Groupe de travail Réseau
Request for Comments : 3903
Catégorie : En cours de normalisation

A. Niemi, éditeur, Nokia
octobre 2004
Traduction Claude Brière de L'Isle

Extension au protocole d'initialisation de session (SIP) pour la publication d'état d'événement

Statut de ce mémoire
Le présent document spécifie un protocole Internet en cours de normalisation pour la communauté de l'Internet, et appelle à des discussions et des suggestions pour son amélioration. Prière de se reporter à l'édition actuelle du STD 1 "Normes des protocoles officiels de l'Internet" pour connaître l'état de normalisation et le statut de ce protocole. La distribution du présent mémoire n'est soumise à aucune restriction.

Résumé
Le présent document décrit une extension au protocole d'initialisation de session (SIP, *Session Initiation Protocol*) pour la publication de l'état d'événement utilisé au sein du cadre des événements SIP. La première application de cette extension est pour la publication des informations de présence.

Le mécanisme décrit dans le présent document peut être étendu pour prendre en charge la publication de tout état d'événement pour lequel existe un paquetage d'événements approprié. Il n'est pas destiné à devenir un mécanisme général de transport de données arbitraires, car il existe des mécanismes mieux adaptés à cette fin.

Table des Matières

1. Introduction....2
2. Définitions et conventions du document....2
3. Fonctionnement global....3
4. Construction des demandes PUBLISH....3
4.1 Identification d'état d'événement publié....4
4.2 Création de la publication initiale....5
4.3 Rafraîchissement d'état d'événement....5
4.4 Modification de l'état d'événement....5
4.5 Retrait de l'état d'événement....6
5. Traitement des réponses PUBLISH....6
6. Traitement des demandes PUBLISH....6
7. Traitement des demandes OPTIONS....8
8. Utilisation des étiquettes d'entité dans PUBLISH....8
8.1 Notes générales....8
8.2 Utilisation du client....8
8.3 Utilisation du serveur....9
9. Contrôle du taux de publication....9
10. Considérations sur les paquetages d'événement qui utilisent PUBLISH....9
10.1 Corps PUBLISH....9
10.2 Corps de réponse PUBLISH....9
10.3 Sources multiples pour l'état d'événement....10
10.4 Segmentation d'état d'événement....10
10.5 Taux de publication....10
11. Définitions d'élément de protocole....10
11.1 Nouvelles méthodes....10
11.2 Nouveaux codes de réponse....12
11.3 Nouveaux champs d'en-tête....12
12. Définitions de BNF augmenté....12
13. Considérations relatives à l'IANA....12
13.1 Méthodes....13
13.2 Codes de réponse....13
13.3 Noms de champs d'en-tête....13
14. Considérations pour la sécurité....13
14.1 Contrôle d'accès....13

14.2 Attaques de déni de service....13
14.3 Attaques de répétition....13
14.4 Attaques par interposition....14
14.5 Confidentialité....14
15. Exemples....14
16. Contributeurs....18
17. Remerciements....18
18. Références....18
18.1 Références normatives....18
18.2 Références informatives....19
Adresse de l'auteur....19
Déclaration complète de droits de reproduction....19

1. Introduction

La présente spécification fournit un cadre pour la publication d'un état d'événement d'un agent d'utilisateur à une entité qui est responsable de composer cet état d'événement et de le distribuer aux parties intéressées à travers le cadre d'événements SIP [RFC3265].

En plus de la définition d'un cadre de publication d'événements, la présente spécification définit un usage concret de ce cadre pour la publication de l'état de présence [RFC3856] par un agent d'utilisateur de présence [RFC2778] à un composeur de présence, qui a déclanché une relation étroitement couplée avec l'agent de présence [RFC3265].

Les exigences et le modèle pour la publication de présence sont documentés dans [SIMPLE]. La présente spécification va traiter chacune de ces exigences.

Le mécanisme décrit dans le présent document peut être étendu pour prendre en charge la publication de tout état d'événement pour lequel il existe un paquetage d'événements approprié comme défini dans la [RFC3265]. Par exemple, une application d'événements SIP pour les indications de message en attente [RFC3842] pourrait choisir de collecter les états des boîtes de messagerie vocale à travers un ensemble d'agent d'utilisateurs en utilisant le mécanisme PUBLISH. Le composeur dans une telle application serait alors responsable de la collecte et de la distribution de cet état aux abonnés au paquetage d'événements.

Chaque application qui fait usage du mécanisme PUBLISH dans la publication d'un état d'événement va devoir adhérer à l'ensemble des lignes directrices de la Section 10. Le mécanisme décrit dans le présent document n'est pas destiné à être un mécanisme général de transport de données arbitraires, car il existe des mécanismes mieux adaptés à cette fin.

2. Définitions et conventions du document

En plus des définitions des [RFC2778], [RFC3261], et [RFC3265], le présent document introduit quelques nouveaux concepts :

État d'événement *(Event State)* : informations d'état pour une ressource, associées à un paquetage d'événements et une adresse d'entretien.

Agent de publication d'événement (EPA, *Event Publication Agent*) : c'est le client d'agent d'utilisateur (UAC, *User Agent Client*) qui produit les demandes PUBLISH pour publier l'état d'événement.

Composeur d'état d'événement (ESC, *Event State Compositor*) : c'est le serveur d'agent d'utilisateur (UAS, *User Agent Server*) qui traite les demandes PUBLISH, et est responsable de la composition de l'état d'événement dans un état d'événement composite complet d'une ressource.

Composeur de présence *(Presence Compositor)* : Type de composeur d'état d'événement qui est chargé de la composition de l'état de présence pour une présentité.

Publication : Acte d'un EPA qui envoie une demande PUBLISH à un ESC pour publier un état d'événement.

État fixe d'événement *(Event Hard State)* : État permanent ou état d'événement par défaut d'une ressource, que l'ESC peut

utiliser en l'absence de, ou en plus de, publications d'états conditionnels.

État conditionnel d'événement *(Event Soft State)* : État d'événement publié par un EPA en utilisant le mécanisme PUBLISH. Un élément de protocole (c'est-à-dire, une étiquette d'entité) est utilisé pour identifier une entité d'état conditionnel spécifique à l'ESC. L'état conditionnel a une durée de vie définie et va arriver à expiration après un délai négocié.

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 la [RFC2119] et indiquent les niveaux d'exigence pour les mises en œuvre conformes.

> Les passages en retrait comme celui-ci sont utilisés dans le présent document pour donner des informations supplémentaires et des précisions. Ils ne contiennent pas de descriptions de comportements normatifs du protocole.

3. Fonctionnement global

Le présent document définit une nouvelle méthode SIP, PUBLISH, pour publier un état d'événement. PUBLISH est similaire à REGISTER en ce qu'elle permet à un usager de créer, modifier, et supprimer l'état dans une autre entité qui gère cet état au nom de l'usager. Adresser une demande PUBLISH est identique à adresser une demande SUBSCRIBE. L'URI de demande d'une demande PUBLISH est remplie avec l'adresse de la ressource pour laquelle l'usager souhaite publier l'état d'événement. L'usager peut à son tour avoir plusieurs agents d'utilisateur ou points d'extrémité qui publient l'état d'événement. Chaque point d'extrémité peut publier son propre état unique, à partir duquel le composeur d'état d'événement génère l'état d'événement composite de la ressource. En plus d'une ressource particulière, tout état d'événement publié est associé à un paquetage d'événements spécifique. Par un abonnement à ce paquetage d'événements, l'usager est capable de découvrir l'état d'événement composite de toutes les publications actives.

Un client d'agent d'utilisateur (UAC) qui publie un état d'événement est appelé un agent de publication d'événement (EPA, *Event Publication Agent*). Pour la présence, c'est le rôle familier d'agent d'utilisateur de présence (PUA, *Presence User Agent*) comme défini dans la [RFC3856]. L'entité qui traite la demande PUBLISH est appelée composeur d'état d'événement (ESC, *Event State Compositor*). Pour la présence, c'est le rôle familier d'agent de présence (PA, *Presence Agent*) comme défini dans la [RFC3856].

Les demandes PUBLISH créent un état conditionnel dans l'ESC. Cet état d'événement conditionnel a une durée de vie définie et va arriver à expiration après un délai négocié, exigeant que la publication soit rafraîchie par les demandes PUBLISH ultérieures. Il peut aussi y avoir un état fixe d'événement provisionné pour chaque ressource pour un paquetage d'événements particulier. Cet état d'événement représente l'état de la ressource qui est toujours présent, et n'arrive pas à expiration. L'ESC peut utiliser l'état fixe d'événement en l'absence de, ou en plus de, l'état d'événement conditionnel fourni par le mécanisme PUBLISH. Le réglage de cet état fixe d'événement ou la configuration de la politique d'ESC concernant l'agrégation des différents états d'événement sort du domaine d'application de la présente spécification.

Le corps d'une demande PUBLISH porte l'état d'événement publié. En réponse à chaque demande PUBLISH réussie, l'ESC alloue un identifiant à la publication sous la forme d'une étiquette d'entité. Cet identifiant est alors utilisé par l'EPA dans toute demande PUBLISH ultérieure qui modifie, rafraîchie ou supprime l'état d'événement de cette publication. Lorsque l'état d'événement expire ou est explicitement supprimé, l'étiquette d'entité qui y est associée devient invalide. Une publication pour une étiquette d'entité invalide va naturellement échouer, et l'EPA aura besoin de recommencer et renvoyer son état d'événement sans faire référence à une étiquette d'entité précédente.

4. Construction des demandes PUBLISH

Les demandes PUBLISH créent, modifient, et suppriment l'état d'événement associé à une adresse d'entretien. Un tiers convenablement autorisé peut aussi effectuer la publication au nom d'une adresse d'entretien particulière.

Sauf comme on l'a noté, la construction de la demande PUBLISH et le comportement des clients qui envoient une demande PUBLISH sont identiques au comportement général d'un UAC décrit aux paragraphes 8.1 et 17.1 de la [RFC3261].

Si nécessaire, les clients peuvent faire un sondage sur la prise en charge de PUBLISH en utilisant la demande OPTIONS définie dans SIP [RFC3261]. La présence de "PUBLISH" dans le champ d'en-tête "Allow" dans une réponse à une demande OPTIONS indique la prise en charge de la méthode PUBLISH. De plus, le champ d'en-tête "Allow-Events" indique les paquetages d'événements pris en charge.

> Noter qu'il est possible que la demande OPTIONS fourche, et retourne par conséquent une réponse provenant d'un agent d'utilisateur autre que l'ESC. Dans ce cas, la prise en charge de la méthode PUBLISH peut n'être pas représentée de façon appropriée pour cet URI de demande particulier.

Une demande PUBLISH n'établit pas de dialogue. Un UAC PEUT inclure un champ d'en-tête Route dans une demande PUBLISH sur la base d'un chemin préexistant réglé comme décrit au paragraphe 8.1 de la [RFC3261]. Le champ d'en-tête Record-Route n'a pas de signification dans les demandes ou réponses PUBLISH, et DOIT être ignoré si il est présent. En particulier, l'UAC NE DOIT PAS créer un nouveau chemin réglé sur la base de la présence ou l'absence d'un champ d'en-tête Record-Route dans une réponse à une demande PUBLISH.

La demande PUBLISH PEUT contenir un champ d'en-tête Contact, mais en inclure un dans une demande PUBLISH n'a pas de signification dans le contexte de la publication d'événement et sera ignoré par l'ESC. Un EPA PEUT envoyer une demande PUBLISH au sein d'un dialogue existant. Dans ce cas, la demande est reçue dans le contexte de toute ou toutes sessions de supports associées à ce dialogue.

> Noter qu'alors que l'envoi d'une demande PUBLISH au sein d'un dialogue existant n'est pas interdit, il ne va normalement pas résulter en le comportement attendu. Sauf si l'autre extrémité du dialogue est aussi un ESC, elle va probablement rejeter la demande.

Les EPA NE DOIVENT PAS envoyer une nouvelle demande PUBLISH (pas une retransmission) pour le même URI de demande tant qu'ils n'ont pas reçu une réponse finale de l'ESC pour la précédente ou tant que la demande PUBLISH précédente n'est pas arrivée en fin de temporisation.

4.1 Identification d'état d'événement publié

L'identification de l'état d'événement publié est fournie par trois éléments d'information : l'URI de demande, le type d'événement, et (facultativement) une étiquette d'entité.

L'URI de demande d'une demande PUBLISH contient assez d'informations pour acheminer la demande à l'entité appropriée selon les procédures d'acheminement de demande mentionnées dans la [RFC3261]. Il contient aussi assez d'informations pour identifier la ressource dont l'état d'événement est à publier, mais pas assez d'informations pour déterminer le type de l'état d'événement publié.

Pour déterminer le type de l'état d'événement publié, l'EPA DOIT inclure un seul champ d'en-tête Event dans les demandes PUBLISH. La valeur de ce champ d'en-tête indique le paquetage d'événements pour lequel cette demande publie l'état d'événement.

Pour chaque demande PUBLISH réussie, l'ESC va générer et allouer une étiquette d'entité et la retourner dans le champ d'en-tête SIP-Etag de la réponse 2xx.

Lorsque elles mettent à jour un état d'événement publié précédemment, les demandes PUBLISH DOIVENT contenir un seul champ d'en-tête SIP-If-Match qui identifie l'état d'événement spécifique que rafraîchit, modifie ou supprime la demande. Ce champ d'en-tête DOIT contenir une seule étiquette d'entité qui a été retournée par l'ESC dans le champ d'en-tête SIP-ETag de la réponse à une publication précédente.

La demande PUBLISH PEUT contenir un corps qui contient un état d'événement que le client souhaite publier. Le format et la sémantique du contenu dépendent du paquetage d'événements identifié dans le champ d'en-tête Event.

La présence d'un corps et du champ d'en-tête SIP-If-Match déterminent l'opération spécifique qu'effectue la demande, comme décrit au Tableau 1.

Opération	Corps ?	SIP-If-Match ?	Valeur d'expiration
Initiale	oui	non	> 0
Rafraîchissement	non	oui	> 0
Modification	oui	oui	> 0
Suppression	non	oui	0

Tableau 1 : Opérations de publication

Une publication 'Initiale' règle l'état d'événement initial pour un EPA particulier. Il peut, bien sûr, y avoir déjà un état d'événement publié par d'autres EPA (pour la même adresse d'entretien). Cet état n'est pas affecté par une publication initiale. Une publication 'Rafraîchissement' rafraîchit la durée de vie d'une publication antérieure, tandis qu'une publication

'Modification' modifie l'état d'événement d'une publication précédente. Une publication 'Suppression' demande la suppression immédiate de l'état d'événement. Ces opérations sont décrites plus en détails dans les paragraphes qui suivent.

4.2 Création de la publication initiale

Une publication initiale est une demande PUBLISH créée par l'EPA et envoyée à l'ESC qui établit l'état conditionnel pour le paquetage d'événements indiqué dans le champ d'en-tête Event de la demande, et lié à l'adresse dans l'URI de demande de la demande.

Une demande PUBLISH initiale NE DOIT PAS contenir un champ d'en-tête SIP-If-Match. Cependant, si l'EPA s'attend à ce qu'une étiquette d'entité appropriée, mémorisée localement soit encore valide, il DEVRAIT d'abord essayer de modifier cet état d'événement comme décrit au paragraphe 4.4, au lieu de soumettre une publication initiale.

Une demande PUBLISH initiale DOIT contenir un corps qui contient l'état d'événement publié.

Une demande PUBLISH initiale PEUT contenir un seul champ d'en-tête Expires. Cette valeur indique la durée de vie suggérée de la publication d'état d'événement.

L'ESC peut diminuer la durée de vie suggérée de la publication, mais ne va jamais l'allonger. Si un champ d'en-tête Expires n'est pas présent, l'EPA indique son désir que l'ESC choisisse. Le champ d'en-tête Expires dans une réponse 2xx à la PUBLISH initiale indique la durée réelle pendant laquelle la publication va rester active. Sauf si elle est rafraîchie avant que cette durée de vie soit écoulée, la publication va expirer.

4.3 Rafraîchissement d'état d'événement

Un EPA est responsable du rafraîchissement de ses publications établies précédemment avant que leur intervalle d'expiration soit écoulé. Pour rafraîchir une publication, l'EPA DOIT créer une demande PUBLISH qui inclut dans un champ d'en-tête SIP-If-Match l'étiquette d'entité de la publication à rafraîchir.

Le champ d'en-tête SIP-If-Match contenant une étiquette d'entité conditionne la demande PUBLISH à rafraîchir un état d'événement spécifique établi par une publication antérieure. Si l'étiquette d'entité correspond à un état d'événement précédemment publié à l'ESC, le rafraîchissement réussit, et l'EPA reçoit une réponse 2xx.

Comme la réponse 2xx à une demande PUBLISH initiale, la réponse 2xx à une demande de rafraîchissement PUBLISH va contenir un champ d'en-tête SIP-ETag avec une étiquette d'entité. L'EPA DOIT mémoriser cette étiquette d'entité, et remplacer toute étiquette d'entité existante pour l'état d'événement rafraîchi. Voit au paragraphe 8.2 plus d'informations sur le traitement par l'EPA de l'étiquette d'entités.

Si il n'y a pas d'état d'événement correspondant, par exemple, l'état d'événement à rafraîchir est déjà arrivé à expiration, l'EPA reçoit une réponse 412 (Échec de demande conditionnelle) à la demande PUBLISH.

Un rafraîchissement de publication PEUT contenir un seul champ d'en-tête Expires. Cette valeur indique la durée de vie suggérée de l'état d'événement.

L'ESC peut diminuer la durée de vie suggérée du rafraîchissement de la publication, mais il ne peut jamais l'allonger. Si un champ d'en-tête Expires n'est pas présent, l'EPA indique qu'il désire que l'ESC choisisse. Le champ d'en-tête Expires dans une réponse 2xx au rafraîchissement de publication indique la durée réelle pendant laquelle la publication va rester active.

Un rafraîchissement de publication étend seulement le temps d'expiration d'un état d'événement déjà existant. Il n'affecte pas cet état d'événement d'une autre façon. Donc, une demande PUBLISH qui rafraîchit un état d'événement NE DOIT PAS avoir de corps.

4.4 Modification de l'état d'événement

Modifier un état d'événement ressemble étroitement à la création d'un état d'événement initial. Cependant, au lieu d'établir complètement un nouvel état d'événement à l'ESC, l'état d'événement déjà existant est mis à jour avec un état d'événement modifié. La nature de cette mise à jour dépend du contenu du corps, et de la sémantique associée au format de ce corps.

Pour modifier un état d'événement, l'EPA DOIT construire une demande PUBLISH qui inclut dans un champ d'en-tête SIP-If-Match l'étiquette d'entité de la publication d'état d'événement à modifier. Une demande PUBLISH qui modifie un état d'événement DOIT contenir un corps qui inclut l'état d'événement modifié.

Le champ d'en-tête SIP-If-Match conditionne la demande PUBLISH à modifier un état d'événement spécifique établi par une publication antérieure, et identifié par l'étiquette d'entité. Si l'étiquette d'entité correspond à l'état d'événement antérieurement publié à l'ESC, cet état d'événement est remplacé par l'état d'événement porté dans la demande PUBLISH, et l'EPA reçoit une réponse 2xx.

Comme la réponse 2xx à une demande PUBLISH initiale, la réponse 2xx à une demande PUBLISH modificatrice va contenir un champ d'en-tête SIP-ETag avec une étiquette d'entité. L'EPA DOIT mémoriser cette étiquette d'entité, en remplaçant toute étiquette d'entité existante pour l'état d'événement modifié. Voir au paragraphe 8.2 plus d'informations sur le traitement par l'EPA de l'étiquette d'entité.

Si il n'y a pas de correspondance de l'état d'événement à l'ESC, par exemple, l'état d'événement à modifier a déjà expiré, l'EPA reçoit une réponse 412 (Échec de la demande conditionnelle) à la demande PUBLISH.

Une demande PUBLISH modificatrice PEUT contenir un seul champ d'en-tête Expires. Cette valeur indique la durée de vie suggérée de la publication d'état d'événement.

L'ESC peut diminuer la durée de vie suggérée de la publication, mais il ne va jamais l'augmenter. Si un champ d'en-tête Expires n'est pas présent, l'EPA indique son désir que l'ESC choisisse. Le champ d'en-tête Expires dans une réponse 2xx à la demande PUBLISH modificatrice indique la durée réelle pendant laquelle la publication va rester active. Sauf rafraîchissement avant le dépassement de cette durée de vie, la publication va arriver à expiration.

4.5 Retrait de l'état d'événement

L'état d'événement établi par une publication antérieure peut aussi être explicitement retiré.

Pour demander le retrait immédiat d'un état d'événement, un EPA DOIT créer une demande PUBLISH avec une valeur Expires de "0", et régler le champ d'en-tête SIP-If-Match à contenir l'étiquette d'entité de la publication d'état d'événement à supprimer.

> Noter que le retrait d'un état d'événement est effectivement un rafraîchissement de publication suggérant un intervalle d'expiration infinitésimal. Par conséquent, l'état d'événement rafraîchi expire immédiatement après avoir été rafraîchi.

Comme pour un rafraîchissement d'état d'événement, la suppression de l'état d'événement affecte seulement l'expiration de l'état d'événement. Donc, une demande PUBLISH qui supprime un état d'événement NE DOIT PAS contenir de corps.

5. Traitement des réponses PUBLISH

Lors du traitement des réponses aux demandes PUBLISH, les étapes du paragraphe 8.1.2 de la [RFC3261] s'appliquent.

Si un EPA reçoit une réponse 412 (Échec de demande conditionnelle) il NE DOIT PAS tenter de refaire la demande PUBLISH. Pour publier l'état d'événement, l'EPA DEVRAIT plutôt effectuer une publication initiale, c'est-à-dire, une demande PUBLISH sans champ d'en-tête SIP-If-Match, comme décrit au paragraphe 4.2. L'EPA DOIT aussi éliminer l'étiquette d'entité qui a produit cette réponse d'erreur.

Si un EPA reçoit une réponse 423 (Intervalle trop bref) à une demande PUBLISH, il PEUT réessayer la publication après avoir changé l'intervalle d'expiration dans le champ d'en-tête Expires pour qu'il soit égal ou supérieur à l'intervalle d'expiration contenu dans le champ d'en-tête Min-Expires de la réponse 423 (Intervalle trop bref).

6. Traitement des demandes PUBLISH

Le composeur d'état d'événement (ESC, *Event State Compositor*) est un serveur d'agent d'utilisateur (UAS) qui traite et répond aux demandes PUBLISH, et tient une liste des publications pour une certaine adresse d'entretien. L'ESC doit savoir (par exemple, par configuration) l'ensemble des adresses pour lesquelles il tient l'état d'événement.

L'ESC DOIT ignorer le champ d'en-tête Record-Route si il est inclus dans une demande PUBLISH. L'ESC NE DOIT PAS inclure de champ d'en-tête Record-Route dans une réponse à une demande PUBLISH. L'ESC DOIT ignorer le champ d'en-tête Contact si il en est un présent dans une demande PUBLISH.

Les demandes PUBLISH avec le même URI de demande DOITVENT être traitées dans l'ordre dans lequel elles sont reçues. Les demandes PUBLISH DOIVENT aussi être traitées de façon atomique, c'est à dire qu'une demande PUBLISH particulière est traité soit complètement, soit pas du tout.

Lorsque il reçoit une demande PUBLISH, l'ESC suit les étapes qui définissent le comportement général d'UAS du paragraphe 8.2 de la [RFC3261]. De plus, pour le comportement spécifique de PUBLISH, l'ESC suit ces étapes :

1. L'ESC inspecte l'URI de demande pour déterminer si cette demande est ciblée sur une ressource pour laquelle l'ESC est chargé de la maintenance de l'état d'événement. Sinon, l'ESC DOIT retourner une réponse 404 (Pas trouvé) et sauter les étapes suivantes.
   Il se peut aussi que l'URI de demande pointe sur un domaine dont l'ESC n'est pas responsable. Dans ce cas, l'UAS qui reçoit la demande peut assumer le rôle d'un serveur mandataire et transmettre la demande à une cible plus appropriée.

2. L'ESC examine le champ d'en-tête Event de la demande PUBLISH. Si le champ d'en-tête Event manque ou contient un paquetage d'événements que l'ESC ne prend pas en charge, l'ESC DOIT répondre à la demande PUBLISH par une réponse 489 (Mauvais événement) et sauter les étapes restantes.

3. L'ESC examine le champ d'en-tête SIP-If-Match de la demande PUBLISH à la recherche de la présence d'une précondition de la demande.
   * Si la demande ne contient pas de champ d'en-tête SIP-If-Match, l'ESC DOIT générer et mémoriser une étiquette d'entité unique en local pour identifier la publication. Cette étiquette d'entité est associée à l'état d'événement porté dans le corps de la demande PUBLISH.
   * Autrement, si la demande a un champ d'en-tête SIP-If-Match, l'ESC vérifie si le champ d'en-tête contient une seule étiquette d'entité. Si c'est non, la demande est invalide, et l'ESC DOIT retourner une réponse 400 (Demande invalide) et sauter les étapes restantes.
   * Autrement, l'ESC extrait l'étiquette d'entité contenue dans le champ d'en-tête SIP-If-Match et confronte cette étiquette d'entité à toutes les étiquettes d'entité mémorisées localement pour cette ressource et ce paquetage d'événements. Si aucune correspondance n'est trouvée, l'ESC DOIT rejeter la publication avec une réponse 412 (Échec de la demande conditionnelle) et sauter les étapes restantes.

4. L'ESC traite la valeur du champ d'en-tête Expires de la demande PUBLISH.
   * Si la demande comporte un champ d'en-tête Expires, cette valeur DOIT être prise comme expiration demandée.
   * Autrement, une valeur par défaut configurée localement DOIT être prise comme expiration demandée.
   * L'ESC PEUT choisir une expiration inférieure à l'intervalle d'expiration demandé. C'est seulement si l'intervalle d'expiration demandé est supérieur à zéro et inférieur à un minimum configuré localement que l'ESC PEUT rejeter la publication avec une réponse de 423 (Intervalle trop bref) et sauter les étapes restantes. Cette réponse DOIT contenir un champ d'en-tête Min-Expires qui déclare l'intervalle d'expiration minimum que l'ESC veut honorer.

5. L'ESC traite l'état d'événement publié contenu dans le corps de la demande PUBLISH. Si le type de contenu de la demande ne correspond pas au paquetage d'événements, ou n'est pas compris par l'ESC, celui-ci DOIT rejeter la demande avec une réponse appropriée, comme 415 (Type de support non accepté) et sauter les étapes restantes.
   * L'ESC mémorise l'état d'événement délivré dans le corps de la demande PUBLISH et identifié par l'étiquette d'entité associée, et met à jour tout état d'événement existant pour cette étiquette d'entité. La valeur d'expiration est réglée à l'intervalle d'expiration choisi.
   * Si la demande n'a pas de corps de message et ne contient pas d'étiquette d'entité, l'ESC DEVRAIT rejeter la demande avec une réponse appropriée, comme 400 (Demande invalide) et sauter les étapes restantes. Autrement, au cas où une politique locale d'ESC ou le paquetage d'événements a défini une sémantique pour une publication initiale ne contenant pas de corps de message, l'ESC PEUT l'accepter.
   * Autrement, l'état d'événement identifié par l'étiquette d'entité est rafraîchi, en réglant la valeur d'expiration à l'intervalle d'expiration choisi.
   * Si l'intervalle d'expiration choisi a la valeur spéciale de "0", l'état d'événement identifié par l'étiquette d'entité DOIT être immédiatement supprimé. L'ESC NE DOIT PAS mémoriser un état d'événement par suite d'une telle demande.

   Le traitement de la demande PUBLISH DOIT être atomique. Si des erreurs internes (comme l'incapacité à accéder à une base de données arrière) survient avant l'achèvement du traitement, la publication NE DOIT PAS réussir, et l'ESC DOIT échouer avec une réponse d'erreur appropriée, comme une 504 (Fin de temporisation du serveur) et sauter la dernière étape.

6. L'ESC retourne une réponse 200 (OK). La réponse DOIT contenir un champ d'en-tête Expires indiquant l'intervalle d'expiration choisi par l'ESC. La réponse DOIT aussi contenir un champ d'en-tête SIP-ETag qui contient une seule étiquette d'entité identifiant la publication. L'ESC DOIT générer une nouvelle étiquette d'entité pour chaque publication réussie, remplaçant toute étiquette d'entité antérieure associée à cet état d'événement. L'étiquette d'entité générée DOIT être unique par rapport à toute autre étiquette d'entité actuellement allouée à l'état d'événement associé à cet URI de

demande, et DOIT être différent de toute étiquette d'entité allouée précédemment à l'état d'événement pour cet URI de demande. Voir au paragraphe 8.3 plus d'informations sur le traitement par l'ESC des étiquettes d'entité.

7. Traitement des demandes OPTIONS

Un client peut sonder l'ESC sur la prise en charge de PUBLISH en utilisant la demande OPTIONS définie dans SIP [RFC3261]. L'ESC traite les demandes OPTIONS comme défini au paragraphe 11.2 de la [RFC3261]. Dans la réponse à une demande OPTIONS, l'ESC DEVRAIT inclure "PUBLISH" à la liste des méthodes permises dans le champ d'en-tête Allow. Aussi, il DEVRAIT faire la liste des paquetages d'événements pris en charge dans le champ d'en-tête Allow-Events.

Le champ d'en-tête Allow peut aussi être utilisé pour annoncer spécifiquement la prise en charge des messages PUBLISH lors de l'enregistrement. (Voir les détails dans les capacités de SIP [RFC3840]).

8. Utilisation des étiquettes d'entité dans PUBLISH

Cette section donne une vue d'ensemble générale de l'utilisation des étiquettes d'entité dans PUBLISH. Elle est de nature informative et ne contient donc aucune description de protocole normative.

8.1 Notes générales

Le mécanisme PUBLISH utilise les étiquettes d'entités, comme défini dans HTTP/ 1.1 [RFC2616]. Bien que la fonctionnalité principale soit préservée, la syntaxe et la sémantique des étiquettes d'entités et des champs d'en-tête correspondants sont spécifiquement adaptées à l'utilisation avec la méthode PUBLISH. Les principales différences sont :

- La syntaxe des étiquettes d'entité est un jeton plutôt qu'une chaîne entre guillemets. Il n'y a pas de préfixe défini pour indiquer une étiquette d'entité faible.

- Une précondition PUBLISH ne peut s'appliquer qu'à une seule étiquette d'entité, de sorte que des préconditions de demande avec plusieurs étiquette d'entités ne sont pas permises.

- Une précondition de demande ne peut pas s'appliquer à "toute" entité, à savoir qu'il n'y a pas de valeur d'étiquette d'entité spéciale "*" définie pour PUBLISH.

- Tandis que dans HTTP/1.1, retourner une étiquette d'entité est facultatif pour les serveurs d'origine, dans PUBLISH, les ESC sont obligés de retourner toujours une étiquette d'entité pour une publication réussie.

Le principal motif de l'adaptation ci-dessus est que PUBLISH est par conception un PUT HTTP, pour lequel seul un sous ensemble des caractéristiques en validation d'antémémoire en utilisant une étiquette d'entités est permis dans HTTP/1.1. Il y a peu de sens à permettre des caractéristiques autres que ce sous ensemble pour la publication d'état d'événement.

Pour rendre évident que l'usage de l'étiquette d'entités dans PUBLISH est similaire mais pas identique à celui de HTTP/1.1, on n'a pas adopté les noms de champ d'en-tête directement de HTTP/1.1, mais on a plutôt créé des noms similaires mais distincts, comme on le verra à la Section 11.

8.2 Utilisation du client

À chaque publication réussie est allouée une étiquette d'entité qui est ensuite livrée à l'EPA dans la réponse à la demande PUBLISH. L'EPA a besoin de mémoriser cette étiquette d'entité, en remplaçant toute étiquette d'entité précédente pour cet état d'événement. Si une demande échoue avec une réponse 412 (Échec de demande conditionnelle) l'EPA élimine l'étiquette d'entité qui a causé l'échec.

Les étiquettes d'entité sont des jetons opaques pour l'EPA. L'EPA ne peut pas déduire d'autre signification d'une étiquette d'entité au delà d'un simple identifiant, ou supposer un formatage spécifique. Une étiquette d'entité peut être un compteur à croissance monotone, mais il peut aussi être un jeton totalement aléatoire. Il appartient à la mise en œuvre d'ESC de choisir le format d'une étiquette d'entité.

8.3 Utilisation du serveur

Les étiquettes d'entité sont générées et entretenues par l'ESC. Elles font partie de l'état entretenu par l'ESC qui inclut aussi l'état d'événement réel et son intervalle d'expiration restant. Une étiquette d'entité est générée et mémorisée pour chaque publication d'état d'événement réussie, et est retournée à l'EPA dans une réponse 200 (OK). Chaque publication d'état d'événement provenant de l'EPA qui met à jour une précédente publication va inclure une étiquette d'entité que l'ESC peut utiliser comme clé de recherche dans l'ensemble des publications actives.

La façon dont une étiquette d'entité est générée est une décision de la mise en œuvre. Une façon possible de générer une étiquette d'entité est de la mettre en œuvre comme un compteur d'entiers qui est incrémenté de un pour chaque publication traitée avec succès. D'autres façons tout aussi valides pour générer des étiquettes d'entité existent, et le présent document ne fait aucune recommandation ou n'a de préférence pour une particulière.

9. Contrôle du taux de publication

En tant qu'entité responsable de l'agrégation des informations d'état de sources potentiellement nombreuses, l'ESC peut être soumis à des quantités considérables de trafic de publication. Il y a des façons de réduire la quantité de demandes PUBLISH que l'ESC reçoit :

- Le choix de l'intervalle d'expiration pour une publication peut être affecté par l'ESC. Il peut insister pour qu'un EPA choisisse une valeur d'expiration plus longue que ce qu'il suggère, dans le cas où la valeur locale d'expiration minimale par défaut de l'ESC n'est pas atteinte. Conserver une valeur d'expiration minimum par défaut plus longue chez l'ESC réduit le taux auquel les publications sont rafraîchies.

- Une autre façon de réduire le trafic de publication est d'utiliser un repoussoir de niveau SIP pour étouffer une source spécifique de trafic de publication. Pour repousser les publications d'une source particulière, l'ESC PEUT répondre à une demande PUBLISH par un 503 (Service indisponible) comme défini dans la [RFC3261]. Cette réponse DEVRAIT contenir un champ d'en-tête Retry-After indiquant l'intervalle de temps pendant lequel la source de publication est obligée d'attendre avant d'envoyer une autre demande PUBLISH.

Au moment de la rédaction de la présente spécification, des travaux commencent sur la charge de gestion dans SIP, qui pourraient fournir de nouveaux outils pour la charge de gestion dans les systèmes de publication d'état d'événement.

10. Considérations sur les paquetages d'événement qui utilisent PUBLISH

Cette section expose plusieurs questions qui devraient être prises en considération pour l'application du mécanisme PUBLISH aux paquetages d'événements. Elle montre aussi comment ces questions sont traitées lors de l'utilisation de PUBLISH pour la publication de présence.

Toute future spécification de paquetage d'événements DEVRAIT inclure une discussion de ses considérations sur l'utilisation de PUBLISH. Au minimum, ces considérations DEVRAIENT traiter les questions présentées ici, et PEUVENT inclure des considérations supplémentaires.

10.1 Corps PUBLISH

Le corps de la demande PUBLISH porte normalement l'état d'événement publié. Toute application du mécanisme PUBLISH pour un certain paquetage d'événements DOIT définir quel ou quels types de contenu sont attendus dans les demandes PUBLISH. Chaque paquetage d'événements DOIT aussi décrire la sémantique associée à un type de contenu, et DOIT prescrire un type MIME par défaut de mise en œuvre obligatoire.

Le présent document définit la sémantique de la demande de publication de présence (paquetage d'événements "presence") lorsque le profil commun pour Presence (CPP, *Common Profile for Presence*) du format de données d'informations de présence (PIDF, *Presence Information Data Format*) [RFC3863] est utilisé. Un PUA qui utilise PUBLISH pour publier l'état de présence au PA DOIT prendre en charge le format de présence PIDF. Il PEUT prendre en charge d'autres formats.

10.2 Corps de réponse PUBLISH

La réponse à une demande PUBLISH indique si la demande a réussi ou non. En général, le corps d'une telle réponse va être

vide sauf si le paquetage d'événements définit une signification explicite pour un tel corps.

Il n'y a pas une telle signification pour le corps d'une réponse à une publication de présence.

10.3 Sources multiples pour l'état d'événement

Pour certains paquetages d'événements, le modèle sous-jacent est celui d'une seule entité responsable de l'agrégation des états d'événement (l'ESC), et de plusieurs sources, parmi lesquelles seules quelques unes peuvent utiliser le mécanisme PUBLISH.

Noter que les sources pour un état d'événement autre que celui qui utilise le mécanisme PUBLISH sont explicitement permises. Cependant, il sort du domaine d'application du présent document de définir de telles interfaces.

Les paquetages d'événements qui utilisent le mécanisme PUBLISH DEVRAIENT décrire si ce modèle de publication d'état d'événement est applicable, et PEUT décrire les mécanismes spécifiques utilisés pour agréger les publications à partir de sources multiples.

Pour presence, un PUA peut publier l'état de présence pour juste un sous ensemble des tuplets qui peuvent être composés dans le document de présence que les observateurs reçoivent dans un NOTIFY. Le mécanisme par lequel l'ESC agrège ces informations est l'affaire de la politique locale et sort du domaine d'application de la présente spécification.

10.4 Segmentation d'état d'événement

Pour certains paquetages d'événements, il existe une décomposition naturelle de l'état d'événement en segments. Chaque segment est défini comme une des potentiellement nombreuses sections identifiables dans l'état d'événement publié. Tout paquetage d'événements dont le type de contenu accepte une telle segmentation de l'état d'événement, DEVRAIT décrire la façon dont ces segments d'état d'événement sont identifiés par l'ESC.

Dans une publication de présence, l'EPA DOIT garder les attributs "id" des tuplets cohérents dans le contexte d'une étiquette d'entité. Si une publication modifie le contenu d'un tuplet, celui-ci DOIT conserver son "id" d'origine. L'ESC va interpréter chaque tuplet dans le contexte de l'étiquette d'entité avec laquelle est arrivée la demande. Un tuplet dont le "id" manque par rapport à la publication d'origine sera considéré comme supprimé. De même, un tuplet est interprété comme ajouté si son attribut "id" est un de ceux que la publication d'origine ne contenait pas.

10.5 Taux de publication

Le contrôle du taux de publication est discuté à la Section 9. Les paquetages d'événements individuels PEUVENT à leur tour définir des recommandations (de force DEVRAIT ou DOIT) sur les taux absolus maximum auxquels les publications peuvent être générées par un seul EPA.

Il n'y a pas de recommandation de limitation de taux pour la publication de présence.

11. Définitions d'élément de protocole

Le présente section décrit les extensions requises pour la publication d'événement dans SIP.

11.1 Nouvelles méthodes

11.1.1 Méthode PUBLISH

"PUBLISH" est ajouté à la définition de l'élément "Method" dans la grammaire de message SIP. Comme avec toutes les autres méthodes SIP, le nom de la méthode est sensible à la casse. PUBLISH est utilisé pour publier l'état d'événement à une entité responsable de composer cet état d'événement.

Le Tableau 2 étend le Tableau 2 de la [RFC3261] en ajoutant une colonne supplémentaire, définissant les champs d'en-tête qui peuvent être utilisés dans les demandes et réponses PUBLISH. Les clés de ce tableau sont spécifiées à la Section 20 de la [RFC3261].

Champ d'en-tête	où	PUBLISH
Accept	R	o
Accept	2xx	-
Accept	415	m*
Accept-Encoding	R	o
Accept-Encoding	2xx	-
Accept-Encoding	415	m*|
Accept-Language	R	o
Accept-Language	2xx	-
Accept-Language	415	m*
Alert-Info		-
Allow	R	o
Allow	r	o
Allow	405	m
Allow-Events	R	o
Allow-Events	489	m
Authentication-Info	2xx	o
Authorization	R	o
Call-ID	c	m
Call-Info		o
Contact	R	-
Contact	1xx	-
Contact	2xx	-
Contact	3xx	o
Contact	485	o
Content-Disposition		o
Content-Encoding		o
Content-Language		o
Content-Length		t
Content-Type		*
CSeq	c	m
Date		o
Event	R	m
Error-Info	300-699	o
Expires		o
Expires	2xx	m
From	c	m
In-Reply-To	R	-
Max-Forwards	R	m
Min-Expires	423	m
MIME-Version		o
Organization		o
Priority	R	o
Proxy-Authenticate	407	m
Proxy-Authenticate	401	o
Proxy-Authorization	R	o
Proxy-Require	R	o
Record-Route		-
Reply-To		-
Require		o
Retry-After	404,413,480,486	o
Retry-After	500,503	o
Retry-After	600,603	o
Route	R	c
Server	r	o
Subject	R	o
Supported	R	o
Supported	2xx	o
Timestamp		o
To	c(1)	m
Unsupported	420	o
User-Agent		o

Via	R	m
Via	rc	m
Warning	r	o
WWW-Authenticate	401	m
WWW-Authenticate	407	o

Tableau 2 : Résumé des champs d'en-tête

11.2 Nouveaux codes de réponse

11.2.1 Code de réponse "412 Échec de demande conditionnelle"

La réponse 412 (Échec de demande conditionnelle) est ajoutée à la définition du champ d'en-tête "Client-Error". 412 (Échec de demande conditionnelle) est utilisé pour indiquer que la précondition donnée pour la demande a échoué.

11.3 Nouveaux champs d'en-tête

Les Tableaux 3, 4, et 5 étendent le Tableau 2 de SIP [RFC3261], amendé par les changements du paragraphe 11.1.

Champ d'en-tête	où	mandataire	ACK	BYE	CAN	INF	INV
SIP-ETag	2xx		-	-	-	-	-
SIP-If-Match	R		-	-	-	-	-

Tableau 3 : Résumé des champ d'en-têtes, A à O

Champ d'en-tête	où	mandataire	NOT	OPT	PRA	REG	SUB	UPD	MSG	REF	PUBLISH
SIP-ETag	2xx		-	-	-	-	-	-	-	-	m
SIP-If-Match	R		-	-	-	-	-	-	-	-	o

Tableau 4 : Résumé des champ d'en-têtes, P--Z

11.3.1 Champ d'en-tête "SIP-ETag"

SIP-ETag est ajouté à la définition de l'élément "general-header" dans la grammaire de message SIP. L'usage de cet en-tête est décrit aux sections 4 et 6.

11.3.2 Champ d'en-tête "SIP-If-Match"

SIP-If-Match est ajouté à la définition de l'élément "general-header" dans la grammaire de message SIP. L'usage de cet en-tête est décrit aux sections 4 et 6.

12. Définitions de BNF augmenté

Cette section décrit les extensions de syntaxe exigées pour la publication d'événement dans SIP. Les définitions de syntaxe formelle décrites dans cette section sont exprimées dans le format de BNF augmenté de la [RFC2234] utilisé dans SIP [RFC3261], et contient des références aux éléments qui y sont définis.

PUBLISHm = %x50.55.42.4C.49.53.48 ; PUBLISH en majuscules.
extension-method = PUBLISHm / jeton
SIP-ETag = "SIP-ETag" HCOLON étiquette d'entité
SIP-If-Match = "SIP-If-Match" HCOLON étiquette d'entité
étiquette d'entité = jeton

13. Considérations relatives à l'IANA

Le présent document enregistre un nouveau nom de méthode, un nouveau code de réponse et deux nouveaux noms de champ d'en-tête.

13.1 Méthodes

Le présent document enregistre une nouvelle méthode SIP, définie par les informations suivantes, qui ont été ajoutées aux sous registres method et response-code sous http://www.iana.org/assignments/sip-parameters.

Nom de méthode : PUBLISH
Référence : [RFC3903]

13.2 Codes de réponse

Le présent document enregistre un nouveau code de réponse. Ce code de réponse est défini par les informations suivantes, qui ont été ajoutées aux sous registres method et response-code sous http://www.iana.org/assignments/sip-parameters.

Numéro de code de réponse : 412
Phrase de cause par défaut : Échec de demande conditionnelle

13.3 Noms de champs d'en-tête

Le présent document enregistre deux nouveaux noms de champ d'en-tête SIP. Ces en-têtes sont définis par les informations suivantes, qui ont été ajoutées au sous registre header sous http://www.iana.org/assignments/sip-parameters.

Nom d'en-tête : SIP-ETag
Forme compacte : (aucune)
Nom d'en-tête : SIP-If-Match
Forme compacte : (aucune)

14. Considérations pour la sécurité

14.1 Contrôle d'accès

Comme un état d'événement peut être considéré comme une information sensible, l'ESC devrait avoir la capacité d'accepter de façon sélective les publications de seules sources autorisées, sur la base de l'identité de l'EPA. L'agent d'état DEVRAIT authentifier l'EPA, et DEVRAIT appliquer ses politiques d'autorisation (par exemple, sur la base de listes de contrôle d'accès) à toutes les demandes. Le modèle de composition ne fait pas comme hypothèse que toutes les sources d'entrées pour un ESC sont sur le même réseau, ni dans le même domaine administratif. Les ESC et les EPA DOIVENTT mettre en œuvre Digest pour l'authentification des demandes PUBLISH, comme défini dans la [RFC3261]. Les méthodes exactes pour créer et manipuler les politiques de contrôle d'accès chez l'ESC sortent du domaine d'application du présent document.

14.2 Attaques de déni de service

La création d'état chez l'ESC à réception d'une demande PUBLISH peut être utilisée par des attaquants pour consommer les ressources de la machine d'une victime, la rendant éventuellement inutilisable. Pour réduire les chances de succès d'une telle attaque, les mises en œuvre d'ESC DEVRAIENT exiger l'authentification des demandes PUBLISH. Les mises en œuvre DOIVENT prendre en charge l'authentification par résumé *(Digest)* comme défini dans la [RFC3261]. Aussi, l'ESC DEVRAIT réduire les publications entrantes et les notifications correspondantes résultant de changements d'état d'événement. Comme première étape, un choix attentif des valeurs minimum par défaut du champ d'en-tête Expires pour les paquetages d'événements pris en charge par un ESC peut aider à limiter les rafraîchissements d'état d'événement.

On conseille une logique supplémentaire de réduction et de non rebond chez l'ESC pour réduire encore le trafic de notification produit par suite d'une demande PUBLISH.

14.3 Attaques de répétition

La répétition d'une demande PUBLISH peut avoir des effets dommageables. Un attaquant peut être capable d'effectuer toute publication d'état d'événement dont il a vu qu'elle était effectuée à un moment dans le passé, en répétant cette demande PUBLISH. Entre autres choses, une telle répétition du message peut être utilisée pour imiter de vieilles informations d'état d'événement, bien qu'un mécanisme d'identification de version, par exemple, un horodatage, dans les informations d'état puisse aider à atténuer de telles attaques.

Pour empêcher les attaques en répétition, les mises en œuvre DOIVENT prendre en charge l'authentification par résumé avec protection contre la répétition, comme défini dans la [RFC3261]. D'autres mécanismes pour contrer les attaques en répétition sont exposés dans SIP [RFC3261].

14.4 Attaques par interposition

Même avec l'authentification, les attaques par interposition utilisant PUBLISH peuvent être effectuées pour installer des informations arbitraires d'état d'événement, pour modifier ou supprimer des informations existantes d'état d'événement dans des publications, ou même retirer carrément un état d'événement chez un ESC.

Pour empêcher de telles attaques, les mises en œuvre DEVRAIENT, au minimum, assurer la protection de l'intégrité à travers les champs d'en-tête To, From, Event, SIP-If-Match, Route, et Expires et les corps des demandes PUBLISH.

Si l'ESC reçoit dans une demande PUBLISH un état d'événement qui est protégé en intégrité en utilisant une association de sécurité qui n'est pas avec l'ESC (par exemple, la protection de l'intégrité est appliquée de bout en bout, de l'éditeur à l'abonné) l'agent d'état couplé avec l'ESC NE DOIT PAS modifier l'état d'événement avant de l'exposer aux abonnés de cet état d'événement dans les demandes NOTIFY. Ceci est destiné à préserver l'intégrité de bout en bout de l'état d'événement.

Pour la protection de l'intégrité, les ESC DOIVENT mettre en œuvre TLS [RFC2246], et DOIVENT prendre en charge l'authentification mutuelle et unidirectionnelle, et DOIVENT aussi prendre en charge le schéma d'URI SIPS défini dans SIP [RFC3261]. Les EPA DEVRAIENT être capables d'initier TLS et DEVRAIENT prendre en charge le schéma d'URI SIPS. Les ESC et les EPA PEUVENT prendre en charge S/MIME [RFC3851] pour la protection de l'intégrité, comme défini dans SIP [RFC3261].

14.5 Confidentialité

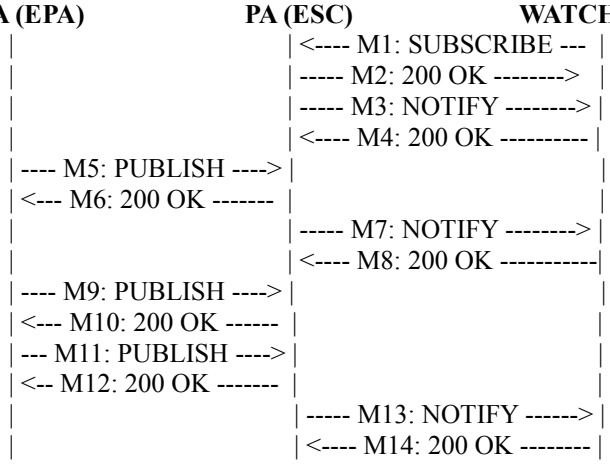
Les informations d'état contenues dans un message PUBLISH peuvent éventuellement contenir des informations sensibles. Les mises en œuvre PEUVENT chiffrer de telles informations pour assurer leur confidentialité.

Pour assurer la confidentialité, les ESC DOIVENT mettre en œuvre TLS [RFC2246], DOIVENT prendre en charge l'authentification mutuelle et unidirectionnelle, et DOIVENT aussi prendre en charge le schéma d'URI SIPS défini dans SIP [RFC3261]. Les EPA DEVRAIENT être capables d'initier TLS et DEVRAIENT prendre en charge le schéma d'URI SIPS. Les ESC et les EPA PEUVENT prendre en charge S/MIME [RFC3851] pour le chiffrement des informations d'état d'événement, comme défini dans SIP [RFC3261].

15. Exemples

Cette section donne un exemple d'utilisation de la méthode PUBLISH pour publier un document de présence d'un agent de présence d'utilisateur (PUA) pour un agent de présence (PA). L'observateur dans cet exemple s'abonne aux informations de présence de la présentité à partir de l'agent de présence. Le PUA peut aussi SUBSCRIBE *(souscrire)* à sa propre présence pour voir l'état de présence composite exposé par le PA. Ceci est une étape facultative mais probable pour le PUA, et n'est pas montré dans cet exemple.

Lorsque la valeur du champ d'en-tête Content-Length est "..." cela signifie que la valeur devrait être la longueur calculée du corps.



Flux de messages :

M1 : L'observateur initie une nouvelle souscription à l'agent de présence de presentity@example.com.

SUBSCRIBE sip:presentity@example.com SIP/2.0
Via: SIP/2.0/UDP host.example.com;branch=z9hG4bKnashds7
To: <sip:presentity@example.com>
From: <sip:watcher@example.com>;tag=12341234
Call-ID: 12345678@host.example.com
CSeq: 1 SUBSCRIBE
Max-Forwards: 70
Expires: 3600
Event: presence
Contact: sip:user@host.example.com
Content-Length: 0

M2 : L'agent de présence pour presentity@example.com traite la demande de souscription et crée un nouvel abonnement. Une réponse 200 (OK) est envoyée pour confirmer l'abonnement.

SIP/2.0 200 OK
Via: SIP/2.0/UDP host.example.com;branch=z9hG4bKnashds7
;received=192.0.2.1
To: <sip:presentity@example.com>;tag=abcd1234
From: <sip:watcher@example.com>;tag=12341234
Call-ID: 12345678@host.example.com
CSeq: 1 SUBSCRIBE
Contact: sip:pa.example.com
Expires: 3600
Content-Length: 0

M3 : Pour achever le procès, l'agent de présence envoie à l'observateur un NOTIFY avec l'état de présence actuel de la présentité.

NOTIFY sip:user@host.example.com SIP/2.0
Via: SIP/2.0/UDP pa.example.com;branch=z9hG4bK8sdf2
To: <sip:watcher@example.com>;tag=12341234
From: <sip:presentity@example.com>;tag=abcd1234
Call-ID: 12345678@host.example.com
CSeq: 1 NOTIFY
Max-Forwards: 70
Event: presence
Subscription-State: active; expires=3599
Contact: sip:pa.example.com
Content-Type: application/pidf+xml
Content-Length: ...

[Document PIDF]

M4 : L'observateur confirme la réception de la demande NOTIFY.

SIP/2.0 200 OK
Via: SIP/2.0/UDP pa.example.com;branch=z9hG4bK8sdf2
;received=192.0.2.2
To: <sip:watcher@example.com>;tag=12341234
From: <sip:presentity@example.com>;tag=abcd1234
Call-ID: 12345678@host.example.com
CSeq: 1 NOTIFY

M5 : Un agent de présence d'utilisateur (agissant pour la présentité) initie une demande PUBLISH à l'agent de présence afin de le mettre à jour avec les nouvelles informations de présence. Le champ d'en-tête Expires indique la durée suggérée de cet état conditionnel d'événement.

PUBLISH sip:presentity@example.com SIP/2.0

Via: SIP/2.0/UDP pua.example.com;branch=z9hG4bK652hsge
To: <sip:presentity@example.com>
From: <sip:presentity@example.com>;tag=1234wxyz
Call-ID: 81818181@pua.example.com
CSeq: 1 PUBLISH
Max-Forwards: 70
Expires: 3600
Event: presence
Content-Type: application/pidf+xml
Content-Length: ...

[Document PIDF publié]

M6 : L'agent de présence reçoit, et accepte la publication de présence. Les données publiées sont incorporées dans les informations de présence de la présentité.

SIP/2.0 200 OK
Via: SIP/2.0/UDP pua.example.com;branch=z9hG4bK652hsge
;received=192.0.2.3
To: <sip:presentity@example.com>;tag=1a2b3c4d
From: <sip:presentity@example.com>;tag=1234wxyz
Call-ID: 81818181@pua.example.com
CSeq: 1 PUBLISH
SIP-ETag: dx200xyz
Expires: 1800

M7 : L'agent de présence détermine qu'un changement rapportable a été fait aux informations de présence de la présentité, et envoie une nouvelle notification de présence à l'observateur.

NOTIFY sip:user@host.example.com SIP/2.0
Via: SIP/2.0/UDP pa.example.com;branch=z9hG4bK4cd42a
To: <sip:watcher@example.com>;tag=12341234
From: <sip:presentity@example.com>;tag=abcd1234
Call-ID: 12345678@host.example.com
CSeq: 2 NOTIFY
Max-Forwards: 70
Event: presence
Subscription-State: active; expires=3400
Contact: sip:pa.example.com
Content-Type: application/pidf+xml
Content-Length: ...

[Nouveau document PIDF]

M8 : L'observateur confirme la réception de la demande NOTIFY.

SIP/2.0 200 OK
Via: SIP/2.0/UDP pa.example.com;branch=z9hG4bK4cd42a
;received=192.0.2.2
To: <sip:watcher@example.com>;tag=12341234
From: <sip:presentity@example.com>;tag=abcd1234
Call-ID: 12345678@host.example.com
CSeq: 2 NOTIFY
Content-Length: 0

M9 : Le PUA détermine que l'état d'événement qu'il a publié précédemment va arriver à expiration, et il rafraîchit cet état d'événement.

PUBLISH sip:presentity@example.com SIP/2.0
Via: SIP/2.0/UDP pua.example.com;branch=z9hG4bK771ash02
To: <sip:presentity@example.com>
From: <sip:presentity@example.com>;tag=1234kljk
Call-ID: 98798798@pua.example.com

CSeq: 1 PUBLISH
Max-Forwards: 70
SIP-If-Match: dx200xyz
Expires: 3600
Event: presence
Content-Length: 0

M10 : L'agent de présence reçoit et accepte les rafraîchissements de la publication. Les temporisateurs concernant l'expiration de l'état d'événement spécifique identifié par l'étiquette d'entité sont mis à jour. Comme toujours, l'ESC retourne une étiquette d'entité dans la réponse à une PUBLISH réussie. Noter qu'aucun changement réel d'état n'a eu lieu de sorte que l'observateur ne va recevoir aucun NOTIFY.

SIP/2.0 200 OK
Via: SIP/2.0/UDP pua.example.com;branch=z9hG4bK771ash02
;received=192.0.2.3
To: <sip:presentity@example.com>;tag=2affde434
From: <sip:presentity@example.com>;tag=1234kljk
Call-ID: 98798798@pua.example.com
CSeq: 1 PUBLISH
SIP-ETag: kwj449x
Expires: 1800

M11 : Le PUA de la présentité détecte un changement dans l'état de présence de l'usager. Il initie une demande PUBLISH à l'agent de présence pour modifier les informations de présence publiées avec le changement récent.

PUBLISH sip:presentity@example.com SIP/2.0
Via: SIP/2.0/UDP pua.example.com;branch=z9hG4bKcdad2
To: <sip:presentity@example.com>
From: <sip:presentity@example.com>;tag=54321mm
Call-ID: 5566778@pua.example.com
CSeq: 1 PUBLISH
Max-Forwards: 70
SIP-If-Match: kwj449x
Expires: 3600
Event: presence
Content-Type: application/pidf+xml
Content-Length: ...

[Document PIDF publié]

M12 : L'agent de présence reçoit, et accepte la modification de publication. Les données publiées sont incorporées dans les informations de présence de la présentité, mettant à jour la publication précédente à partir du même PUA.

SIP/2.0 200 OK
Via: SIP/2.0/UDP pua.example.com;branch=z9hG4bKcdad2
;received=192.0.2.3
To: <sip:presentity@example.com>;tag=effe22aa
From: <sip:presentity@example.com>;tag=54321mm
Call-ID: 5566778@pua.example.com
CSeq: 1 PUBLISH
SIP-ETag: qwi982ks
Expires: 3600

M13 : L'agent de présence détermine qu'un changement rapportable a été fait au document de présence de la présentité, et il envoie une nouvelle notification de présence à tous les abonnés actifs.

NOTIFY sip:user@host.example.com SIP/2.0
Via: SIP/2.0/UDP pa.example.com;branch=z9hG4bK32defd3
To: <sip:watcher@example.com>;tag=12341234
From: <sip:presentity@example.com>;tag=abcd1234
Call-ID: 12345678@host.example.com
CSeq: 2 NOTIFY
Max-Forwards: 70

Event: presence
Subscription-State: active; expires=3400
Contact: sip:pa.example.com
Content-Type: application/pidf+xml
Content-Length: ...

[Nouveau document PIDF]

M14 : L'observateur confirme la réception de la demande NOTIFY.

SIP/2.0 200 OK
Via: SIP/2.0/UDP pa.example.com;branch=z9hG4bK32defd3
;received=192.0.2.3
To: <sip:watcher@example.com>;tag=12341234
From: <sip:presentity@example.com>;tag=abcd1234
Call-ID: 12345678@host.example.com
CSeq: 2 NOTIFY
Content-Length: 0

16. Contributeurs

Les contributeurs à l'origine de la présente spécification sont :

Ben Campbell, Estacado Systems
Sean Olson, Microsoft
Jon Peterson, Neustar, Inc.
Jonathan Rosenberg, dynamicsoft
Brian Stucker, Nortel Networks, Inc.

17. Remerciements

Les auteurs tiennent à remercier les membres du groupe de travail SIMPLE pour leur effort collectif, et plus particulièrement les personnes suivantes pour leur relecture et leur soutien à cet ouvrage : Henning Schulzrinne, Paul Kyzivat, Hisham Khartabil, George Foti, Keith Drage, Samir Srivastava, Arun Kumar, Adam Roach, Pekka Pessi, Kai Wang, Cullen Jennings, Mikko Lonnfors, Eva-Maria Leppanen, Ernst Horvath, Thanos Diacakis, Oded Cnaan, Rohan Mahy, et Dean Willis.

18. Références

18.1 Références normatives

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

[RFC2234] D. Crocker et P. Overell, "BNF augmenté pour les spécifications de syntaxe : ABNF", novembre 1997. (*Obsolète, voir* RFC5234)

[RFC2246] T. Dierks et C. Allen, "Protocole TLS version 1.0", janvier 1999.

[RFC2778] M. Day, J. Rosenberg et H. Sugano, "Modèle pour Presence et la messagerie instantanée", février 2000.

[RFC3261] J. Rosenberg et autres, "SIP : Protocole d'initialisation de session", juin 2002. (*Mise à jour par* RFC3265, RFC3853, RFC4320, RFC4916, RFC5393, RFC6665)

[RFC3265] A.B. Roach, "Notification d'événement spécifique du protocole d'initialisation de session (SIP)", juin 2002. (*MàJ par* RFC6446) (*Remplacée par la* RFC6665)

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

[RFC3856] J. Rosenberg, "Paquetage d'événement Presence pour le protocole d'initialisation de session (SIP)", août 2004.

[RFC3863] H. Sugano et autres, "Format des données d'information de présence (PIDF)", août 2004.

18.2 Références informatives

[RFC2616] R. Fielding et autres, "Protocole de transfert hypertexte -- HTTP/1.1", juin 1999. *(D.S., MàJ par* 2817, 6585)

[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

[RFC3842] R. Mahy, "Paquetage d'événement Résumé de message et Indication de message en attente pour le protocole d'initialisation de session (SIP)", août 2004.

[SIMPLE] B. Campbell, "SIMPLE Presence Publication Requirements", Travail en cours, février 2003.

Adresse de l'auteur

Aki Niemi (éditeur)
Nokia
P.O. Box 407
NOKIA GROUP, FIN 00045
Finland
téléphone : +358 50 389 1644
mél : aki.niemi@nokia.com

Déclaration complète de droits de reproduction

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