

Groupe de travail Réseau
Request for Comments : 3680
 Catégorie : En cours de normalisation

J. Rosenberg, dynamicsoft
 mars 2004
 Traduction Claude Brière de L'Isle

Paquetage d'événement du protocole d'initialisation de session (SIP) pour les enregistrements

Statut du présent mémoire

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

Notice de copyright

Copyright (C) The Internet Society (2004).

Résumé

Le présent document définit un paquetage d'événement du protocole d'initialisation de session (SIP, *Session Initiation Protocol*) pour les enregistrements. Par sa méthode REGISTER, SIP permet aux agents d'utilisateur de créer, modifier, et supprimer les enregistrements. Les enregistrements peuvent aussi être altérés par les administrateurs afin de mettre en application une politique. Il en résulte que ces enregistrements représentent un élément d'état dans le réseau qui peut changer de façon dynamique. Il y a de nombreux cas où un agent d'utilisateur aimerait que lui soient notifiés les changements de cet état. Le présent paquetage d'événement définit un mécanisme par lequel ces agents d'utilisateur peuvent demander et obtenir une telle notification.

Table des Matières

1. Introduction.....	2
2. Terminologie.....	2
3. Scénarios d'utilisation.....	2
3.1 Ré-authentification forcée.....	2
3.2 Composition de présence.....	2
3.3 Notices de bienvenue.....	3
4. Définition du paquetage.....	3
4.1 Nom du paquetage d'événement.....	3
4.2 Paramètres du paquetage d'événement.....	3
4.3 Corps SUBSCRIBE.....	3
4.4 Durée de souscription.....	4
4.5 Corps NOTIFY.....	4
4.6 Traitement du notificateur des demandes SUBSCRIBE.....	4
4.7 Génération par le notificateur des demandes NOTIFY.....	4
4.8 Traitement des demandes NOTIFY par le souscripteur.....	6
4.9 Traitement des demandes fourchés.....	6
4.10 Taux des notifications.....	7
4.11 Agents d'état.....	7
5. Informations d'enregistrement.....	7
5.1 Structure des informations d'enregistrement.....	7
5.2 Calcul des enregistrements à partir du document.....	8
5.3 Exemple.....	9
5.4 Schéma XML.....	9
6. Exemple de flux d'appel.....	11
7. Considérations pour la sécurité.....	13
8. Considérations relatives à l'IANA.....	13
8.1 Enregistrement du paquetage d'événements SIP.....	13
8.2 Enregistrement MIME de application/reginfo+xml.....	13
8.3 Enregistrement du sous espace de noms d'URN pour urn:ietf:params:xml:ns:reginfo.....	14
9. Références.....	14
9.1 Références normatives.....	14
9.2 Références pour information.....	15
10. Contributeurs.....	15
11. Remerciements.....	15

12. Adresse de l'auteur.....	15
13. Déclaration complète de droits de reproduction.....	16

1. Introduction

Le protocole d'initialisation de session (SIP) [RFC3261] procure toutes les fonctions nécessaires pour l'établissement et la maintenance des sessions de communications entre les usagers. Une des fonctions qu'il fournit est une opération d'enregistrement. Un enregistrement est un lien entre un URI SIP, appelé une adresse d'enregistrement, et un ou plusieurs URI de contact. Ces URI de contact représentent des ressources supplémentaires qui peuvent être contactées afin de joindre l'utilisateur identifié par l'adresse d'enregistrement. Lorsque un mandataire reçoit une demande dans son domaine d'administration, il utilise l'URI de demande comme adresse d'enregistrement, et utilise les contacts liés à l'adresse d'enregistrement pour transmettre (ou rediriger) la demande.

La méthode SIP REGISTER donne le moyen à un agent d'utilisateur de manipuler les enregistrements. Des contacts peuvent être ajoutés ou retirés, et l'ensemble actuel de contacts peut être établi. Les enregistrements peuvent aussi changer par suite de la politique d'administration. Par exemple, si un utilisateur est suspecté de fraude, son enregistrement peut être supprimé de façon qu'il ne puisse plus recevoir de demandes. Les enregistrements expirent aussi après un certain temps si ils ne sont pas rafraîchis.

Les enregistrements représentent un élément dynamique d'état entretenu par le réseau. Il y a de nombreux cas où les agents d'utilisateur aimeraient savoir les changements d'état des enregistrements. Le cadre d'événements SIP [RFC3265] définit un cadre général d'abonnement, et de notification des événements qui se rapportent aux systèmes SIP. Le cadre définit les méthodes SUBSCRIBE et NOTIFY, et introduit la notion de paquetage. Un paquetage est une application concrète du cadre d'événement à une classe particulière d'événements. Les paquetages ont été définis, par exemple, pour la présence [RFC3857] des utilisateurs. La présente spécification définit un paquetage pour l'état d'enregistrement.

2. Terminologie

Dans le présent document, les mots clés "DOIT", "NE DOIT PAS", "EXIGÉ", "DEVRA", "NE DEVRA PAS", "DEVRAIT", "NE DEVRAIT PAS", "RECOMMANDE", "PEUT", et "FACULTATIF" sont à interpréter comme décrit dans le BCP 14, [RFC2119] et indiquent les niveaux d'exigence pour les mises en œuvre conformes.

3. Scénarios d'utilisation

Il y a de nombreuses applications de ce paquetage d'événement. Quelques unes sont documentées ici pour illustrer le propos.

3.1 Ré-authentification forcée

On prévoit que beaucoup d'appareils SIP seront des appareils sans fil qui seront toujours connectés, et donc continuellement enregistrés sur le réseau. Malheureusement, l'histoire a montré que ces appareils peuvent être compromis. Pour faire face à cela, un administrateur va vouloir terminer ou abrèger un enregistrement, et demander à l'appareil de se ré-enregistrer afin qu'il puisse être ré-authentifié. Pour ce faire, l'appareil souscrit au paquetage d'événement enregistrement pour l'adresse d'enregistrement pour laquelle il enregistre les contacts. Lorsque l'administrateur abrège les enregistrements (par exemple, lorsque une fraude est suspectée) le serveur d'enregistrement envoie une notification à l'appareil. Il peut alors se ré-enregistrer et se ré-authentifier. Si il ne peut pas se ré-authentifier, l'expiration va terminer la session peu après.

3.2 Composition de présence

Un concept qu'il est important de comprendre est la relation entre ce paquetage d'événement et le paquetage d'événement pour la présence de l'utilisateur [RFC3857]. La présence d'utilisateur représente la volonté et la capacité d'un utilisateur de communiquer avec d'autres usagers sur le réseau. Elle est composée d'un ensemble d'adresses de contact qui représentent les divers moyens de contacter l'utilisateur. Ces adresses de contact peuvent représenter l'adresse de contact pour la voix, par exemple. Normalement, l'adresse de contact mentionnée pour la voix sera une adresse d'enregistrement. L'état de ce contact (si il est ouvert ou fermé) peut dépendre de nombreux facteurs, incluant l'état de tout enregistrement à cette adresse d'enregistrement. Il

en résulte que l'état d'enregistrement peut être vu comme une entrée du processus qui détermine l'état de présence d'un utilisateur. Effectivement, l'état d'enregistrement est une donnée "brute", qui est combinée à d'autres informations sur un utilisateur pour générer un document qui décrit la présence de l'utilisateur.

En fait, ce paquetage d'événement permet qu'un serveur de présence soit séparé d'un serveur d'enregistrement SIP, tout en utilisant les informations d'enregistrement pour construire un document de présence. Lorsque un serveur de présence reçoit un abonnement de présence pour un certain usager, le serveur de présence lui-même va générer une souscription au serveur d'enregistrement pour le paquetage d'événement d'enregistrement. Il en résulte que le serveur de présence va apprendre l'état d'enregistrement pour cet usager, et il pourrait utiliser cette information pour générer des documents de présence.

3.3 Notices de bienvenue

Un service courant dans les réseaux mobiles actuels est la "notice de bienvenue". Lorsque l'utilisateur allume son téléphone dans un pays étranger, il reçoit un message qui l'accueille dans le pays, et lui donne des informations, par exemple sur les services de transport.

Pour mettre en œuvre ce service dans un système SIP, un serveur d'application peut souscrire à l'état d'enregistrement de l'utilisateur. Lorsque l'utilisateur allume son téléphone, celui-ci va générer un enregistrement. Il va en résulter l'envoi d'une notification à l'application que l'utilisateur a enregistrée. L'application peut alors envoyer une demande MESSAGE SIP [RFC3428] à l'appareil, accueillant l'utilisateur et lui fournissant toutes les informations nécessaires.

4. Définition du paquetage

La présente section précise les détails nécessaires pour spécifier un paquetage d'événement comme défini au paragraphe 4.4 de la [RFC3265].

4.1 Nom du paquetage d'événement

La spécification d'événements SIP exige que les définitions de paquetage spécifient le nom de leur paquetage ou gabarit de paquetage.

Le nom de ce paquetage est "reg". Comme défini dans la [RFC3265], cette valeur apparaît dans l'en-tête Event présent dans les demandes SUBSCRIBE et NOTIFY.

Exemple : Event: reg

4.2 Paramètres du paquetage d'événement

La spécification d'événements SIP exige des définitions de paquetage et de gabarit de paquetage qu'elles spécifient tous les paramètres spécifiques du paquetage de l'en-tête Event qu'elles utilisent.

Aucun paramètre du paquetage spécifique de l'en-tête Event n'est défini pour ce paquetage d'événement.

4.3 Corps SUBSCRIBE

La spécification d'événements SIP exige des définitions de paquetage et de gabarit de paquetage qu'elles définissent l'usage, si il en est, des corps dans les demandes SUBSCRIBE.

Un SUBSCRIBE pour des événements d'enregistrement PEUT contenir un corps. Ce corps servira aux besoins de filtrage de la souscription. La définition d'un tel corps sort du domaine d'application de cette spécification.

Un SUBSCRIBE pour le paquetage d'enregistrement PEUT être envoyé sans corps. Cela implique que la politique de filtrage d'enregistrement par défaut ait été demandée. La politique par défaut est :

- o les notifications sont générées chaque fois qu'il y a un changement d'état d'un des contacts enregistrés pour la ressource objet de l'abonnement. Ces notifications ne contiennent que des informations sur les contacts dont l'état a changé ;
- o les notifications déclanchées à partir d'un SUBSCRIBE contiennent l'état complet (la liste de tous les contacts liés à l'adresse d'enregistrement).

Bien sûr, le serveur peut appliquer toutes les politiques qu'il veut à l'abonnement.

4.4 Durée de souscription

La spécification d'événements SIP exige des définitions de paquetage qu'elles définissent une valeur par défaut pour les durées de souscription, et de discuter les choix raisonnables de durée lorsque ils sont explicitement spécifiés.

L'état d'enregistrement change lorsque des contacts sont créés par des demandes REGISTER, et elles arrivent à expiration lorsque il n'y a plus de rafraîchissements. Leur taux de changement est donc en rapport avec l'expiration normale d'enregistrement. Comme l'expiration par défaut pour les enregistrements est de 3600 secondes, la durée par défaut des souscriptions à l'état d'enregistrement est légèrement plus long, 3761 secondes. Cela aide à éviter des problèmes potentiels de couplage de la souscription et des rafraîchissements d'enregistrement. Bien sûr, les clients PEUVENT inclure un en-tête Expires dans la demande SUBSCRIBE pour demander une durée différente.

4.5 Corps NOTIFY

La spécification d'événements SIP exige des définitions de paquetage qu'elles décrivent l'ensemble permis de types de corps dans les demandes NOTIFY, et qu'elles spécifient la valeur par défaut à utiliser lorsque il n'y a pas d'en-tête Accept dans la demande SUBSCRIBE.

Le corps d'une notification de changement d'état d'enregistrement contient un document d'informations d'enregistrement. Ce document décrit tout ou partie des contacts associés à une adresse d'enregistrement particulière. Tous les souscripteurs et notificateurs DOIVENT prendre en charge le format "application/reginfo+xml" décrit à la Section 5. La demande de souscription PEUT contenir un champ d'en-tête Accept. Si un tel champ d'en-tête n'est pas présent, il a une valeur par défaut de "application/reginfo+xml". Si le champ d'en-tête est présent, il DOIT inclure "application/reginfo+xml", et PEUT inclure tout autre type capable de représenter les informations d'enregistrement.

Bien sûr, les notifications générées par le serveur DOIVENT être dans un des formats spécifiés dans le champ d'en-tête Accept dans la demande SUBSCRIBE.

4.6 Traitement du notificateur des demandes SUBSCRIBE

Le cadre d'événements SIP spécifie que les paquetages devraient définir tout traitement spécifique du paquetage des demandes SUBSCRIBE à un notificateur, spécifiquement à l'égard de l'authentification et de l'autorisation.

L'état d'enregistrement peut être une information sensible. Donc, toutes les souscriptions DEVRAIENT être authentifiées et autorisées avant d'être approuvées. L'authentification PEUT être effectuée en utilisant toute technique disponible par SIP, incluant le résumé, S/MIME, TLS, ou autre mécanisme spécifique du transport [RFC3261]. La politique d'autorisation est à la discrétion de l'administrateur, comme toujours. Cependant, on peut faire quelques recommandations.

Il est RECOMMANDÉ qu'un utilisateur soit autorisé à souscrire à son propre état d'enregistrement. De tels abonnements sont utiles lorsque il y a de nombreux appareils qui représentent un utilisateur, dont chacun doit apprendre l'état d'enregistrement des autres appareils. On prévoit aussi que les applications et les automates vont souvent être les souscripteurs de l'état d'enregistrement. Dans ces cas, les politiques d'autorisation seront normalement fournies à l'avance.

4.7 Génération par le notificateur des demandes NOTIFY

Le cadre d'événement SIP demande que les paquetages spécifient les conditions dans lesquelles les notifications sont envoyées pour ce paquetage, et comment de telles notifications sont construites.

Pour déterminer quand un notificateur devrait envoyer les notifications de changements d'état d'enregistrement, on définit un automate à états finis (FSM, *finite state machine*) qui représente l'état d'un contact pour une adresse d'enregistrement particulière. Les transitions dans ce FSM PEUVENT résulter en la génération de notifications. Ces notifications vont porter les informations sur le nouvel état et l'événement qui a déclenché le changement d'état. Il est important de noter que ce FSM est juste un modèle de la machinerie d'état d'enregistrement entretenue par un serveur. Une mise en œuvre transposerait son propre automate à états en celui-ci d'une façon spécifique.

4.7.1 Automate à états d'enregistrement

L'automate à états sous-jacent pour un enregistrement est montré à la Figure 1. La machine est très simple. Une instance de

demande REGISTER, le FSM reste dans le même état, mais un événement rafraîchi est généré. De même, lorsque un administrateur modifie l'heure d'expiration d'un lien (sans supprimer le lien) pour déclencher le réenregistrement du contact et éventuellement sa ré-authentification, le FSM reste dans l'état actif, mais un événement abrégé est généré.

Lorsque un contact n'est plus lié à l'adresse d'enregistrement, le FSM passe à l'état terminé, et une fois qu'un NOTIFY est envoyé, l'automate à états est détruit. Par suite, l'état terminé est effectivement transitoire. Il y a plusieurs raisons pour cela. La première est une expiration, qui survient lorsque le contact n'a pas été rafraîchi par une demande REGISTER. La seconde raison est la désactivation. Cela se produit lorsque l'administrateur a retiré le contact comme lien valide, mais souhaite encore que le client tente de réenregistrer le contact. À l'opposé, l'événement rejeté survient lorsque un contact actif est retiré par l'administrateur, mais qu'un réenregistrement ne va pas aider à le rétablir. Cela peut arriver, par exemple si un utilisateur ne paie pas ses factures. L'événement probation survient lorsque un contact actif est retiré par l'administrateur, et que celui-ci veut que le client se réenregistre, mais qu'il le fasse ultérieurement. L'événement non enregistré survient lorsque une demande REGISTER règle à zéro l'heure d'expiration de ce contact.

4.7.2 Application de l'automate à états

Le serveur PEUT générer une notification aux souscripteurs lorsque un événement survient soit dans l'automate à états d'adresse d'enregistrement, soit dans celui par contact, sauf pour la transition de terminé à init dans l'automate d'adresse d'état d'enregistrement. Comme on l'a noté ci-dessus, une notification NE DOIT PAS être envoyée dans ce cas. Pour les autres transitions, que le serveur envoie une notification ou non dépend de la politique. Cependant, plusieurs lignes directrices sont définies.

En règle générale, quand un souscripteur est autorisé à recevoir des notifications sur un ensemble d'enregistrements, il est RECOMMANDÉ que les notifications contiennent les informations sur les contacts qui ont changé d'état (et ont donc déclenché une notification) plutôt que de livrer l'état actuel de chaque contact dans tous les enregistrements. Cependant, les notifications déclanchées par suite de l'opération de récupération (un SUBSCRIBE avec un Expires de 0) DEVRAIENT résulter en ce que l'état complet de tous les contacts pour tous les enregistrements soit présent dans le NOTIFY.

4.8 Traitement des demandes NOTIFY par le souscripteur

Le cadre d'événements SIP attend des paquetages qu'ils spécifient comment un souscripteur traite les demandes NOTIFY de toutes les façons spécifiques du paquetage, et en particulier, comment il utilise les demandes NOTIFY pour construire une vue cohérente de l'état de la ressource souscrite. Normalement, le NOTIFY va seulement contenir des informations sur les contacts dont l'état a changé. Pour construire une vue cohérente de l'état total de tous les enregistrements, le souscripteur va avoir besoin de combiner les NOTIFY reçus au fil du temps. Les détails de ce processus dépendent du format de document utilisé pour porter l'état d'enregistrement. La Section 5 précise le processus pour le format application/reginfo+xml.

4.9 Traitement des demandes fourchées

Le cadre d'événements SIP rend obligatoire que les paquetages indiquent si les demandes SUBSCRIBE fourchées peuvent ou non installer plusieurs souscriptions.

L'état d'enregistrement est normalement mémorisé dans un répertoire (qu'il soit colocalisé avec un mandataire/registraire ou dans une base de données séparée). À ce titre, il y a généralement un seul endroit où résident les informations de contact pour une adresse d'enregistrement particulière. Cela implique qu'une souscription à ces informations soit directement traitée par un seul élément avec accès à ce répertoire. Il n'y a donc pas de besoin pressant qu'une souscription aux informations d'enregistrement fourche. Par suite, un souscripteur NE DOIT PAS créer plusieurs dialogues pour une seule demande de souscription. Le traitement exigé pour garantir qu'un seul dialogue est établi est décrit au paragraphe 4.4.9 du cadre d'événements SIP [RFC3265].

4.10 Taux des notifications

Le cadre d'événements SIP rend obligatoire que les paquetages définissent un taux maximum de notifications.

Pour des raisons de contrôle d'encombrement, il est important que le taux de notifications ne devienne pas excessif. Par suite, il est RECOMMANDÉ que le serveur ne génère pas de notifications pour un seul souscripteur à un taux supérieur à une toutes les cinq secondes.

4.11 Agents d'état

Le cadre d'événements SIP demande aux paquetages de prendre en considération le rôle des agents d'état dans leur conception.

Les agents d'état n'ont pas de rôle dans le traitement de ce paquetage.

5. Informations d'enregistrement

5.1 Structure des informations d'enregistrement

Les informations d'enregistrement sont un document XML [XML1.0] qui DOIT être bien formé et DEVRAIT être valide. Les documents d'informations d'enregistrement DOIVENT être fondés sur XML 1.0 et DOIVENT être codés en utilisant UTF-8. La présente spécification utilise l'espace de noms XML pour identifier les documents d'informations d'enregistrement et les fragments de document. L'URI d'espace de noms pour les éléments définis dans la présente spécification est un URN [RFC2141], qui utilise l'identifiant d'espace de nom 'ietf' défini par la [RFC2648] et étendu par la [RFC3688]. Cet URN est :

urn:ietf:params:xml:ns:reginfo

Un document d'informations d'enregistrement commence par l'étiquette d'élément racine "reginfo". Il consiste en un nombre quelconque de sous éléments "enregistrement", dont chacun contient l'état d'enregistrement d'une adresse d'enregistrement particulière. Les informations d'enregistrement pour une certaine adresse d'enregistrement DOIVENT être contenues dans un seul élément "enregistrement" ; elles ne peuvent pas être dispersées sur plusieurs éléments "enregistrement" au sein d'un document. D'autres éléments provenant d'espaces de noms différents PEUVENT être présents à des fins d'extensibilité ; des éléments ou attributs provenant d'espaces de noms inconnus DOIVENT être ignorés. Il y a deux attributs associés aux éléments "reginfo", qui DOIVENT tous deux être présents :

version : cet attribut permet au receveur du document d'informations d'enregistrement de les ranger correctement. Les versions commencent à 0, et s'incrémentent de un pour chaque nouveau document envoyé à un abonné. Les versions ont une portée limitée à un abonnement. Les versions DOIVENT être représentables en utilisant un entier de 32 bits.

state : cet attribut indique si le document contient l'état d'enregistrement complet, ou si il contient seulement les informations sur les enregistrements qui ont changé depuis le document précédent (partiel).

Noter que le format de document permet explicitement de porter des informations sur plusieurs adresses d'enregistrement. Cela permet à des souscriptions de grouper des enregistrements, et un tel groupe est identifié par une sorte d'URI. Par exemple, un domaine peut définir sip:allusers@example.com comme ressource à laquelle on peut s'abonner et qui génère des notifications lorsque change l'état d'une adresse d'enregistrement dans le domaine.

L'élément "enregistrement" a une liste de tous les sous éléments de "contact", dont chacun contient les informations sur un seul contact. D'autres éléments provenant d'espaces de noms différents PEUVENT être présents pour des besoins d'extensibilité ; les éléments ou attributs provenant d'espaces de noms inconnus DOIVENT être ignorés. Trois attributs sont associés à l'élément "enregistrement" et tous DOIVENT être présents :

aor : l'attribut aor contient un URI qui est l'adresse d'enregistrement à laquelle se réfère cet enregistrement.

id : l'attribut id identifie cet enregistrement. Il DOIT être unique parmi tous les autres attributs id présents dans d'autres éléments enregistrement portés au souscripteur dans la portée de leur abonnement. En particulier, si deux URI qui identifient une adresse d'enregistrement différent après leur canonisation conformément aux procédures de l'étape 5 du paragraphe 10.3 de la [RFC3261], les attributs id dans les éléments "enregistrement" pour ces adresses d'enregistrement DOIVENT différer. De plus, l'attribut id pour un élément "enregistrement" pour une certaine adresse d'enregistrement DOIT être le même pour toutes les notifications envoyées dans l'abonnement.

state : l'attribut "state" indique l'état de l'enregistrement. Les valeurs valides sont "init", "actif" et "terminé".

L'élément "contact" contient un élément "uri", un élément facultatif "display-name" (*nom d'affichage*), et un élément facultatif "unknown-param" (*paramètre inconnu*). D'autres éléments provenant d'espaces de noms différents PEUVENT être présents pour des besoins d'extensibilité ; les éléments ou attributs provenant d'espaces de noms inconnus DOIVENT être ignorés. Il y a plusieurs attributs associés à l'élément "contact" qui DOIVENT être présents:

id : l'attribut id identifie ce contact. Il DOIT être unique parmi tous les autres attributs id présents dans les autres éléments contact portés au souscripteur dans la portée de son abonnement. En particulier, si l'URI pour deux contacts diffère (sur

la base des règles de comparaison d'URI de la [RFC3261]) les attributs id pour ces contacts DOIVENT différer. Cependant, à la différence de l'attribut id pour une adresse d'enregistrement, si l'URI pour deux contacts est le même, leurs attributs id DEVRAIENT être les mêmes dans les notifications. Cette exigence est un DEVRAIT et non un DOIT car il est difficile de calculer un tel id en fonction de l'URI sans conserver plus d'état. Aucune fonction de hachage appliquée à l'URI ne peut, en fait, satisfaire une exigence absolue. Cela, parce que l'égalité de l'URI SIP n'est pas transitive. Cependant, une fonction de hachage qui inclut des paramètres d'URI inconnus (c'est-à-dire non définis dans la RFC 3261) va toujours résulter en une valeur qui sera différente si deux URI sont différents, et qui sera usuellement la même si les URI sont égaux.

state : l'attribut state indique l'état du contact. Les valeurs valides sont "actif" et "terminé".

event : l'attribut event indique l'événement qui a causé le passage de l'automate à états de contact à l'état actuel. Les valeurs valides sont enregistré, créé, rafraîchi, abrégé, expiré, désactivé, probation, non enregistré et rejeté.

Si l'attribut event a une valeur de abrégé, l'attribut "expires" DOIT être présent. Il contient un long entier non signé qui indique le nombre de secondes restant jusqu'à l'arrivée à expiration du lien. Cet attribut PEUT être inclus avec toute valeur d'attribut event pour laquelle l'état du contact est actif.

Si l'attribut event a une valeur de probation, l'attribut "retry-after" DOIT être présent. Il contient un long entier non signé qui indique le nombre de secondes après quoi le propriétaire du contact est supposé réessayer son enregistrement.

L'attribut facultatif "duration-registered" porte la durée pendant laquelle le contact a été lié à l'adresse d'enregistrement, en secondes. L'attribut facultatif "q" porte la priorité relative de ce contact par rapport aux autres contacts enregistrés. L'attribut facultatif "callid" contient l'identifiant d'appel actuel porté dans le dernier REGISTER utilisé pour mettre à jour ce contact, et l'attribut facultatif "cseq" contient la dernière valeur de CSeq présente dans une demande REGISTER qui a mis à jour cette valeur de contact.

L'élément "uri" contient l'URI associé à ce contact. L'élément "display-name" contient le nom d'affichage pour le contact. L'élément "display-name" PEUT contenir l'attribut xml:lang pour indiquer le langage du nom d'affichage.

L'élément "unknown-param" est utilisé pour porter des paramètres de champ d'en-tête contact qui ne sont pas spécifiés dans la RFC 3261. Un exemple est celui des paramètres de capacités de l'agent d'utilisateur spécifiés dans la [RFC3840]. Chaque élément "unknown-param" décrit un seul paramètre de champ d'en-tête contact. Le nom du paramètre est contenu dans l'attribut de nom obligatoire de l'élément "unknown-param", et la valeur du paramètre est le contenu de l'élément "unknown-param". Pour les paramètres de champ d'en-tête contact qui n'ont pas de valeur, le contenu de l'élément "unknown-param" est vide.

5.2 Calcul des enregistrements à partir du document

Normalement, le NOTIFY pour des informations d'enregistrement ne va contenir que des informations sur les contacts dont l'état a changé. Pour construire une vue cohérente de l'état total de tous les enregistrements, un souscripteur va avoir besoin de combiner les NOTIFY reçus au fil du temps. Le souscripteur tient un tableau pour chaque enregistrement pour lequel il reçoit des informations. Chaque enregistrement est identifié de façon univoque par l'attribut "id" dans l'élément "enregistrement". Chaque tableau contient une rangée pour chaque contact de l'enregistrement. Chaque rangée est indexée par l'identifiant univoque pour ce contact. Il est porté dans l'attribut "id" de l'élément "contact". Le contenu de chaque rangée contient l'état de ce contact tel que porté par l'élément "contact". Les tableaux sont aussi associés à un numéro de version. Le numéro de version DOIT être initialisé avec la valeur de l'attribut "version" provenant de l'élément "reginfo" dans le premier document reçu. Chaque fois qu'un nouveau document est reçu, la valeur du numéro de version local, et l'attribut "version" dans le nouveau document, sont comparés. Si la valeur dans le nouveau document est supérieure de un à celle du numéro de version local, le numéro de version local est augmenté de un, et le document est traité. Si la valeur dans le document est supérieure de plus de un à celle du numéro de version local, celui-ci est réglé à la valeur dans le nouveau document, le document est traité, et le souscripteur DEVRAIT générer une demande de rafraîchissement pour déclencher une notification d'état plein. Si la valeur dans le document est inférieure à celle de la version locale, le document est éliminé sans traitement.

Le traitement du document dépend de si il contient un état plein ou partiel. Si il contient l'état plein, indiqué par la valeur de l'attribut "state" dans l'élément "reginfo", le contenu de tous les tableaux associés à cette souscription est purgé. Il est repeuplé à partir du document. Un nouveau tableau est créé pour chaque élément "enregistrement", et une nouvelle rangée est créée dans chaque tableau pour chaque élément "contact". Si le reginfo contient l'état partiel, comme indiqué par la valeur de l'attribut "state" dans l'élément "reginfo", le document est utilisé pour mettre à jour les tableaux existants. Pour chaque élément "enregistrement", le souscripteur vérifie qu'un tableau existe pour cet enregistrement. Cette vérification est faite en comparant la valeur de l'attribut "id" de l'élément "enregistrement" à l'identifiant associé au tableau. Si il n'existe pas de tableau pour cet

enregistrement, il en est créé un. Pour chaque élément "contact" dans l'enregistrement, le souscripteur vérifie si une rangée existe pour ce contact. Cette vérification est faite en comparant l'identifiant dans l'attribut "id" de l'élément "contact" à l'identifiant associé à la rangée. Si le contact n'existe pas dans le tableau, une rangée est ajoutée, et son état est réglé sur les informations provenant de cet élément "contact". Si le contact existe, son état est mis à jour sur celui des informations provenant de cet élément "contact". Si une rangée est mise à jour ou créée, de telle sorte que son état soit maintenant terminé, cette entrée PEUT être retirée du tableau à tout moment.

5.3 Exemple

Voici un exemple de document d'informations d'enregistrement :

```
<?xml version="1.0"?>
<reginfo xmlns="urn:ietf:params:xml:ns:reginfo"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  version="0" state="full">
  <registration aor="sip:user@example.com" id="as9"
    state="active">
    <contact id="76" state="active" event="registered"
      duration-registered="7322"
      q="0.8">
      <uri>sip:user@pc887.example.com</uri>
    </contact>
    <contact id="77" state="terminated" event="expired"
      duration-registered="3600"
      q="0.5">
      <uri>sip:user@university.edu</uri>
    </contact>
  </registration>
</reginfo>
```

5.4 Schéma XML

Voici un schéma de définition du format reginfo :

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="urn:ietf:params:xml:ns:reginfo"
  xmlns:tns="urn:ietf:params:xml:ns:reginfo"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified" attributeFormDefault="unqualified">
  <!-- Cette importation apporte l'attribut de langage XML xml:lang-->
  <xs:import namespace="http://www.w3.org/XML/1998/namespace"
    schemaLocation="http://www.w3.org/2001/03/xml.xsd"/>
  <xs:element name="reginfo">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="tns:registration" minOccurs="0"
          maxOccurs="unbounded"/>
        <xs:any namespace="##other" processContents="lax" minOccurs="0"
          maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute name="version" type="xs:nonNegativeInteger"
        use="required"/>
      <xs:attribute name="state" use="required">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="full"/>
            <xs:enumeration value="partial"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
```

```

</xs:complexType>
</xs:element>
<xs:element name="registration">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="tns:contact" minOccurs="0" maxOccurs="unbounded"/>
      <xs:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="aor" type="xs:anyURI" use="required"/>
    <xs:attribute name="id" type="xs:string" use="required"/>
    <xs:attribute name="state" use="required">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="init"/>
          <xs:enumeration value="active"/>
          <xs:enumeration value="terminated"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
  </xs:complexType>
</xs:element>
<xs:element name="contact">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="uri" type="xs:anyURI"/>
      <xs:element name="display-name" minOccurs="0">
        <xs:complexType>
          <xs:simpleContent>
            <xs:extension base="xs:string">
              <xs:attribute ref="xml:lang" use="optional"/>
            </xs:extension>
          </xs:simpleContent>
        </xs:complexType>
      </xs:element>
      <xs:element name="unknown-param" minOccurs="0"
maxOccurs="unbounded">
        <xs:complexType>
          <xs:simpleContent>
            <xs:extension base="xs:string">
              <xs:attribute name="name" type="xs:string" use="required"/>
            </xs:extension>
          </xs:simpleContent>
        </xs:complexType>
      </xs:element>
      <xs:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="state" use="required">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="active"/>
          <xs:enumeration value="terminated"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="event" use="required">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="registered"/>
          <xs:enumeration value="created"/>
          <xs:enumeration value="refreshed"/>
          <xs:enumeration value="shortened"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
  </xs:complexType>
</xs:element>

```

```

<xs:enumeration value="expired"/>
<xs:enumeration value="deactivated"/>
<xs:enumeration value="probation"/>
<xs:enumeration value="unregistered"/>
<xs:enumeration value="rejected"/>
</xs:restriction>
</xs:simpleType>
</xs:attribute>
<xs:attribute name="duration-registered" type="xs:unsignedLong"/>
<xs:attribute name="expires" type="xs:unsignedLong"/>
<xs:attribute name="retry-after" type="xs:unsignedLong"/>
<xs:attribute name="id" type="xs:string" use="required"/>
<xs:attribute name="q" type="xs:string"/>
<xs:attribute name="callid" type="xs:string"/>
<xs:attribute name="cseq" type="xs:unsignedLong"/>
</xs:complexType>
</xs:element>
</xs:schema>

```

6. Exemple de flux d'appel

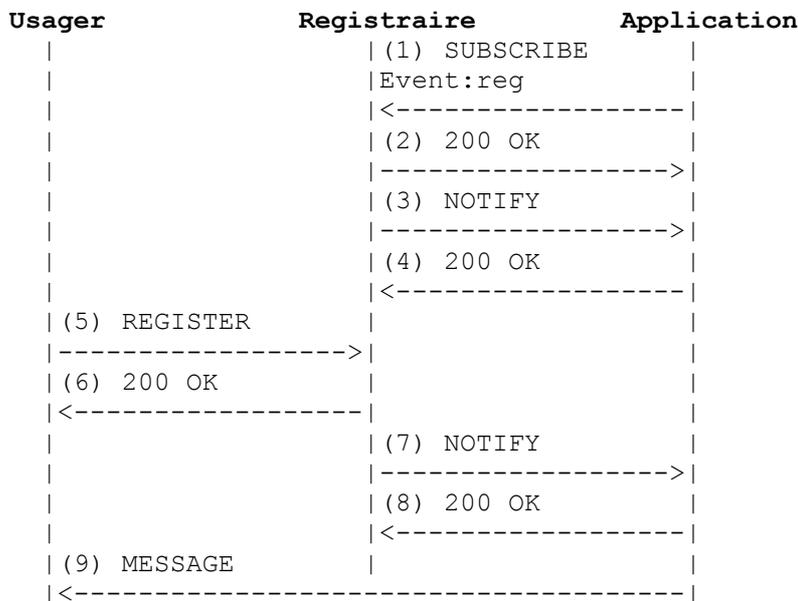


Figure 3 : Exemple de flux d'appel

Ce paragraphe donne un exemple de flux d'appel, montré à la Figure 3. Elle montre une mise en œuvre de l'application de la notice de bienvenue décrite au paragraphe 3.3. D'abord, l'application envoie un SUBSCRIBE au paquetage d'événement enregistrement pour l'utilisateur désiré (1) :

```

SUBSCRIBE sip:joe@example.com SIP/2.0
Via: SIP/2.0/UDP app.example.com;branch=z9hG4bKnashds7
From: sip:app.example.com;tag=123aa9
To: sip:joe@example.com
Call-ID: 9987@app.example.com
CSeq: 9887 SUBSCRIBE
Contact: sip:app.example.com
Event: reg
Max-Forwards: 70
Accept: application/reginfo+xml

```

Le registraire (agissant comme notificateur pour le paquetage d'événement enregistrement) génère un 200 OK au SUBSCRIBE :

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP app.example.com;branch=z9hG4bKnashds7
;received=192.0.2.1
From: sip:app.example.com;tag=123aa9
To: sip:joe@example.com;tag=xyzygg
Call-ID: 9987@app.example.com
CSeq: 9987 SUBSCRIBE
Contact: sip:server19.example.com
Expires: 3600
```

Le registraire génère ensuite une notification (3) avec l'état en cours. Comme il n'y a pas d'enregistrement actif, l'état de l'enregistrement est "init" :

```
NOTIFY sip:app.example.com SIP/2.0
Via: SIP/2.0/UDP server19.example.com;branch=z9hG4bKnasaii
From: sip:joe@example.com;tag=xyzygg
To: sip:app.example.com;tag=123aa9
Call-ID: 9987@app.example.com
CSeq: 1288 NOTIFY
Contact: sip:server19.example.com
Event: reg
Subscription-State:active;expires=3600
Max-Forwards: 70
Content-Type: application/reginfo+xml
Content-Length: ...
```

```
<?xml version="1.0"?>
<reginfo xmlns="urn:ietf:params:xml:ns:reginfo"
  version="0" state="full">
  <registration aor="sip:joe@example.com" id="a7" state="init" />
</reginfo>
```

Plus tard, l'utilisateur s'enregistre (5) :

```
REGISTER sip:example.com SIP/2.0
Via: SIP/2.0/UDP pc34.example.com;branch=z9hG4bKnaaff
From: sip:joe@example.com;tag=99a8s
To: sip:joe@example.com
Call-ID: 88askjda9@pc34.example.com
CSeq: 9976 REGISTER
Contact: sip:joe@pc34.example.com
```

Il en résulte la génération d'un NOTIFY à l'application (7) :

```
NOTIFY sip:app.example.com SIP/2.0
Via: SIP/2.0/UDP server19.example.com;branch=z9hG4bKnasaij
From: sip:joe@example.com;tag=xyzygg
To: sip:app.example.com;tag=123aa9
Call-ID: 9987@app.example.com
CSeq: 1289 NOTIFY
Contact: sip:server19.example.com
Event: reg
Subscription-State:active;expires=3000
Max-Forwards: 70
Content-Type: application/reginfo+xml
Content-Length: ...
```

```
<?xml version="1.0"?>
<reginfo xmlns="urn:ietf:params:xml:ns:reginfo"
  version="1" state="partial">
  <registration aor="sip:joe@example.com" id="a7" state="active">
    <contact id="76" state="active" event="registered"
```

```

    duration-registered="0">
    <uri>sip:joe@pc34.example.com</uri>
  </contact>
</registration>
</reginfo>

```

L'application peut alors envoyer son message instantané à l'appareil (9) :

```

MESSAGE sip:joe@pc34.example.com SIP/2.0
Via: SIP/2.0/UDP app.example.com;branch=z9hG4bKnashds8
From: sip:app.example.com;tag=123aa10
To: sip:joe@example.com
Call-ID: 9988@app.example.com
CSeq: 82779 MESSAGE
Max-Forwards: 70
Content-Type: text/plain
Content-Length: ...

```

Bienvenue au service example.com !

7. Considérations pour la sécurité

Les considérations sur la sécurité pour le paquetage d'événements SIP sont discutées dans la [RFC3265], et ces considérations s'appliquent ici.

Les informations d'enregistrement sont des informations sensibles, potentiellement confidentielles. L'abonnement à ce paquetage d'événement DEVRAIT être authentifié et autorisé conformément à la politique locale. Des lignes directrices pour la politique sont suggérées au paragraphe 4.6. De plus, les notifications DEVRAIENT être envoyées d'une façon telle qu'elle assure la confidentialité, l'intégrité du message et la vérification de l'identité du souscripteur, comme l'envoi des souscriptions et notifications en utilisant un URL SIPS ou en protégeant les corps de notification avec S/MIME.

8. Considérations relatives à l'IANA

Le présent document enregistre un nouveau paquetage d'événement SIP, un nouveau type MIME (application/reginfo+xml), et un nouvel espace de noms XML.

8.1 Enregistrement du paquetage d'événements SIP

Nom du paquetage : reg
 Type : paquetage
 Contact : Jonathan Rosenberg, <jdrosen@jdrosen.net>
 Spécification publiée : RFC 3680.

8.2 Enregistrement MIME de application/reginfo+xml

Nom de type de support MIME : application
 Nom de sous-type MIME : reginfo+xml
 Paramètres obligatoires : aucun
 Paramètres facultatifs : les mêmes que le paramètre de jeu de caractères application/xml spécifiés dans la [RFC3023].
 Considérations de codage : les mêmes que les considérations de codage de application/xml spécifiées dans la [RFC3023].
 Considérations de sécurité : voir la Section 10 de la [RFC3023] et la Section 7 de la présente spécification.
 Considérations d'interopérabilité : aucune.
 Spécification publiée : le présent document.
 Applications qui utilisent ce type de support : ce type de document est utilisé dans les notifications pour alerter les agents d'utilisateur SIP que leur enregistrement a expiré et doit être refait.
 Informations supplémentaires :
 Numéro magique : aucun.

Extension de fichier : .rif or .xml

Code de type de fichier Macintosh : "TEXT"

Adresse personnelle et de messagerie pour d'autres informations : Jonathan Rosenberg, <jdrosen@jdrosen.net>

Utilisation prévue : courante.

Auteur/contrôleur des modifications : IETF.

8.3 Enregistrement du sous espace de noms d'URN pour urn:ietf:params:xml:ns:reginfo

Ce paragraphe enregistre un nouvel espace de noms XML, selon les lignes directrices de la [RFC3688].

URI : l'URI pour cet espace de noms est urn:ietf:params:xml:ns:reginfo.

Contact du déposant : IETF, groupe de travail SIMPLE, <simple@ietf.org>, Jonathan Rosenberg <jdrosen@jdrosen.net>.

XML :

```
DÉBUT
<?xml version="1.0"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML Basic 1.0//EN"
    "http://www.w3.org/TR/xhtml-basic/xhtml-basic10.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta http-equiv="content-type"
    content="text/html; charset=iso-8859-1"/>
  <title>Registration Information Namespace</title>
</head>
<body>
  <h1>Namespace for Registration Information</h1>
  <h2>urn:ietf:params:xml:ns:reginfo</h2>
  <p>See <a href="ftp://ftp.rfc-editor.org/in-notes/rfc3680.txt">
    RFC3680</a>.</p>
</body>
</html>
FIN
```

9. Références

9.1 Références normatives

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

[RFC2141] R. Moats, "[Syntaxe des URN](#)", mai 1997.

[RFC2648] R. Moats, "Espace de nom d'URN pour les documents de l'IETF", août 1999. (*Information*)

[RFC3023] M. Murata, S. St.Laurent et D. Kohn, "Types de support XML", janvier 2001. (*Obsolète, voir RFC7303*)

[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*)

[RFC3688] M. Mealling, "[Registre XML de l'IETF](#)", BCP 81, janvier 2004.

[XML1.0] W3C, "Extensible markup language (xml) 1.0." On peut trouver la spécification XML 1.0 à <http://www.w3.org/TR/1998/REC-xml-19980210>

9.2 Références pour information

- [RFC3428] B. Campbell et autres, "[Extension de messagerie instantanée](#) pour le protocole d'initialisation de session (SIP)", décembre 2002.
- [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
- [RFC3857] J. Rosenberg, "[Paquetage-gabarit d'événement d'information](#) d'observateur pour le protocole d'initialisation de session (SIP)", août 2004. (P.S.)
- [12] Mayer, G. and M. Beckmann, "Registration Event package", Non publié.

10. Contributeurs

Ce document se fonde largement sur le paquetage d'événement Enregistrement proposé à l'origine par Beckmann et Mayer dans [12]. On peut les contacter à :

Georg Mayer
Siemens AG
Hoffmannstr. 51
Munich 81359
Germany

mél : Georg.Mayer@icn.siemens.de

Mark Beckmann
Siemens AG
P.O. Box 100702
Salzgitter 38207
Germany

mél : Mark.Beckmann@siemens.com

Rohan Mahy a fourni un travail rédactionnel pour faire progresser cette spécification. Son adresse est :

Rohan Mahy
Cisco Systems
170 West Tasman Dr, MS: SJC-21/3/3
Phone: +1 408 526 8570
mél : rohan@cisco.com

11. Remerciements

On tient à remercier Dean Willis de son soutien.

12. Adresse de l'auteur

Jonathan Rosenberg
dynamicsoft
600 Lanidex Plaza
Parsippany, NJ 07054
USA

mél : jdrosen@dynamicsoft.com

13. Déclaration complète de droits de reproduction

Copyright (C) The Internet Society (2004)

Le présent document est soumis aux droits, licences et restrictions contenus dans le BCP 78, et sauf pour ce qui est mentionné ci-après, les auteurs conservent tous leurs droits.

Le présent document et les informations qui y sont contenues sont fournies sur une base "EN L'ÉTAT" et le contributeur, l'organisation qu'il ou elle représente ou qui le/la finance (s'il en est), la INTERNET SOCIETY, l'IETF TRUST et la

INTERNET ENGINEERING TASK FORCE déclinent toutes garanties, exprimées ou implicites, y compris mais non limitées à toute garantie que l'utilisation des informations ci encloses ne violent aucun droit ou aucune garantie implicite de commercialisation ou d'aptitude à un objet particulier.

Propriété intellectuelle

L'IETF ne prend pas position sur la validité et la portée de tout droit de propriété intellectuelle ou autres droits qui pourraient être revendiqués au titre de la mise en œuvre ou l'utilisation de la technologie décrite dans le présent document ou sur la mesure dans laquelle toute licence sur de tels droits pourrait être ou n'être pas disponible ; pas plus qu'elle ne prétend avoir accompli aucun effort pour identifier de tels droits. Les informations sur les procédures de l'ISOC au sujet des droits dans les documents de l'ISOC figurent dans les BCP 78 et BCP 79.

Des copies des dépôts d'IPR faites au secrétariat de l'IETF et toutes assurances de disponibilité de licences, ou le résultat de tentatives faites pour obtenir une licence ou permission générale d'utilisation de tels droits de propriété par ceux qui mettent en œuvre ou utilisent la présente spécification peuvent être obtenues sur répertoire en ligne des IPR de l'IETF à <http://www.ietf.org/ipr>.

L'IETF invite toute partie intéressée à porter son attention sur tous copyrights, licences ou applications de licence, ou autres droits de propriété qui pourraient couvrir les technologies qui peuvent être nécessaires pour mettre en œuvre la présente norme. Prière d'adresser les informations à l'IETF à ietf-ipr@ietf.org.

Remerciement

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