu'il est réellement connecté à B. A livre ensuite le secret fourni par B, afin que B puisse vérifier qu'il est connecté à A. Dans ce cas, un mandataire SIP malhonnête peut voir le secret dans le trafic de signalisation SIP et pourrait potentiellement s'insérer en tant qu'interposé.

En réalité, l'utilisation de TLS avec des certificats provenant d'autorités de certification bien connues est difficile pour les connexions d'homologue à homologue, car les types d'hôtes que les clients finaux utilisent pour envoyer des messages instantanés ont peu de chances d'avoir des adresses IP ou des noms DNS stables à long terme auxquels les certificats peuvent se lier. En outre, le coût des certificats de serveur émanant d'autorités de certification bien connues est actuellement suffisamment élevé pour décourager leur utilisation pour chaque client. L'utilisation de TLS en mode d'homologue à homologue sans certificats bien connus est abordée au paragaphe 14.4.

TLS devient beaucoup plus pratique lorsqu'une forme de relais est introduite. Les clients peuvent alors établir des connexions TLS avec les relais, qui sont beaucoup plus susceptibles d'avoir des certificats TLS. Bien que cette spécification ne traite pas de ces relais, ils sont décrits dans un document complémentaire [RFC4976]. Ce document fait un usage intensif de TLS pour protéger le trafic entre les clients et les relais, et entre les relais.

TLS est utilisé pour authentifier les appareils et pour assurer l'intégrité et la confidentialité des champs d'en-tête transportés. Les éléments MSRP DOIVENT mettre en œuvre TLS et également les informations de hello TLS étendu ClientExtendedHello pour l'indication du nom du serveur, comme décrit dans la [RFC4366]. La suite de chiffrement TLS de TLS_RSA_WITH_AES_128_CBC_SHA [RFC3268] DOIT être prise en charge (d'autres suites de chiffrement

PEUVENT également être prises en charge).

14.3 S/MIME

La seule sécurité forte pour les connexions non TLS est assurée par S/MIME.

Comme MSRP transporte un contenu MIME arbitraire, il peut trivialement transporter aussi des messages protégés par S/MIME. Toutes les mises en œuvre MSRP DOIVENT prendre en charge le type de support multipart/signé même si elles ne prennent pas en charge S/MIME. Comme SIP peut transporter une clé de session, les messages S/MIME dans le contexte d'une session pourraient également être protégés à l'aide d'un secret partagé à clé enveloppée [RFC3217] fourni lors de l'établissement de la session. MSRP peut transporter des charges utiles binaires non codées. Par conséquent, les corps MIME DOIVENT être transférés avec un codage de transfert binaire. Si un message est à la fois signé et chiffré, il DEVRAIT être signé d'abord, puis chiffré. Si S/MIME est pris en charge, SHA-1, SHA-256, RSA et AES-128 DOIVENT être pris en charge. Pour RSA, les mises en œuvre DOIVENT prendre en charge des tailles de clés d'au moins 1024 bits et DEVRAIENT prendre en charge des tailles de clés de 2048 bits ou plus.

Il n'est pas nécessaire pour cela que le point d'extrémité dispose de certificats d'une autorité de certification bien connue. Quand MSRP est utilisé avec SIP, les mécanismes Identity [RFC4474] et Certificates [RFC6072] fournissent une livraison fondée sur S/MIME d'un secret entre A et B. Aucun intermédiaire SIP, à l'exception du service d'authentification explicitement de confiance (un par utilisateur) ne peut voir le secret. Le chiffrement S/MIME du SDP peut aussi être utilisé par SIP pour échanger du matériel de chiffement qui peut être utilisé dans MSRP.

La session MSRP peut alors utiliser S/MIME avec ce matériel de chiffement pour signer et chiffrer les messages envoyés sur MSRP. La connexion peut toujours être détournée puisque le secret est envoyé en clair à l'autre extrémité de la connexion TCP, mais les conséquences sont atténuées si tout le contenu MSRP est signé et chiffré avec S/MIME. Bien que cela sorte du domaine d'application du présent document, la négociation SIP d'une session MSRP peut négocier du matériel de chiffement symétrique à utiliser avec S/MIME pour l'intégrité et la confidentialité.

14.4 Utilisation de TLS en mode d'homologue à homologue

TLS peut être utilisé avec un certificat auto-signé à condition qu'il existe un mécanisme permettant aux deux parties de s'assurer que l'autre partie a utilisé le certificat correct. Quand il est utilisé avec SDP et SIP, le certificat correct peut être vérifié en transmettant une empreinte digitale du certificat dans le SDP et en s'assurant que le SDP dispose d'une protection d'intégrité appropriée. Quand SIP est utilisé pour transporter le SDP, l'intégrité peut être fournie par le mécanisme d'identité SIP [RFC4474]. Le reste de cette section décrit les détails de cette approche.

Si des certificats auto-signés sont utilisés, le contenu de l'attribut subjectAltName dans le certificat PEUT utiliser l'URI de l'utilisateur. Dans SIP, cet URI de l'utilisateur est l'adresse d'enregistrement de l'utilisateur (AOR, *Address of Record*). Ceci n'est utile qu'à des fins de débogage et n'est pas nécessaire pour lier le certificat à l'un des points dextrémité de la communication. Contrairement aux opérations TLS normales de ce protocole, lors d'une opération TLS d'homologue à homologue, le subjectAltName n'est pas un composant important de la vérification du certificat. Si le point d'extrémité est aussi capable d'établir des sessions anonymes, un certificat distinct et unique DOIT être utilisé à cette fin. Pour un client qui travaille avec plusieurs utilisateurs, chaque utilisateur DEVRAIT avoir son propre certificat. La génération de paires de clés publiques/privées étant relativement coûteuse, les points d'extrémité ne sont pas tenus de générer des certificats pour chaque session.

Une empreinte digitale de certificat est le résultat d'une fonction de hachage à sens unique calculée sur la forme des règles de codage dinstinctif (DER, *Distinguished Encoding Rules*) du certificat. Le point d'extrémité DOIT utiliser l'attribut d'empreinte digitale de certificat tel que spécifié dans la [RFC4572] et DOIT l'inclure dans le SDP. Le certificat présenté lors de l'échange TLS doit correspondre à l'empreinte digitale échangée via le SDP, et si l'empreinte digitale ne correspond pas au certificat haché, le point d'extrémité DOIT alors interrompre immédiatement la session de supports.

Quand on utilise SIP, l'intégrité de l'empreinte digitale peut être assurée par le mécanisme d'identité SIP [RFC4474]. Quand un client souhaite utiliser SIP pour établir une session MSRP sécurisée avec un autre point d'extrémité, il envoie une offre SDP dans un message SIP à l'autre point d'extrémité. Cette offre inclut, dans la charge utile SDP, l'empreinte digitale du certificat que le point d'extrémité souhaite utiliser. Le message SIP contenant l'offre est envoyé au mandataire SIP de l'offreur, qui ajoutera un en-tête Identity conformément aux procédures décrites dans la [RFC4474]. Lorsque le point d'extrémité distant reçoit le message SIP, il peut vérifier l'identité de l'envoyeur à l'aide de l'en-tête Identity. Comme l'en-tête Identity est une signature numérique sur plusieurs en-têtes SIP, en plus du ou des corps du message SIP, le receveur

peut également être certain que le message n'a pas été altéré après l'ajout de la signature numérique au message SIP.

Un exemple de SDP avec un attribut d'empreinte digitale est présenté dans la figure suivante. L'empreinte digitale devrait figurer sur une seule ligne.

```
c=IN IP4 atlanta.example.com
m=message 7654 TCP/TLS/MSRP *
a=accept-types:text/plain
a=path:msrps://atlanta.exemple.com:7654/jshA7weso3ks;tcp
a=fingerprint:SHA-1 \ 4A:AD:B9:B1:3F:82:18:3B:54:02:12:DF:3E:5D:49:6B:19:E5:7C:AB
```

Figure 19 : SDP avec attribut Fingerprint

14.5 Autres problèmes de sécurité

MSRP ne peut pas être utilisé comme amplificateur pour les attaques de DoS, mais il peut être utilisé pour former une attaque répartie afin de consommer les ressources de connexion TCP sur les serveurs. L'attaquant, Mallory, envoie un INVITE SIP sans offre à Alice. Alice renvoie un 200 avec une offre et Mallory renvoie une réponse avec SDP indiquant que son adresse MSRP est l'adresse de Tom. Comme Alice a envoyé l'offre, Alice va initier unee connexion avec Tom en utilisant les ressources du serveur de ce dernier. Étant donné le nombre énorme de clients de messagerie instantanée et le nombre relativement faible de connexions TCP que la plupart des serveurs prennent en charge, il s'agit d'une attaque assez simple.

SIP tente de résoudre les problèmes liés aux pourriels. Le problème du pourriel est probablement mieux traité au niveau SIP lorsqu'une session MSRP est initiée et non au niveau MSRP.

Si un envoyeur choisit d'utiliser S/MIME pour protéger un message, toutes les opérations S/MIME s'appliquent au message complet, avant tout découpage du message en tronçons.

La signalisation aura établi la session vers ou à partir de certains URI spécifiques qui auront souvent des schémas d'URI "im:" ou "sip:". Quand la signalisation a été établie pour un utilisateur final spécifique et que S/MIME est mis en œuvre, le client doit alors vérifier que le nom dans le SubjectAltName du certificat contient une entrée qui correspond à l'URI qui a été utilisé pour l'autre extrémité dans la signalisation. Dans certains cas, tels que les conférences IM, le nom du certificat S/MIME et l'identité signalée ne correspondent pas. Dans ce cas, le client devrait s'assurer que l'utilisateur est informé que le message provient de l'utilisateur identifié dans le certificat et qu'il ne suppose pas que le message provient de la partie qu'il a signalée.

Dans certains cas, un appareil envoyeur peut avoir besoin d'attribuer un message à une autre identité, et peut utiliser différentes identités pour différents messages dans la même session. Par exemple, un serveur de conférence peut envoyer des messages au nom de plusieurs utilisateurs dans la même session. Plutôt que d'ajouter des champs d'en-tête supplémentaires à MSRP à cette fin, MSRP s'appuie sur le format message/cpim. L'envoyeur peut envelopper un tel message dans un corps de message/cpim et placer l'identité réelle de l'expéditeur dans le champ From. La fiabilité d'une telle attribution est affectée par les propriétés de sécurité de la session de la même manière que la fiabilité de l'identité de l'homologue réel, avec la question supplémentaire de déterminer si le receveur fait confiance à l'envoyeur pour affirmer l'identité.

Cette approche peut entraîner l'imbrication d'enveloppes de messages/cpim. Par exemple, un message provient d'une passerelle CPIM et est ensuite transmis par un serveur de conférence à une nouvelle session. La passerelle et le serveur de conférence introduisent tous deux des enveloppes. Dans ce cas, le client receveur DEVRAIT indiquer à l'utilisateur la chaîne d'assertions d'identité, plutôt que de laisser l'utilisateur supposer que la passerelle ou le serveur de conférence est à l'origine du message.

Il est possible qu'un receveur reçoive des messages attribués au même expéditeur via différentes sessions MSRP. Par exemple, Alice pourrait être en conversation avec Bob via une session MSRP sur un canal protégé par TLS. Alice pourrait ensuite recevoir un message différent de Bob au cours d'une autre session, peut-être avec un serveur de conférence qui affirme l'identité de Bob dans une enveloppe message/cpim signée par le serveur.

MSRP n'interdit pas les sessions multiples simultanées entre la même paire d'identités. Il n'interdit pas non plus à un point d'extrémité d'envoyer un message au nom d'une autre identité, comme cela peut être le cas pour un serveur de conférence. Le point d'extrémité du receveur devrait déterminer son niveau de confiance dans l'authenticité de l'expéditeur

indépendamment pour chaque session. Le fait qu'un point d'extrémité fasse confiance à l'authenticité de l'envoyeur lors d'une session donnée ne doit pas affecter le niveau de confiance qu'il accorde apparemment au même expéditeur lors d'une session différente.

Quand des clients MSRP forment ou acquièrent un certificat, ils DEVRAIENT s'assurer que le subjectAltName comporte une entrée GeneralName de type uniformResourceIdentifier pour chaque URI correspondant à ce client et devrait toujours inclure un URI "im:". Le certificat peut contenir d'autres URI tels que "sip:" ou "xmpp:".

Les mises en œuvre de MSRP doivent être conscientes d'une attaque potentielle sur les appareils MSRP qui consiste à placer de très grandes valeurs dans le champ d'en-tête byte-range, ce qui peut amener l'appareil à allouer de très grandes mémoires tampons pour contenir le message. Les mises en œuvre DEVRAIENT appliquer un certain degré de vérification de bonne santé sur les valeurs de la plage d'octets avant d'allouer de telles mémoires tampons.

15. Considérations relatives à l'IANA

La présente spécification demande à l'IANA de créer un nouveau registre pour les paramètres MSRP. Le registre des paramètres MSRP est un conteneur pour des sous-registres. Cette section introduit également des sous-registres pour les noms de méthodes MSRP, les codes d'état et les noms des champs d'en-tête.

En outre, les paragraphes 15.4 à 15.7 enregistrent de nouveaux paramètres dans les registres IANA existants.

15.1 Nom des méthodes MSRP

Cette spécification établit le sous-registre Méthodes dans le registre Paramètres MSRP et initie son remplissage comme suit. Les nouveaux paramètres de ce sous-registre doivent être publiés dans une RFC (sous la forme d'une soumission à l'IETF ou de l'éditeur des RFC).

SEND - [RFC4975]
REPORT - [RFC4975]

Les informations suivantes DOIVENT être fournies dans une publication de RFC afin d'enregistrer un nouvelle méthode MSRP :
o nom de la méthode.
o numéro de la RFC dans laquelle la méthode est enregistrée.

15.2 Champs d'en-tête MSRP

La présente spécification établit le sous registre Champs de champ d'en-tête dans Paramètres MSRP. Les nouveaux paramètres dans ce sous registre doivent être publiés dans une RFC (de soumission IETF ou de l'éditeur des RFC). Son remplissage initial est défini comme suit :

To-Path - [RFC4975]
From-Path - [RFC4975]
Message-ID - [RFC4975]
Success-Report - [RFC4975]
Failure-Report - [RFC4975]
Byte-Range - [RFC4975]
Status - [RFC4975]

Les informations suivantes DOIVENT être fournies dans une RFC publiée afin d'enregistrer un nouveau champ d'en-tête MSRP :
o le nom du champ d'en-tête
o le numéro de la RFC dans laquelle la méthode est enregistrée.

15.3 Codes d'état MSRP

La présente spécification établit le sous registre Status-Code dans Paramètres MSRP. Les nouveaux paramètres dans ce

sous registre doivent être publiés dans une RFC (de soumission IETF ou de l'éditeur des RFC). Son remplissage initial est défini à la Section 10. Il prend le format suivant :

Code [Numéro de RFC]

Les informations suivantes DOIVENT être fournies dans une RFC publiée afin d'enregistrer un nouveau code d'état MSRP :
- o le numéro du code d'état,
- o le numéro de la RFC dans laquelle la méthode est enregistrée.

15.4 Accès MSRP

MSRP utilise l'accès TCP 2855, à partir de la gamme d'accès "registered". L'usage de cette valeur est décrit à la Section 6.

15.5 Schéma d'URI

Le présent document demande l'enregistrement permanent des schémas d'URI "msrp" et "msrps".

15.5.1 Schéma MSRP

Nom de schéma d'URI : "msrp"
Syntaxe de schéma d'URI : voir la construction ABNF pour "MSRP-URI" à la Section 9 de la RFC 4975.
Sémantique de schéma d'URI : voir la Section 6 de la RFC 4975.
Considérations de codage : voir la Section 6 de la RFC 4975.
Applications/protocoles qui utilisent ce schéma d'URI : protocole de relais de session de message (MSRP).
Considérations d'interopérabilité : les URI MSRP sont supposés n'être utilisés que par des mises en œuvre de MSRP. Aucun problème d'interopérabilité supplémentaire n'est prévu.
Considérations de sécurité : voir au paragraphe 14.1 de la RFC 4975 les considérations de sécurité spécifiques des URI MSRP, et à la Section 14 de la RFC 4975 les considérations de sécurité pour MSRP en général.
Contact : Ben Campbell (ben@estacado.net).
Auteur/contrôleur des changements : IESG.

15.5.2 Schéma MSRPS

Nom de schéma d'URI : "msrps"
Syntaxe de schéma d'URI : voir la construction ABNF pour "MSRP-URI" à la Section 9 de la RFC 4975.
Sémantique de schéma d'URI : voir la Section 6 de la RFC 4975.
Considérations de codage : voir la Section 6 de la RFC 4975.
Applications/protocoles qui utilisent ce schéma d'URI : protocole de relais de session de message (MSRP).
Considérations d'interopérabilité : les URI MSRP sont supposés n'être utilisés que par des mises en œuvre de MSRP. Aucun problème d'interopérabilité supplémentaire n'est prévu.
Considérations de sécurité : voir au paragraphe 14.1 de la RFC 4975 les considérations de sécurité spécifiques des URI MSRP, et à la Section 14 de la RFC 4975 les considérations de sécurité pour MSRP en général.
Contact : Ben Campbell (ben@estacado.net).
Auteur/contrôleur des changements : IESG.

15.6 Protocole de transport SDP

MSRP définit les nouvelles valeurs de champ de protocole SDP "TCP/MSRP" et "TCP/TLS/MSRP", qui devraient être enregistrées dans le registre sdp-parameters sous "proto". Cette première valeur indique le protocole MSRP quand TCP est utilisé comme transport sous-jacent. La seconde indique l'utilisation de TLS sur TCP.

Les spécifications qui définissent de nouvelles valeurs de protocole doivent définir les règles pour l'espace de noms de format de supports associé. Les valeurs de protocole "TCP/MSRP" et "TCP/TLS/MSRP" ne permettent qu'une seule valeur dans le champ de format (fmt) qui est une seule occurrence de "*". La détermination réelle du format est faite en utilisant les attributs "accept-types" et "accept-wrapped-types".

15.7 Noms d'attribut SDP

Le présent document enregistre les noms de paramètre d'attribut SDP suivants dans le registre sdp-parameters. Ces noms sont à utiliser dans le champ SDP att-name.

15.7.1 Types Accept

Informations de contact : Ben Campbell (ben@estacado.net)
Nom d'attribut : accept-types
Forme longue de nom d'attribut : types de support acceptables
Type : niveau support
Soumis à l'attribut Charset : non
Objet et valeurs appropriées : l'attribut "accept-types" contient une liste des types de supports que le point d'extrémité accepte de recevoir. Il peut contenir zéro, un ou plusieurs types de supports enregistrés, ou "*" dans une chaîne d'espace délimité.

15.7.2. Types Wrapped

Informations de contact : Ben Campbell (ben@estacado.net)
Nom d'attribut : accept-wrapped-types
Forme longue de nom d'attribut : types de support acceptables à l'intérieur d'enveloppes
Type : niveau support
Soumis à l'attribut Charset : non
Objet et valeurs appropriées : l'attribut "accept-wrapped-types" contient une liste des types de supports que le point d'extrémité accepte de recevoir dans un message MSRP avec un contenu multi parties, mais ne peut pas être utilisé comme type le plus externe du message. Il peut contenir zéro, un ou plusieurs types de supports enregistrés, ou "*" dans une chaîne d'espace délimité.

15.7.3 Taille maximum

Informations de contact : Ben Campbell (ben@estacado.net)
Nom d'attribut : max-size
Forme longue de nom d'attribut : taille maximum de message
Type : niveau support
Soumis à l'attribut Charset : non
Objet et valeurs appropriées : l'attribut "max-size" indique le plus grand message qu'un point d'extrémité souhaite accepter. Il peut prendre n'importe quelle valeur numérique entière, spécifiée en octets.

15.7.4 Path

Informations de contact : Ben Campbell (ben@estacado.net)
Nom d'attribut : path
Forme longue de nom d'attribut : chemin d'URI MSRP
Type : niveau support
Soumis à l'attribut Charset : non
Objet et valeurs appropriées : l'attribut "path" indique une série d'appareils MSRP qui doivent être visités par les messages envoyés dans la session, incluant le point d'extrémité final. L'attribut contient un ou plusieurs URI MSRP, délimités par le caractère espace (SP).

16. Contributeurs et remerciements

En plus des éditeurs, les personnes suivantes ont contribué de façon conséquente au présent document : Chris Boulton, Paul Kyzivat, Orit Levin, Hans Persson, Adam Roach, Jonathan Rosenberg, et Robert Sparks.

Les personnes suivantes ont contribué par des discussions substancielles et des retours sur le présent travail : Eric Burger, Allison Mankin, Jon Peterson, Brian Rosen, Dean Willis, Aki Niemi, Hisham Khartabil, Pekka Pessi, Miguel Garcia, Peter Ridler, Sam Hartman, et Jean Mahoney.

17. Références

17.1 Références normatives

[RFC2045] N. Freed et N. Borenstein, "Extensions de messagerie Internet multi-objets (MIME) Partie 1 : Format des corps de message Internet", novembre 1996. *(D. S., MàJ par* 2184, 2231, 5335.)

[RFC2046] N. Freed et N. Borenstein, "Extensions de messagerie Internet multi-objets (MIME) Partie 2 : Types de support", novembre 1996. *(D. S., MàJ par* 2646, 3798, 5147, 6657, 8098)

[RFC2119] S. Bradner, "Mots clés à utiliser dans les RFC pour indiquer les niveaux d'exigence", BCP 14, mars 1997. *(MàJ par* RFC8174)

[RFC2183] R. Troost, S. Dorner, K. Moore, éd., "Communication des informations de présentation dans les messages Internet : le champ d'en-tête Contenu-disposition", août 1997. (*MàJ par* RFC2184, RFC2231) *(P.S.)*

[RFC3261] J. Rosenberg et autres, "SIP : Protocole d'initialisation de session", juin 2002. (*Mise à jour par* 3265, 3853, 4320, 4916, 5393, 6665, 8217, 8760)

[RFC3264] J. Rosenberg et H. Schulzrinne, "Modèle d'offre/réponse avec le protocole de description de session (SDP)", juin 2002. *(P.S. ; MàJ par* RFC8843, 9143)

[RFC3268] P. Chown, "Suites de chiffrement de la norme de chiffrement évolué (AES) pour la sécurité de la couche Transport (TLS)", juin 2002. (*Obsolète, voir* RFC5246) *(P.S.)*

[RFC3280] R. Housley, W. Polk, W. Ford et D. Solo, "Profil de certificat d'infrastructure de clé publique X.509 et de liste de révocation de certificat (CRL) pour l'Internet", avril 2002. *(Obsolète, voir* RFC5280)

[RFC3629] F. Yergeau, "UTF-8, un format de transformation de la norme ISO 10646", STD 63, novembre 2003.

[RFC3851] B. Ramsdell, "Spécification du message d'extensions de messagerie Internet multi-objets/sécurisé (S/MIME) version 3.1", juillet 2004. *(Obsolète, voir* RFC5751)

[RFC3862] G. Klyne, D. Atkins, "Profil commun pour la messagerie instantanée (CPIM) : format de message ", août 2004. *(P.S.)*

[RFC3986] T. Berners-Lee, R. Fielding et L. Masinter, "Identifiant de ressource uniforme (URI) : Syntaxe générique", STD 66, janvier 2005. *(P.S. ; MàJ par* RFC8820)

[RFC4234] D. Crocker et P. Overell, "BNF augmenté pour les spécifications de syntaxe : ABNF", octobre 2005. *(Remplace* RFC2234, *remplacée par* RFC5234)

[RFC4346] T. Dierks et E. Rescorla, "Protocole de sécurité de la couche Transport (TLS) version 1.1", avril 2006. *(Remplace* RFC2246 *; Remplacée par* RFC5246 *; MàJ par* RFC4366, 4680, 4681, 5746, 6176, 7465, 7507, 7919)

[RFC4366] S. Blake-Wilson et autres, "Extensions de sécurité de la couche Transport (TLS)", avril 2006. (*Obsolète,* RFC5246) *(P.S.)*

[RFC4474] J. Peterson et C. Jennings, "Améliorations de la gestion d'identité authentifiée dans le protocole d'initialisation de session (SIP)", août 2006. *(P.S. ; Remplacée par* RFC8224)

[RFC4566] M. Handley, V. Jacobson et C. Perkins, "SDP : Protocole de description de session", juillet 2006. *(P.S. ; remplacée par RFC8866)*

[RFC4572] J. Lennox, "Transport sur support en mode connexion sur le protocole de sécurité de la couche Transport (TLS) dans le protocole de description de session (SDP)", juillet 2006. (*MàJ* RFC4145) *(P.S. ; remplacée par* RFC8122)

17.2 Références pour information

[RFC3217] R. Housley, "Enveloppe de clé en Triple-DES et RC2", décembre 2001. *(Information)*

[RFC3311] J. Rosenberg, "Méthode UPDATE du protocole d'initialisation de session (SIP) ", octobre 2002.

[RFC3428] B. Campbell et autres, "Extension de messagerie instantanée pour le protocole d'initialisation de session (SIP)", décembre 2002.

[RFC3725] J. Rosenberg et autres, "Bonnes pratiques actuelles pour la commande d'appel de tiers (3pcc) dans le protocole d'initialisation de session (SIP)", avril 2004. (BCP0085)

[RFC3840] J. Rosenberg, H. Schulzrinne et P. Kyzivat, "Indication des capacités d'agent d'utilisateur dans le protocole d'initialisation de session (SIP)", août 2004

[RFC3860] J. Peterson, "Profil commun pour la messagerie instantanée (CPIM)", août 2004. *(P.S.)*

[RFC3861] J. Peterson, "Résolution d'adresse pour la messagerie instantanée et les services de présence", août 2004. *(P.S.)*

[RFC3921] P. Saint-Andre, éd., "Protocole extensible de messagerie et de présence (XMPP) : messagerie instantanée et présence", octobre 2004. *(P.S.) (Remplacée par* RFC6121)

[RFC3960] G. Camarillo, H. Schulzrinne, "Support précoce et génération des tonalités d'appel dans le protocole d'initialisation de session (SIP)", décembre 2004. *(Information)*

[RFC4145] D. Yon, G. Camarillo, "Transport de support fondé sur TCP dan le protocole de description de session (SDP)", septembre 2005. (*MàJ par* RFC4572) *(P.S.)*

[RFC4579] A. Johnston, O. Levin, "Commande d'appel du protocole d'initialisation de session (SIP) – Conférence pour agents d'utilisateur", août 2006. (BCP0119)

[RFC4976] C. Jennings et autres, "Extensions de relais au protocole de relais de session de message (MSRP)", septembre 2007. *(P.S.. ; MàJ par* RFC7977 ; RFC 8553, RFC 8996)

[RFC5589] R. Sparks, A. Johnston, éd., D. Petrie, "Contrôle du transfert d'appel dans le protocole d'initialisation de session (SIP)", juin 2009. (BCP0149)

[RFC6072] C. Jennings, J. Fischl, éditeurs, "Service de gestion de certificats pour le protocole d'initialisation de session (SIP)", février 2011. *(P. S.)*

Adresse des auteurs

Ben Campbell
Estacado Systems
17210 Campbell Road
Suite 250
Dallas, TX 75252
USA
mél : ben@estacado.net

Rohan Mahy
Plantronics
345 Encincal Street
Santa Cruz, CA 95060
USA
mél : rohan@ekabal.com

Cullen Jennings
Cisco Systems, Inc.
170 West Tasman Dr.
MS: SJC-21/2
San Jose, CA 95134
USA
mél : fluffy@cisco.com

Déclaration complète de droits de reproduction

Remerciement
Le financement de la fonction d'édition des RFC est actuellement fourni par la Internet Society.