

Groupe de travail Réseau
Request for Comments : 3265
 RFC mise à jour : 2543
 Catégorie : Sur la voie de la normalisation
 Traduction Claude Brière de L'Isle

A. B. Roach
 dynamicsoft
 juin 2002
 février 2008

Notification d'événement spécifique du protocole d'initialisation de session (SIP)

Statut de ce mémoire

Le présent document spécifie un protocole Internet sur la voie de la 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.

Notice de copyright

Copyright (C) The Internet Society (2002). Tous droits réservés.

Résumé

Le présent document décrit une extension au protocole d'initialisation de session (SIP, *Session Initiation Protocol*). L'objet de cette extension est de fournir un cadre extensible selon lequel les nœuds SIP puissent demander une notification aux nœuds distants de l'occurrence de certains événements. Des utilisations concrètes du mécanisme décrit dans le présent document pourront être normalisées à l'avenir. Noter que le mécanisme de notification d'événement défini ici N'EST PAS destiné à constituer une infrastructure d'utilisation générale pour toutes les classes de souscription et notification d'événement.

Table des matières

1 Introduction.....	2
1.1 Généralités sur le fonctionnement.....	2
1.2 Conventions de documentation.....	3
2 Définitions.....	3
3 Comportement du nœud.....	3
3.2 Description du comportement NOTIFY.....	8
3.3 Généralités.....	11
4 Paquetages d'événements.....	13
4.1 Adéquation de l'utilisation.....	13
4.2 Gabarit de paquetage d'événements.....	13
4.3 Quantité d'états à convoier.....	14
4.4 Responsabilités des paquetages d'événements.....	14
5 Considérations pour la sécurité.....	16
5.1 Contrôle d'accès.....	16
5.2 Mécanisme de confidentialité du notificateur.....	17
5.3 Attaques de déni de service.....	17
5.4 Attaques en répétition.....	17
5.5 Attaques par intrusion.....	17
5.6 Confidentialité.....	17
6 Considérations relatives à l'IANA.....	18
6.1 Informations d'enregistrement.....	18
6.2 Gabarit d'enregistrement.....	19
6.3 Noms de champ d'en-tête.....	19
6.4 Codes de réponse.....	19
7 Syntaxe.....	19
7.1 Nouvelles méthodes.....	19
7.2 Nouveaux en-têtes.....	21
7.3 Nouveaux codes de réponse.....	21
7.4 Définitions de BNF augmenté.....	22
8 Références normatives.....	22

9 Références pour information.....	23
10 Remerciements.....	23
11 Notice concernant les droits de propriété intellectuelle.....	23
12 Adresse de l'auteur.....	23
13 Déclaration complète de droits de reproduction.....	23

1 Introduction

La capacité à demander une notification asynchrone des événements se révèle utile dans de nombreux types de services SIP pour lesquels est nécessaire une coopération entre les nœuds d'extrémité. Des exemples de tels services sont le rappel automatique (fondé sur des événements d'état du terminal) les listes de contacts (fondées sur les événements de présence des usagers) les indications de message en attente (fondées sur les événements de changement d'état des boîtes à lettres) et les états d'inter réseautage RTPC/Internet (PINT, *PSTN and Internet Internetworking*) [2] (fondés sur les événements d'état d'appel).

Les méthodes décrites dans le présent document donnent le cadre dans lequel la notification de ces événements peut être ordonnée.

Les mécanismes de notification d'événement définis ici NE SONT PAS destinés à devenir une infrastructure d'utilisation générale pour toutes les classes d'abonnement et de notification d'événement. La satisfaction des exigences pour le problème général de l'ensemble des abonnements et notifications est bien trop complexe pour un seul protocole. Notre objectif est de fournir un cadre de travail spécifique de SIP pour la notification d'événement qui n'est pas si complexe qu'il ne puisse être utilisable pour des dispositifs simples, mais est suffisamment flexible pour rendre de grands services. Noter cependant que les paquetages d'événements fondés sur le présent cadre de travail peuvent définir des règles d'élaboration arbitraires qui gouvernent l'abonnement et la notification pour les événements ou classes d'événements qu'ils décrivent.

Le présent document ne décrit pas une extension qui pourrait être utilisée directement ; il doit être étendu par d'autres documents (qu'on appellera ici "paquetages d'événements"). Dans la terminologie de conception orientée objet, on pourrait voir cela en termes de classe de base abstraite qui doit être déclinée en une classe instanciable par des extensions ultérieures. Des lignes directrices pour la création de telles extensions sont exposées à la section 4.

1.1 Généralités sur le fonctionnement

Le concept général est que les entités du réseau peuvent s'abonner à une ressource ou un état d'appel pour divers ressources ou appels dans le réseau, et ces entités (ou entités agissant en leur nom) peuvent envoyer des notifications quand ces états changent.

Un flux de messages normal serait :

Abonné	Notificateur
----SUBSCRIBE----->	Demande d'état d'abonnement
<-----200-----	Accusé de réception d'abonnement
<----NOTIFY-----	Retour des informations sur l'état en cours
-----200----->	
<----NOTIFY-----	Retour des informations sur l'état en cours
-----200----->	

Les abonnements sont arrivés à expiration et doivent être rafraîchis par de nouveaux messages SUBSCRIBE.

1.2 Conventions de documentation

Plusieurs paragraphes tout au long du présent document apportent des éclaircissements sur les motivations. Ces passages ne sont pas normatifs, et sont fournis pour faciliter la compréhension par le lecteur. Ces passages sont en retrait du reste du texte et se présentent de la façon suivante :

Ceci est un exemple de texte explicatif non normatif. Il ne fait pas partie de la spécification, et n'est utilisé que dans un but de clarification.

Les numéros entre crochets (par exemple, [1]) notent une référence à une des entrées dans les sections de référence ; voir les sections 8 et 9.

Les termes en majuscules "DOIT", "DEVRAIT", "PEUT", "NE DEVRAIT PAS", "NE DOIT PAS", et "RECOMMANDE" sont utilisés selon la définition de la RFC 2119 [5].

L'utilisation de guillemets après des points et des virgules suit la convention utilisée par la Société américaine de mathématiques ; bien que contraire aux conventions traditionnelles de l'anglo-américain, cet usage apporte de la clarté à certains passages.

2 Définitions

Paquetage d'événement (*Event Package*) : un paquetage d'événement est une spécification supplémentaire qui définit un ensemble d'informations d'état que rapporte un notificateur à un abonné. Les paquetages d'événements définissent aussi plus en détail la syntaxe et la sémantique fondées sur le cadre de travail défini par le présent document qui vont convoyer de telles informations d'état.

Gabarit de paquetage d'événement (*Event Template-Package*) : c'est un type particulier de paquetage d'événements qui définit un ensemble d'états qui peuvent être appliqués à tous les paquetages d'événements possibles, y compris lui-même.

Notification : c'est l'acte d'un notificateur qui envoie un message NOTIFY à un abonné pour informer l'abonné de l'état d'une ressource.

notificateur : c'est un agent d'utilisateur qui génère des demandes NOTIFY afin de notifier aux abonnés l'état d'une ressource. Les notificateurs acceptent normalement aussi des demandes SUBSCRIBE pour créer des abonnements.

Agent d'état : un agent d'état est un notificateur qui publie des informations d'état au nom d'une ressource ; afin de le faire, il peut avoir besoin de rassembler de telles informations d'état à parti de plusieurs sources. Les agents d'état ont toujours des informations d'état complètes pour la ressource pour laquelle ils créent des notifications.

Abonné : c'est un agent d'utilisateur qui reçoit des demandes NOTIFY venant des notificateurs ; ces demandes NOTIFY contiennent des informations sur l'état d'une ressource à laquelle l'abonné s'intéresse. Les abonnés génèrent normalement aussi des demandes SUBSCRIBE et les envoient aux notificateurs pour créer des abonnements.

Abonnement : c'est un ensemble d'états d'application associé à un dialogue. Cet état d'application inclut un pointeur sur le dialogue associé, le nom du paquetage d'événements, et éventuellement un jeton d'identification. Les paquetages d'événements définiront des informations d'état d'abonnement supplémentaires. Par définition, les abonnements existent à la fois chez un abonné et chez un notificateur.

Migration d'abonnement : c'est l'acte de déplacement d'un abonnement d'un notificateur à un autre notificateur.

3 Comportement du nœud

3.1 Description du comportement SUBSCRIBE

La méthode SUBSCRIBE est utilisée pour demander l'état en cours et les mises à jour d'état à partir d'un nœud distant.

3.1.1 Durée d'abonnement

Les demandes SUBSCRIBE DEVRAIENT contenir un en-tête "Expires" (défini dans SIP [1]). Cette valeur de Expires indique la durée de l'abonnement. Afin de prolonger effectivement les abonnements au delà de la durée communiquée dans l'en-tête "Expires", les abonnés ont besoin de rafraîchir les abonnements de façon périodique en utilisant un nouveau message SUBSCRIBE sur le même dialogue que défini dans SIP [1].

Si aucun en-tête "Expires" n'est présent dans une demande SUBSCRIBE, la valeur par défaut que cela implique est définie par le paquetage d'événements utilisé.

Les réponses de classe 200 aux demandes SUBSCRIBE DOIVENT aussi contenir un en-tête "Expires". La période de

temps dans la réponse PEUT être plus courte mais NE DOIT PAS être plus longue que celle spécifiée dans la demande. La durée de la période dans la réponse est celle qui définit la durée de l'abonnement.

Un paramètre "Expires" dans l'en-tête "Contact" n'a pas de sémantique pour SUBSCRIBE et est explicitement non équivalent à un en-tête "Expires" dans une demande ou réponse SUBSCRIBE.

Une conséquence naturelle de ce schéma est qu'un SUBSCRIBE avec un "Expires" de 0 constitue une demande de désabonnement d'un événement.

En plus d'être une demande de désabonnement, un message SUBSCRIBE avec un "Expires" de 0 cause aussi une recherche d'état ; voir le paragraphe 3.3.6.

Les notificateurs peuvent aussi souhaiter annuler des abonnements aux événements ; c'est utile, par exemple, quand la ressource à laquelle un abonnement se réfère n'est plus disponible. Des précisions sur ce mécanisme sont exposées au paragraphe 3.2.2.

3.1.2 Identification des événements de l'abonnement et des classes d'événement

L'identification des événements est fournie par trois éléments d'information : URI de demande, type d'événement, et (facultativement) corps de message.

L'URI de demande d'une demande SUBSCRIBE, contient principalement assez d'informations pour acheminer la demande à l'entité appropriée selon les procédures d'acheminement de demande exposées dans SIP [1]. Il contient aussi assez d'informations pour identifier la ressource pour laquelle la notification d'événement est désirée, mais pas nécessairement assez d'informations pour identifier de façon univoque la nature de l'événement (par exemple, "sip:adam@dynamicsoft.com" serait un URI approprié pour s'abonner à l'état de ma présence ; il pourrait aussi être un URI approprié pour s'abonner à l'état de ma boîte de messagerie vocale).

Les abonnés DOIVENT inclure exactement un en-tête "Event" dans les demandes SUBSCRIBE, indiquant à quel événement ou classe d'événements ils s'abonnent. L'en-tête "Event" va contenir un jeton qui indique le type d'état pour lequel un abonnement est demandé. Ce jeton sera enregistré auprès de l'IANA et va correspondre à un paquetage d'événements qui décrit plus en détails la sémantique de l'événement ou classe d'événement. L'en-tête "Event" PEUT aussi contenir un paramètre "id". Ce paramètre "id", s'il est présent, contient un jeton opaque qui identifie l'abonnement spécifique au sein d'un dialogue. Un paramètre "id" n'est valide qu'au sein du domaine d'application d'un seul dialogue.

Si le paquetage d'événements auquel le jeton d'événement correspond définit le comportement associé au corps de ses demandes SUBSCRIBE, leur sémantique s'applique.

Les paquetages d'événements peuvent aussi définir des paramètres pour l'en-tête Event ; s'ils le font, ils doivent définir la sémantique pour de tels paramètres.

3.1.3 Valeurs d'en-tête SUBSCRIBE supplémentaires

Puisque les demandes SUBSCRIBE créent un dialogue comme défini dans SIP [1], elles PEUVENT contenir un en-tête "Accept". Cet en-tête, s'il est présent, indique les formats de corps permis dans les demandes NOTIFY suivantes. Les paquetages d'événements DOIVENT définir le comportement pour les demandes SUBSCRIBE sans en-tête "Accept" ; habituellement, cela va indiquer un seul type de corps, par défaut.

Les valeurs d'en-tête décrites dans le présent document sont à interpréter comme décrit dans SIP [1].

3.1.4 Comportement de l'abonné SUBSCRIBE

3.1.4.1 Demande d'abonnement

SUBSCRIBE est une méthode créatrice de dialogue, comme décrit dans SIP [1].

Lorsqu'un abonné souhaite s'abonner à un état particulier pour une ressource, il forme un message SUBSCRIBE. Si le SUBSCRIBE initial représente une demande en-dehors d'un dialogue (comme il l'est normalement) sa construction suit les procédures exposées dans SIP [1] pour la génération de demande UAC en-dehors d'un dialogue.

Cette demande SUBSCRIBE sera confirmée par une réponse finale. Les réponses de classe 200 indiquent que l'abonnement a été accepté, et qu'un NOTIFY va immédiatement être envoyé. Une réponse 200 indique que l'abonnement a été accepté et que l'utilisateur est autorisé à s'abonner à la ressource demandée. Une réponse 202 indique simplement que la demande d'abonnement a été comprise, et que l'autorisation peut être accordée ou non.

L'en-tête "Expires" dans une réponse de classe 200 à SUBSCRIBE indique la durée réelle pour laquelle l'abonnement va rester actif (sauf rafraîchissement).

Les réponses finales de classe non 200 indiquent qu'aucun abonnement ou dialogue n'a été créé, et qu'aucun message NOTIFY ultérieur ne sera envoyé. Toutes les réponses de classe non 200 (à l'exception de "489", décrit plus loin) ont la même signification et le même traitement comme décrit dans SIP [1].

Une demande SUBSCRIBE PEUT inclure un paramètre "id" dans son en-tête "Event" pour permettre la différenciation de plusieurs abonnements dans le même dialogue.

3.1.4.2 **Rafraîchissement des abonnements**

À tout moment avant l'expiration d'un abonnement, l'abonné peut rafraîchir le temporisateur de cet abonnement en envoyant une autre demande SUBSCRIBE sur le même dialogue que l'abonnement existant, et avec le même paramètre "id" d'en-tête "Event" (si il en était un présent dans l'abonnement initial). Le traitement d'une telle demande est le même que pour la création initiale d'un abonnement excepté ce qui est décrit ci-dessous.

Si le message SUBSCRIBE initial contenait un paramètre "id" dans l'en-tête "Event", le rafraîchissement de l'abonnement doit aussi contenir un paramètre "id" identique ; il serait autrement considéré comme nouvel abonnement dans un dialogue existant.

Si une demande SUBSCRIBE de rafraîchissement d'un abonnement reçoit une réponse "481", cela indique que l'abonnement s'est terminé et que l'abonné n'a pas reçu la notification de ce fait. Dans ce cas, l'abonné devrait considérer que l'abonnement est invalide. Si l'abonné souhaite se réabonner à l'état, il le fait en composant une demande initiale SUBSCRIBE sans rapport avec la précédente avec un nouvel identifiant d'appel fraîchement généré et une nouvelle étiquette "From" unique (voir au paragraphe 3.1.4.1.)

Si une demande SUBSCRIBE pour rafraîchir un abonnement échoue avec une réponse différente de 481, l'abonnement original est toujours considéré comme valide pour la durée de la valeur "Expires" la plus récemment connue selon ce qui a été négocié par SUBSCRIBE et sa réponse, ou comme communiqué par NOTIFY dans le paramètre "Expires" de l'en-tête "Subscription-State".

Noter que beaucoup de ces erreurs indiquent qu'il peut y avoir un problème avec le réseau ou le notificateur et qu'aucun autre message NOTIFY ne sera reçu.

3.1.4.3 **Désabonnement**

Le désabonnement est traité de la même façon que le rafraîchissement d'un abonnement, avec l'en-tête "Expires" réglé à "0". Noter qu'un désabonnement réussi va aussi déclencher un message final NOTIFY.

3.1.4.4 **Confirmation de la création d'un abonnement**

L'abonné peut s'attendre à recevoir un message NOTIFY de chaque nœud qui a traité un abonnement réussi ou un rafraîchissement d'abonnement. Jusqu'à ce que le premier message NOTIFY arrive, l'abonné devrait considérer l'état de la ressource à laquelle il s'est abonné comme étant à un stade neutre. Les documents qui définissent de nouveaux paquetages d'événements DOIVENT définir cet "état neutre" de telle façon qu'il ait un sens pour leur application (voir le paragraphe 4.4.7.).

Du fait de la possibilité à la fois de messages déclassés et de fourchement, l'abonné DOIT être prêt à recevoir des messages NOTIFY avant que la transaction SUBSCRIBE ne soit achevée.

Sauf comme noté ci-dessus, le traitement de ce NOTIFY est le même que celui du paragraphe 3.2.4.

3.1.5 Comportement de mandataire SUBSCRIBE

Les mandataires n'ont pas besoin de comportement supplémentaire au delà de celui décrit dans SIP [1] pour prendre en charge SUBSCRIBE. Si un mandataire souhaite voir toutes les demandes SUBSCRIBE et NOTIFY pour un dialogue donné, il DOIT enregistrer l'acheminement de la demande SUBSCRIBE initiale et des demandes NOTIFY qui établissent le dialogue. De tels mandataires DEVRAIENT aussi enregistrer le chemin de toutes les autres demandes SUBSCRIBE et NOTIFY.

Noter que les abonnés et notificateurs peuvent choisir d'utiliser le chiffrement S/MIME des demandes SUBSCRIBE et NOTIFY ; par conséquent, les mandataires ne peuvent pas s'appuyer sur la capacité d'accéder à une information dont il ne serait pas explicitement exigé par SIP [1] qu'elle soit lisible par le mandataire.

3.1.6 Comportement du notificateur de SUBSCRIBE

3.1.6.1 Traitement de la transaction SUBSCRIBE initiale

Une transaction SUBSCRIBE ne devrait en aucun cas s'étendre sur une durée plus longue que le temps nécessaire pour un traitement automatique. En particulier, les notificateurs NE DOIVENT PAS attendre la réponse d'un usager avant de retourner une réponse finale à une demande SUBSCRIBE.

Cette exigence est imposée principalement pour empêcher que le temporisateur F des transactions non-INVITE F (voir [1]) arrive à expiration durant la transaction SUBSCRIBE, car l'interaction avec un usager pourrait souvent dépasser $64 * T1$ secondes.

Le notificateur DEVRAIT vérifier que le paquetage d'événements spécifié dans l'en-tête "Event" est compris. Sinon, le notificateur DEVRAIT retourner une réponse "489 Mauvais événement" pour indiquer que l'événement/classe d'événement spécifié n'est pas compris.

Le notificateur DEVRAIT aussi effectuer les authentifications et autorisations nécessaires selon sa politique locale. Voir au paragraphe 3.1.6.3.

Le notificateur PEUT aussi vérifier que la durée dans l'en-tête "Expires" n'est pas trop faible. Si et seulement si l'intervalle d'expiration est supérieur à zéro ET inférieur à une heure ET inférieur à un minimum configuré par le notificateur, celui-ci PEUT retourner une erreur "423 Intervalle trop court" qui contient un champ d'en-tête "Min-Expires". Le champ d'en-tête "Min-Expires" est décrit dans SIP [1].

Si le notificateur est capable de déterminer immédiatement qu'il comprend le paquetage d'événements, si l'abonné authentifié est autorisé à s'abonner, et si il n'y a pas d'autre entrave à la création de l'abonnement, il crée l'abonnement et un dialogue (si nécessaire) et retourne une réponse "200 OK" (sauf si ce faisant il révélerait de façon indésirable sa politique d'autorisation ; voir le paragraphe 5.2).

Si le notificateur ne peut pas créer immédiatement l'abonnement (par exemple, il a besoin d'attendre une entrée de l'utilisateur pour donner son autorisation, ou s'il agit pour un autre nœud qui n'est pas joignable actuellement) ou s'il souhaite masquer sa politique d'autorisation, il va retourner une réponse "202 Accepté". Cette réponse indique que la demande a été reçue et comprise, mais n'implique pas nécessairement que l'abonnement a déjà été autorisé.

Lorsqu'un abonnement est créé chez le notificateur, il mémorise le nom des paquetages d'événement et le paramètre "id" de l'en-tête "Event" (s'il est présent) au titre des informations d'abonnement.

Les valeurs "Expires" présentes dans les réponses SUBSCRIBE de classe 200 se comportent de la même façon qu'elles le font dans les réponses REGISTER : le serveur PEUT écourter l'intervalle, mais NE DOIT PAS le rallonger.

Si la durée spécifiée dans un message SUBSCRIBE est inacceptablement courte, le notificateur peut être capable d'envoyer une réponse 423, comme décrit plus haut dans la présente section.

Les réponses de classe 200 aux demandes SUBSCRIBE ne contiendront généralement aucune information utile au delà de la durée d'abonnement ; leur principal objet est de servir de mécanisme de fiabilité. Les informations d'état seront communiquées via une demande NOTIFY suivante de la part du notificateur.

Les autres codes de réponse définis dans SIP [1] peuvent être utilisés dans des réponses aux demandes SUBSCRIBE, selon

le cas approprié.

3.1.6.2 Confirmation de création/rafraîchissement d'abonnement

Ayant réussi à accepter ou rafraîchir un abonnement, les notificateurs DOIVENT envoyer immédiatement un message NOTIFY pour communiquer l'état en cours de la ressource à l'abonné. Ce message NOTIFY est envoyé sur le même dialogue que créé par la réponse SUBSCRIBE. Si la ressource n'a pas d'état significatif au moment du traitement du message SUBSCRIBE, ce message NOTIFY PEUT contenir un corps vide ou neutre. Voir au paragraphe 3.2.2. des précisions sur la génération du message NOTIFY.

Noter qu'un message NOTIFY est toujours envoyé immédiatement après toute réponse de classe 200 à une demande SUBSCRIBE, que l'abonnement ait déjà été autorisé ou non.

3.1.6.3 Authentification/autorisation des demandes SUBSCRIBE

Le souci de confidentialité peut exiger des notificateurs qu'ils appliquent une politique pour déterminer si un abonné particulier les autorise à souscrire à certains ensemble d'événements. Une telle politique peut être définie par des mécanismes tels que des listes de contrôle d'accès ou une interaction en temps réel avec l'utilisateur. En général, l'autorisation des abonnés avant l'authentification n'est pas particulièrement utile.

Les mécanismes d'authentification de SIP sont exposés dans SIP [1]. Noter que, même si le nœud notificateur agit normalement comme un mandataire, l'authentification des demandes SUBSCRIBE sera toujours effectuée via une réponse "401", et non une "407" ; les notificateurs agissent toujours comme agents d'utilisateur lorsqu'ils acceptent des abonnements et envoient des notifications.

Bien sûr, en agissant comme un mandataire, un nœud va effectuer l'authentification normale de mandataire (en utilisant une 407). L'explication précédente est un rappel que les notificateurs sont toujours des UA, et comme tels effectuent l'authentification d'UA.

Si l'autorisation échoue sur la base d'une liste d'accès ou quelque autre mécanisme automatisé (c'est-à-dire, il peut être déterminé automatiquement d'autorité que l'abonné n'est pas autorisé à souscrire) le notificateur DEVRAIT répondre à la demande par une réponse "403 Interdit" ou "603 Refus", sauf si faire ainsi pourrait révéler des informations qui devraient rester confidentielles ; voir le paragraphe 5.2.

Si le propriétaire notificateur est interrogé interactivement pour déterminer si un abonnement est admis, une réponse "202 Accept" est retournée immédiatement. Noter qu'un message NOTIFY est cependant formé et envoyé dans ces circonstances, comme décrit au paragraphe précédent.

Si l'autorisation d'abonnement a été retardée et si le notificateur souhaite faire savoir qu'une telle autorisation a été refusée, il peut le faire en envoyant un message NOTIFY contenant un en-tête "État-d'abonnement" avec une valeur de "terminé" et un paramètre de cause de "rejeté".

3.1.6.4 Rafraîchissement des abonnements

Lorsqu'un notificateur reçoit un rafraîchissement d'abonnement, en supposant que l'abonné soit encore autorisé, le notificateur met à jour l'heure d'expiration de l'abonnement. Comme pour un abonnement initial, le serveur PEUT raccourcir la durée jusqu'à l'expiration, mais NE DOIT PAS l'accroître. L'heure d'expiration finale est placée dans l'en-tête "Expires" de la réponse. Si la durée spécifiée dans un message SUBSCRIBE est inacceptablement courte, le notificateur DEVRAIT répondre par un message "423 Abonnement trop bref".

Si aucun rafraîchissement n'est reçu pour une adresse de notification avant son heure d'expiration, l'abonnement est supprimé. Lors de la suppression d'un abonnement, le notificateur DEVRAIT envoyer un message NOTIFY avec une valeur "État-d'abonnement" de "terminé" pour l'informer que l'abonnement a été supprimé. Si un tel message est envoyé, l'en-tête "État-d'abonnement" DEVRAIT contenir un paramètre "raison=expiration".

L'envoi d'un NOTIFY lors de l'expiration d'un abonnement permet au dialogue correspondant de se terminer, si approprié.

3.2 Description du comportement NOTIFY

Les messages NOTIFY sont envoyés pour informer les abonnés des changements de l'état auquel l'abonné a souscrit. Les abonnements sont normalement mis en place en utilisant la méthode SUBSCRIBE ; cependant, il est possible que d'autres moyens aient été utilisés.

Si des mécanismes différents de SUBSCRIBE sont définis pour créer des abonnements, il est de la responsabilité des parties qui définissent ces mécanismes de s'assurer que la corrélation d'un message NOTIFY avec l'abonnement correspondant est possible. Les concepteurs de tels mécanismes sont aussi prévenus d'avoir à faire une distinction entre l'envoi d'un message NOTIFY à un abonné qui est au courant de l'abonnement et l'envoi d'un message NOTIFY à un nœud qui ne se doute de rien. Ce dernier comportement est invalide, et DOIT recevoir une réponse "481 Abonnement inexistant" (sauf si quelque autre code d'erreur de classe 400 ou 500 est plus applicable) comme décrit au paragraphe 3.2.4. En d'autres termes, la connaissance d'un abonnement doit exister à la fois chez l'abonné et chez le notificateur pour être valide, même si elle est installée via un mécanisme non SUBSCRIBE.

Un NOTIFY ne termine pas son abonnement correspondant ; en d'autres termes, une seule demande SUBSCRIBE peut déclencher plusieurs demandes NOTIFY.

3.2.1 Identification des événements, classes d'événement et de l'état en cours rapportés

L'identification des événements qui sont rapportés dans une notification est très similaire à celle décrite pour l'abonnement aux événements (voir au paragraphe 3.1.2.).

Comme dans les demandes SUBSCRIBE, les en-têtes "Event" de NOTIFY vont contenir un seul nom de paquetage d'événements pour lequel une notification est générée. Le nom de paquetage dans l'en-tête "Event" DOIT correspondre à l'en-tête "Event" dans le message SUBSCRIBE correspondant. Si un paramètre "id" était présent dans le message SUBSCRIBE, ce paramètre "id" DOIT aussi être présent dans les messages NOTIFY correspondants.

Les paquetages d'événements peuvent définir une sémantique associée au corps de leurs demandes NOTIFY ; si ils le font, c'est leur sémantique qui s'applique. Les corps de NOTIFY sont supposés fournir des précisions supplémentaires sur la nature de l'événement qui est survenu et l'état de ressource en résultant.

Lorsqu'il est présent, le corps de la demande NOTIFY DOIT être formaté en un des formats de corps spécifiés dans l'en-tête "Accept" de la demande SUBSCRIBE correspondante. Ce corps va contenir l'état de la ressource souscrite ou un pointeur sur un tel état en forme d'URI (voir au paragraphe 4.4.13).

3.2.2 Comportement NOTIFY du notificateur

Lorsqu'une demande SUBSCRIBE reçoit une réponse de classe 200, le notificateur DOIT immédiatement construire et envoyer une demande NOTIFY à l'abonné. Lorsque survient un changement dans l'état souscrit, le notificateur DEVRAIT immédiatement construire et envoyer une demande NOTIFY, sous réserve des considérations d'autorisation, de politique locale, et d'encombrement.

Une demande NOTIFY est considérée comme échouée si la réponse ne parvient pas dans le délai d'expiration, ou si un code de réponse de classe différente de 200 est reçu sans en-tête "Reessai-après" et sans action ultérieure implicite qui pourrait être entreprise pour réessayer la demande (par exemple, "401 Autorisation exigée".)

Si la demande NOTIFY échoue (comme défini ci-dessus) suite à une condition d'expiration de temporisation, et si l'abonnement avait été installé en utilisant un mécanisme à états conditionnels (comme SUBSCRIBE) le notificateur DEVRAIT retirer l'abonnement.

Ce comportement empêche les transmissions inutiles d'informations d'état pour les abonnés qui ont connu une défaillance ou ont disparu du réseau. Comme ces transmissions seront effectuées plusieurs fois, selon l'algorithme de retransmission défini dans SIP [1] (au lieu de la seule transmission normale pour les clients du fonctionnement) continuer de les servir alors qu'aucun client n'est disponible pour en accuser réception place une contrainte induite sur un réseau. Au redémarrage du client ou au rétablissement d'une connexion réseau, on s'attend à ce que les clients envoient des messages SUBSCRIBE pour potentiellement rafraîchir des informations d'état périmées ; de tels messages vont réinstaller les abonnements dans tous les nœuds pertinents.

Si la demande NOTIFY échoue (comme défini ci-dessus) du fait d'une réponse d'erreur, et si l'abonnement avait été installé en utilisant un mécanisme à états conditionnels, le notificateur DOIT retirer les abonnements correspondants.

Une réponse d'erreur NOTIFY serait généralement pour indiquer que quelque chose ne va pas avec l'abonné ou un mandataire sur le chemin de l'abonné. Si l'abonné est en erreur, il paraît très raisonnable de permettre à l'abonné de rectifier la situation (en se réabonnant) une fois que la condition d'erreur a été traitée. Si un mandataire est en erreur, le rafraîchissement périodique de SUBSCRIBE va réinstaller l'état d'abonnement une fois que le problème de réseau est résolu.

Si une demande NOTIFY reçoit une réponse 481, le notificateur DOIT retirer l'abonnement correspondant même si un tel abonnement avait été installé par un moyen non SUBSCRIBE (comme un interface administratif).

Si le comportement ci-dessus n'était pas exigé, les abonnés recevant un NOTIFY pour un abonnement inconnu auraient besoin d'envoyer un code d'état d'erreur en réponse au NOTIFY et aussi d'envoyer une demande SUBSCRIBE pour supprimer l'abonnement. Comme ce comportement rendrait les abonnés disponibles pour être utilisés comme amplificateurs dans des attaques de déni de service, nous avons plutôt choisi de donner à la réponse 481 une signification particulière : elle est utilisée pour indiquer qu'un abonnement doit être annulé dans toutes les circonstances.

Les demandes NOTIFY DOIVENT contenir un en-tête "État-d'abonnement" avec une valeur de "actif", "en cours", ou "terminé". La valeur "actif" indique que l'abonnement a été accepté et a été autorisé (dans la plupart des cas ; voir au paragraphe 5.2.). La valeur "en cours" indique que l'abonnement a été reçu, mais que les informations de politique sont insuffisantes pour accepter ou refuser l'abonnement à ce moment. La valeur "terminé" indique que l'abonnement n'est pas actif.

Si la valeur de l'en-tête "État-d'abonnement" est "actif" ou "en cours", le notificateur DEVRAIT aussi inclure dans l'en-tête "État-d'abonnement" un paramètre "Expires" qui indique le temps restant pour l'abonnement. Le notificateur PEUT utiliser ce mécanisme pour raccourcir un abonnement ; cependant, ce mécanisme NE DOIT PAS être utilisé pour rallonger un abonnement.

Il est utile d'inclure des informations sur l'expiration pour les abonnements actifs et en cours dans le cas de demande SUBSCRIBE fourchue, car la réponse à un SUBSCRIBE fourchu peut n'être pas reçue par l'abonné. Noter bien que cette valeur "expires" est un paramètre dans l'en-tête "État-d'abonnement", ET NON un en-tête "Expires".

Si la valeur de l'en-tête "État-d'abonnement" est "terminé", le notificateur DEVRAIT aussi inclure un paramètre "cause". Le notificateur PEUT aussi inclure un paramètre "réessai-après", le cas échéant. Pour des précisions sur la valeur et la sémantique des paramètres "cause" et "réessai-après", voir au paragraphe 3.2.4.

3.2.3 Comportement du mandataire NOTIFY

Les mandataires n'ont pas besoin de comportement supplémentaire au delà de celui décrit dans SIP [1] pour prendre en charge NOTIFY. Si un mandataire souhaite voir toutes les demandes SUBSCRIBE et NOTIFY pour un dialogue donné, il DOIT enregistrer le chemin de la demande SUBSCRIBE initial et de toute demande NOTIFY établissant le dialogue. De tels mandataires DEVRAIENT aussi enregistrer le chemin de toutes les autres demandes SUBSCRIBE et NOTIFY.

Noter que les abonnés et notificateurs peuvent choisir d'utiliser le chiffrement S/MIME des demandes SUBSCRIBE et NOTIFY ; par conséquent, les mandataires ne peuvent pas s'appuyer sur la capacité à accéder à des informations dont il n'est pas explicitement exigé qu'elles soient lisibles par un mandataire dans SIP [1].

3.2.4 Comportement NOTIFY de l'abonné

À réception d'une demande NOTIFY, l'abonné devrait vérifier qu'elle correspond au moins à un de ses abonnements en cours ; sinon, il DOIT retourner une réponse "481 Abonnement inexistant" sauf si une autre réponse de classe 400 ou 500 est plus appropriée. Les règles pour confronter les demandes NOTIFY aux abonnements qui créent un nouveau dialogue sont décrites au paragraphe 3.3.4. Les notifications pour les abonnements qui ont été créés à l'intérieur d'un dialogue existant correspondent si elles sont dans le même dialogue et si les en-têtes "Event" correspondent (comme décrit au paragraphe 7.2.1.).

Si, pour une raison quelconque, le paquetage d'événements désigné dans l'en-tête "Event" de la demande NOTIFY n'est pas pris en charge, l'abonné va répondre par une "489 Mauvais événement".

Pour empêcher le truquage des événements, les demandes NOTIFY DEVRAIENT être authentifiées, en utilisant tout mécanisme d'authentification défini dans SIP.

Les demandes NOTIFY DOIVENT contenir des en-têtes "État-d'abonnement" qui indiquent l'état de l'abonnement.

Si la valeur de l'en-tête "État-d'abonnement" est "actif", cela signifie que l'abonnement a été accepté et (en général) a été autorisé. Si l'en-tête contient aussi un paramètre "expire", l'abonné DEVRAIT le prendre comme la durée de l'autorisation d'abonnement et se régler en conséquence. Les paramètres "réessai-après" et "raison" n'ont pas de sémantique pour "actif".

Si la valeur de "État-d'abonnement" est "en cours", l'abonnement a été reçu par le notificateur, mais les informations de politique sont insuffisantes pour accorder ou refuser l'abonnement pour l'instant. Si l'en-tête contient aussi un paramètre "expire", l'abonné DEVRAIT le prendre comme la durée de l'autorisation d'abonnement et se régler en conséquence. Aucune autre action n'est nécessaire de la part de l'abonné. Les paramètres "réessai-après" et "raison" n'ont pas de sémantique pour "en cours".

Si la valeur de "État-d'abonnement" est "terminé", l'abonné devrait considérer que l'abonnement est terminé. Le paramètre "expire" n'a pas de sens pour "terminé". Si un code de cause est présent, le client devait se comporter comme décrit ci-dessous. Si aucun code de cause n'est présent ou un code de cause inconnu, le client PEUT essayer de re-souscrire à tout moment (sauf si un paramètre "réessai-après" est présent, auquel cas le client NE DEVRAIT PAS tenter de re-souscrire pendant le nombre de secondes spécifié par le paramètre "réessai-après"). Les codes de cause définis sont :

désactivé : L'abonnement est terminé, mais l'abonné DEVRAIT réessayer immédiatement un nouvel abonnement. Le principal usage de ce code d'état est de permettre la migration des abonnements entre deux nœuds. Le paramètre "réessai-après" n'a pas de sens pour "désactivé".

probation : L'abonnement est terminé, mais le client DEVRAIT réessayer un peu plus tard. Si un paramètre "réessai-après" est aussi présent, le client DEVRAIT attendre au moins pendant le nombre de secondes spécifié par ce paramètre avant de tenter le réabonnement.

rejeté : L'abonnement est terminé à cause d'un changement de la politique d'autorisation. Les clients NE DEVRAIENT PAS essayer de se réabonner. Le paramètre "réessai-après" n'a pas de sens pour "rejeté".

expiration : L'abonnement s'est terminé parce qu'il n'a pas été rafraîchi avant son expiration. Les clients PEUVENT se réabonner immédiatement. Le paramètre "réessai-après" n'a pas de sens pour "expiration".

abandonné : L'abonnement s'est terminé parce que le notificateur n'a pas pu obtenir l'autorisation à temps. Si un paramètre "réessai-après" est aussi présent, le client DEVRAIT attendre au moins le nombre de secondes spécifié par ce paramètre avant d'essayer de se réabonner ; autrement, le client PEUT réessayer immédiatement, mais il sera vraisemblablement remis dans l'état en instance.

pas-de-ressource : L'abonnement s'est terminé parce que l'état de la ressource qui était surveillée n'existe plus. Les clients NE DEVRAIENT PAS essayer de se réabonner. Le paramètre "réessai-après" n'a pas de sens pour "pas-de-ressource".

Une fois que la notification est réputée acceptable pour l'abonné, l'abonné DEVRAIT retourner une réponse 200. En général, il n'est pas prévu que les réponses NOTIFY contiennent un corps ; cependant, elles le PEUVENT, si la demande NOTIFY contenait un en-tête "Accept".

D'autres réponses définies dans SIP [1] peuvent aussi être retournées, selon le cas approprié. Une transaction NOTIFY ne devrait en aucun cas durer plus longtemps que la durée nécessaire pour un traitement automatique. En particulier, les abonnés NE DOIVENT PAS attendre une réponse d'usage avant de retourner une réponse finale à une demande NOTIFY.

3.3 Généralités

3.3.1 Détection de la prise en charge de SUBSCRIBE et NOTIFY

Ni SUBSCRIBE ni NOTIFY ne nécessitent l'utilisation des en-têtes "Require" ou "Proxy-Require" ; de même, aucun jeton n'est défini pour les en-têtes "Supported". Si nécessaire, les clients peuvent tester la prise en charge de SUBSCRIBE et NOTIFY en utilisant la demande OPTIONS définie dans SIP [1].

La présence de l'en-tête "Allow-Events" dans un message est suffisante pour indiquer la prise en charge de SUBSCRIBE et

NOTIFY.

Le paramètre "methods" pour Contact peut aussi être utilisé pour annoncer spécifiquement la prise en charge des messages SUBSCRIBE et NOTIFY lors de l'enregistrement. (Voir à la référence [8] des détails sur le paramètre "methods").

3.3.2 Demandes CANCEL

Aucune sémantique n'est associée à l'annulation de SUBSCRIBE ou NOTIFY.

3.3.3 Fourchement

Conformément aux règles pour le mandatement de demandes non INVITE telles que définies dans SIP [1], les demandes SUBSCRIBE réussies recevront seulement une réponse de classe 200 ; cependant, du fait du fourchement, l'abonnement peut avoir été accepté par plusieurs nœuds. L'abonné DOIT donc être prêt à recevoir des demandes NOTIFY avec des étiquettes "From:" qui diffèrent des étiquettes "To:" reçues dans la réponse SUBSCRIBE de classe 200.

Si plusieurs messages NOTIFY sont reçus dans des dialogues différents en réponse à un seul message SUBSCRIBE, chaque dialogue représente une destination différente à laquelle la demande SUBSCRIBE a été envoyé en fourche. Pour les informations sur le traitement de l'abonné dans de telles situations, voir au paragraphe 4.4.9.

3.3.4 Création et terminaison de dialogue

Si une demande SUBSCRIBE initiale n'est pas envoyée sur un dialogue préexistant, l'abonné va attendre une réponse à la demande SUBSCRIBE ou un NOTIFY correspondant.

Les réponses sont confrontées à de telles demandes SUBSCRIBE pour voir si elles contiennent le même "Call-ID", la même étiquette d'en-tête "From", et le même "CSeq". Les règles pour la comparaison de ces en-têtes sont décrites dans SIP [1]. Si une réponse de classe 200 correspond à une telle demande SUBSCRIBE, elle crée un nouvel abonnement et un nouveau dialogue (sauf si ils ont déjà été créés par une demande NOTIFY correspondante ; voir ci-dessous).

Les demandes NOTIFY sont confrontées à de telles demandes SUBSCRIBE pour voir si elles contiennent le même "Call-ID", un paramètre d'étiquette d'en-tête "To" qui correspond au paramètre d'étiquette d'en-tête "From" de SUBSCRIBE, et le même champ d'en-tête "Event" pour les comparaisons des en-têtes "Event" comme décrit au paragraphe 7.2.1. Si une demande NOTIFY correspondante contient un "État-d'abonnement" de "actif" ou "en cours", elle crée un nouvel abonnement et un nouveau dialogue (sauf si ils ont déjà été créés par une réponse correspondante, comme décrit ci-dessus).

Si un SUBSCRIBE initial est envoyé sur un dialogue préexistant, une réponse correspondante de classe 200 ou une demande NOTIFY réussie crée simplement un nouvel abonnement associé à ce dialogue.

Plusieurs abonnements peuvent être associés à un seul dialogue. Des abonnements peuvent aussi exister dans des dialogues associés à des états d'application créés par INVITE et d'autres états d'application créés par des mécanismes définis dans d'autres spécifications. Ces ensembles d'états d'application n'interagissent pas au-delà du comportement décrit pour un dialogue (par exemple, traitement du réglage d'acheminement).

Un abonnement est détruit lorsque un notificateur envoie une demande NOTIFY avec un "État-d'abonnement" de "terminé".

Un abonné peut envoyer une demande SUBSCRIBE avec un en-tête "Expires" de 0 afin de déclencher l'envoi d'une telle demande NOTIFY ; cependant, pour les besoins de la durée de vie de l'abonnement et du dialogue, l'abonnement n'est pas considéré comme terminé jusqu'à ce que soit envoyé le NOTIFY avec un "État-d'abonnement" de "terminé".

Si la destruction d'un abonnement ne laisse pas d'autre état d'application associé au dialogue, celui-ci se termine. La destruction des autres états d'application (comme ceux créés par un INVITE) ne terminera pas le dialogue si un abonnement est toujours associé à ce dialogue.

Noter que le comportement ci-dessus signifie qu'un dialogue créé avec un INVITE ne se termine pas nécessairement à réception d'un BYE. De même, dans le cas de plusieurs abonnements associés à un seul dialogue, celui-ci ne se termine pas tant que tous les abonnements qu'il contient ne sont pas détruits.

3.3.5 Migration d'agents d'état et de notificateur

Lorsqu'on utilise des agents d'état (voir au paragraphe 4.4.11.) il est souvent utile de permettre la migration des abonnements entre les agents d'état et les nœuds pour lesquels ils fournissent une agrégation d'état (ou même parmi divers agents d'état). Une telle migration peut être effectuée en envoyant un message NOTIFY avec un en-tête "État-d'abonnement" de "terminé", et un paramètre de cause de "désactivé". Cette demande NOTIFY est par ailleurs normale et elle est formée comme décrit au paragraphe 3.2.2.

À réception de ce message NOTIFY, l'abonné DEVRAIT essayer de se réabonner (comme décrit dans les paragraphes précédents). Noter que cet abonnement est établi sur un nouveau dialogue, et ne réutilise pas l'établissement d'acheminement du dialogue d'abonnement précédent.

La migration réelle est effectuée en faisant un changement de la politique (comme les décisions d'acheminement) d'un ou plusieurs serveurs auxquels la demande SUBSCRIBE sera envoyée de telle sorte qu'un nœud différent termine la réponse à la demande SUBSCRIBE. Cela peut être aussi simple qu'un changement de la politique locale chez le notificateur duquel l'abonnement migre de façon qu'il serve de serveur mandataire ou de redirection au lieu d'être un notificateur.

Si, quand et pourquoi effectuer des migrations de notificateur peut être décrit dans les paquetages d'événements individuels ; autrement, de telles décisions sont une question de politique locale du notificateur, et sont laissées à la discrétion des mises en œuvre individuelles.

3.3.6 Consultation de l'état de la ressource

Une conséquence naturelle du comportement décrit aux paragraphes précédents est qu'un apport immédiat sans un abonnement permanent peut être effectué par l'envoi d'un SUBSCRIBE avec un "Expires" de 0.

Bien sûr, un apport immédiat alors qu'un abonnement est actif peut être effectué par l'envoi d'un SUBSCRIBE avec un "Expires" égal au nombre de secondes restant à l'abonnement.

À réception de cette demande SUBSCRIBE, le notificateur (ou les notificateurs, si la demande SUBSCRIBE était fourchée) enverra une demande NOTIFY contenant l'état de la ressource dans le même dialogue.

Noter que les messages NOTIFY déclenchés par des messages SUBSCRIBE avec des en-têtes "Expires" de 0 contiendront une valeur de "État-d'abonnement" de "terminé, et un paramètre "cause" de "expiration".

La consultation de l'état d'événement peut causer un accroissement significatif de la charge du réseau et des notificateurs ; à ce titre, elle ne devrait être utilisée qu'avec parcimonie. En particulier, la consultation NE DEVRAIT PAS être utilisée dans des circonstances où elle va normalement résulter en plus de messages réseau que d'abonnements à long terme.

Lorsque la consultation est utilisée, les abonnés DEVRAIENT essayer de mettre en antémémoire les accreditifs d'authentification entre les consultations afin de réduire le nombre de messages envoyés.

3.3.7 Utilisation de l'en-tête Allow-Events

L'en-tête "Allow-Events", s'il est présent, inclut une liste de jetons qui indiquent les paquetages d'événements pris en charge par le client (si ils sont envoyés dans une demande) ou le serveur (si ils sont envoyés dans une réponse). En d'autres termes, un nœud qui envoie un en-tête "Allow-Events" avertit qu'il peut traiter les demandes SUBSCRIBE et générer des demandes NOTIFY pour tous les paquetages d'événements listés dans cet en-tête.

Tout nœud qui met en œuvre un ou plusieurs paquetages d'événements DEVRAIT inclure un en-tête "Allow-Events" approprié indiquant tous les événements pris en charge dans toutes les méthodes qui initient des dialogues et leurs réponses (comme INVITE) et les réponses OPTIONS.

Ces informations sont très utiles, par exemple, pour permettre aux agents d'utilisateur de rendre les éléments d'interface particuliers de façon appropriée conformément au fait que les événements dont il est exigé qu'ils mettent en œuvre les caractéristiques qu'ils représentent sont pris en charge par les nœuds appropriés.

Noter que les en-têtes "Allow-Events" NE DOIVENT PAS être insérés par des mandataires.

3.3.8 Compatibilité PINT

L'en-tête "Event" est considéré comme obligatoire pour les besoins du présent document. Cependant, pour conserver la

compatibilité avec PINT (voir [2]) les serveurs PEUVENT interpréter une demande SUBSCRIBE sans en-tête "Event" comme demandant un événement à des événements PINT. Si un serveur ne prend pas en charge PINT, il DEVRAIT retourner "489 Mauvais événement" à tout message SUBSCRIBE sans en-tête "Event".

4 Paquetages d'événements

La présente section traite de diverses questions qui devraient être prises en considération lors de la proposition de paquetages d'événements fondés sur SUBSCRIBE et NOTIFY.

4.1 Adéquation de l'utilisation

Lors de la conception d'un paquetage d'événements utilisant les méthodes décrites dans le présent document pour une notification d'événement, il est important de considérer si : SIP est un mécanisme approprié pour le problème posé ? SIP est choisi à cause d'une caractéristique unique fournie par le protocole (par exemple, la mobilité de l'utilisateur) ou simplement parce que "il peut le faire ?" Si vous vous trouvez vous-même en train de définir des paquetages d'événements pour des notifications qui se rapportent, par exemple, à la gestion de réseau ou à la température du moteur de votre véhicule, vous pourriez vouloir reconsidérer votre choix de protocoles.

Ceux qui sont intéressés par l'extension des mécanismes définis dans le présent document sont invités à suivre le développement des "Lignes directrices pour les auteurs d'extensions SIP" [7] pour des indications complémentaires au sujet des utilisations appropriées de SIP.

De plus, il est prévu que ce mécanisme ne soit pas utilisé dans des applications où la fréquence de événements à rapporter est excessivement rapide (par exemple, plus d'un par seconde). Un réseau SIP va généralement être provisionné pour un volume de signalisation raisonnable ; l'envoi d'une notification chaque fois que la position GPS d'un usager change d'un centième de seconde pourrait facilement surcharger un tel réseau.

4.2 Gabarit de paquetage d'événements

Les paquetages d'événements normaux définissent un ensemble d'états appliqués à un type de ressource spécifique, comme la présence d'un usager, un état d'appel, et un état de boîte de messagerie.

Les gabarits de paquetage d'événements sont un type particulier de paquetage qui définit un ensemble d'états appliqués à d'autres paquetages, comme des statistiques, une politique d'accès, et des listes d'abonnés. Les gabarits de paquetage d'événements peuvent même être appliqués à d'autres gabarits de paquetage d'événements.

Pour étendre l'analogie orientée objet faite plus haut, des gabarits de paquetage d'événements peuvent être vus comme des paquetages C++ normalisés qui doivent être appliqués à d'autres paquetages pour être utiles.

Le nom d'un gabarit de paquetage d'événement tel qu'appliqué à un paquetage est formé par l'ajout d'un point suivi par le nom du gabarit de paquetage d'événement à la fin du paquetage. Par exemple, si un gabarit de paquetage appelé "winfo" s'appliquait à un paquetage appelé "presence", le jeton d'événement utilisé dans "Event" et "Allow-Events" serait "presence.winfo".

Les gabarits de paquetage d'événement doivent être définis de telle sorte qu'ils puissent s'appliquer à tout paquetage arbitraire. En d'autres termes, les gabarits de paquetage d'événements ne peuvent pas être spécifiquement liés à un ou quelques paquetages "parent" d'une façon qui ferait qu'ils ne puissent fonctionner avec les autres paquetages.

4.3 Quantité d'états à convoier

Lors de la conception de paquetages d'événements, il est important de considérer le type des informations qui seront convoiées durant une notification.

Une tentation naturelle est de convoier simplement l'événement (par exemple, "un nouveau message vocal vient d'arriver") sans état pour l'accompagner (par exemple, "7 messages vocaux"). Cela complique la mise en œuvre des entités abonnées (car elles doivent entretenir l'état complet pour l'entité à laquelle elles se sont abonnées), et c'est aussi particulièrement susceptible de problèmes de synchronisation.

Les paquetages d'événements peuvent choisir entre la mise en œuvre de deux solutions possibles à ce problème.

4.3.1 Informations d'état complètes

Pour les paquetages qui portent normalement des informations d'état raisonnablement petites (de l'ordre de 1 kbit environ) il est suggéré que les paquetages d'événements soient conçus de façon à envoyer des informations d'état complètes lorsque survient un événement.

Dans certaines circonstances, convoier le seul état en cours peut être insuffisant pour une classe d'événements particulière. Dans ce cas, les paquetages d'événements devraient inclure des informations d'état complètes ainsi que l'événement qui est survenu. Par exemple, porter "pas de représentant du service consommateur disponible" peut n'être pas aussi utile que de dire "pas de représentant du service consommateur disponible ; le représentant sip:46@cs.xyz.int vient juste de se déconnecter".

4.3.2 Différences d'états

Dans le cas où les informations d'état à convoier sont importantes, le paquetage d'événements peut choisir de montrer un schéma par lequel les messages NOTIFY contiennent les différences d'état au lieu de l'état complet.

Un tel schéma pourrait fonctionner comme suit : tout NOTIFY qui envoie une réponse immédiate à un SUBSCRIBE contient les informations d'état complètes. Les messages NOTIFY envoyés à cause d'un changement d'état ne vont contenir que les informations d'état qui ont changé ; l'abonné fusionnera alors ces informations avec ses autres connaissances sur l'état de la ressource.

Tout paquetage d'événements qui accepte les différences de changements d'états DOIT inclure un numéro de version qui augmente exactement de un pour chaque transaction NOTIFY d'un abonnement. Noter que le numéro de version d'état apparaît dans le corps du message, et non dans un en-tête SIP.

Si un NOTIFY arrive avec un numéro de version qui est augmenté de plus de un, l'abonné sait qu'une différence d'état a été manquée ; il ignore le message NOTIFY qui contient la différence d'état (sauf pour le numéro de version, qu'il conserve pour détecter la perte de message) et renvoie un SUBSCRIBE pour forcer un NOTIFY contenant un instantané complet de l'état.

4.4 Responsabilités des paquetages d'événements

Les paquetages d'événements ne sont pas obligés de réitérer un des comportements décrits dans ce document, bien qu'ils puissent choisir de le faire dans un souci de clarté ou d'emphase. En général, cependant, de tels paquetages sont prévus pour ne décrire que le comportement qui étend ou modifie le comportement décrit dans le présent document.

Noter que tout comportement conçu avec "DEVRAIT" ou "DOIT" dans ce document n'est pas autorisé à être affaibli pour des documents d'extension ; cependant, de tels documents peuvent choisir de renforcer les exigences "DEVRAIT" conditionnelles en exigences "DOIT" fortes si cela est exigé par leur application.

En plus des paragraphes normaux attendus par les RFC sur la voie de la normalisation et les documents d'extension SIP, les auteurs de paquetages d'événements doivent traiter chacune des questions détaillées dans les paragraphes suivants, chaque fois qu'ils sont applicables.

4.4.1 Nom de paquetage d'événements

Ce paragraphe, qui DOIT être présent, définit le nom du jeton à utiliser pour désigner le paquetage d'événements. Il DOIT inclure les informations qui apparaissent dans l'enregistrement du jeton par l'IANA. Pour les informations sur l'enregistrement de ces types, voir la section 6.

4.4.2 Paramètres de paquetage d'événements

Si des paramètres doivent être utilisés dans l'en-tête "Event" pour modifier le comportement du paquetage d'événements, la syntaxe et la sémantique de tels en-têtes DOIT être clairement définie.

4.4.3 Corps de messages SUBSCRIBE

Il est prévu que la plupart, mais pas la totalité, des paquetages d'événements définissent la syntaxe et la sémantique des corps de messages SUBSCRIBE ; ces corps vont normalement modifier, étendre, filtrer, restreindre, et/ou établir des seuils pour la classe des événements demandés. Les concepteurs de paquetages d'événements sont fortement encouragés à

réutiliser les types MIME existants pour les corps de message lorsque c'est possible.

Ce paragraphe obligatoire de paquetage d'événements définit quel type ou quels types de corps d'événement sont attendus dans les demandes SUBSCRIBE (ou spécifie qu'aucun corps d'événement n'est attendu). Il devrait comporter des définitions détaillées de la syntaxe et de la sémantique pour tous les types de corps référencés.

4.4.4 Durée d'abonnement

Il est RECOMMANDÉ que les paquetages d'événements donnent une gamme de temps suggérée considérée comme raisonnable pour la durée d'un abonnement. De tels paquetages DOIVENT aussi définir une valeur "Expires" par défaut à utiliser si aucune n'est spécifiée.

4.4.5 Corps de message NOTIFY

Le corps NOTIFY est utilisé pour rapporter l'état de la ressource surveillée. Chaque paquetage DOIT définir quel type ou quels types de corps d'événement sont attendus dans les demandes NOTIFY. De tels paquetages DOIVENT spécifier ou citer les spécifications détaillées pour la syntaxe et la sémantique associées à de tels corps d'événement.

Les paquetages d'événements DOIVENT aussi définir quel type MIME doit être supposé si aucun n'est spécifié dans l'en-tête "Accept" de la demande SUBSCRIBE.

4.4.6 Traitement des demandes SUBSCRIBE par le notificateur

Ce paragraphe décrit le traitement à effectuer par le notificateur à réception d'une demande SUBSCRIBE. Un tel paragraphe est exigé.

Les informations de ce paragraphe comportent des détails sur la façon d'authentifier les abonnés et les questions d'autorisation pour le paquetage. De telles questions d'autorisation peuvent comporter, par exemple, si toutes les demandes SUBSCRIBE pour ce paquetage reçoivent des réponses 202 (voir le paragraphe 5.2.).

4.4.7 Génération des demandes NOTIFY par le notificateur

Ce paragraphe de paquetage d'événements décrit le processus par lequel le notificateur génère et envoie une demande NOTIFY. Cela inclut des informations détaillées sur les événements qui causent l'envoi d'un NOTIFY, comment calculer les informations d'état dans le NOTIFY, comment générer des informations d'état neutres ou fausses pour cacher les délais d'autorisation et de décisions aux usagers, et si les informations d'état sont complètes ou leurs variations pour les notifications ; voir le paragraphe 4.3. Un tel paragraphe est exigé.

Ce paragraphe peut facultativement décrire le comportement utilisé pour traiter la réponse résultante.

4.4.8 Traitement des demandes NOTIFY par l'abonné

Ce paragraphe de paquetage d'événements décrit le processus suivi par l'abonné à réception d'une demande NOTIFY, et inclut toute la logique nécessaire pour former un état de ressource cohérent (si c'est applicable).

4.4.9 Traitement des demandes fourchées

Chaque paquetage d'événements DOIT spécifier si les demandes SUBSCRIBE fourchées sont admises pour installer plusieurs abonnements.

Si un tel comportement n'est pas admis, le premier message établissant un dialogue potentiel va créer un dialogue. Tous les messages NOTIFY ultérieurs qui correspondent au message SUBSCRIBE (c'est-à-dire, correspondent à "To", "From", au paramètre "tag" de l'en-tête "From", à "Call-ID", "CSeq", "Event", et au paramètre "id" de l'en-tête "Event") mais qui ne correspondent pas au dialogue, seraient rejetés avec une réponse 481. Noter que la réponse de classe 200 au SUBSCRIBE peut arriver après la réception d'un NOTIFY correspondant ; de telles réponses peuvent ne pas se corréliser au même dialogue que celui établi par le NOTIFY. Excepté quand c'est nécessaire pour terminer la transaction SUBSCRIBE, de telles réponses non correspondantes de classe 200 sont ignorées.

Si l'installation de plusieurs abonnements au moyen d'un seul SUBSCRIBE fourché est admise, l'abonné établit un nouveau dialogue avec chaque notificateur en retournant une réponse de classe 200 à chaque NOTIFY. Chaque dialogue est alors traité comme une entité autonome, et est rafraîchi indépendamment des autres dialogues.

Dans le cas où plusieurs abonnements sont admis, le paquetage d'événements DOIT spécifier si la fusion des notifications

pour former un seul état est demandée, et comment une telle fusion doit être effectuée. Noter qu'il est possible que certains paquetages d'événements soient définis de telle façon que chaque dialogue soit lié à un état mutuellement exclusif qui ne soit pas affecté par les autres dialogues ; ceci DOIT être clairement établi si c'est le cas.

4.4.10 Débit des notifications

Chaque paquetage d'événements est supposé définir une exigence (de force DEVRAIT ou DOIT) qui fixe un maximum absolu sur le débit auquel la génération des notifications est admise pour un seul notificateur.

Chaque paquetage PEUT ensuite définir un mécanisme de restriction qui permette aux abonnés de limiter encore le taux de notification.

4.4.11 Agents d'état

Les concepteurs de paquetages d'événements devraient considérer si leur paquetage peut tirer avantage des points d'agrégation du réseau (agents d'état) et/ou des nœuds au nom desquels ils agissent sur les autres nœuds. (Par exemple, les nœuds qui fournissent des informations d'état sur une ressource quand une telle ressource est incapable ou non désireuse de fournir de telles informations d'état par elle-même). Un exemple d'une telle application est un nœud qui retrace la présence et la disponibilité d'un usager dans le réseau.

Si les agents d'état sont utilisés par le paquetage, le paquetage DOIT spécifier comment de tels agents d'état agrègent les informations et comment ils fournissent l'authentification et l'autorisation.

Les paquetages d'événements PEUVENT aussi développer des scénarios spécifiques dans lesquels prennent place les migrations de notificateur.

4.4.12 Exemples

Les paquetages d'événements DEVRAIENT inclure plusieurs diagrammes de démonstration de flux de message appariés avec plusieurs messages complets typiques, syntaxiquement corrects.

Il est RECOMMANDÉ que les documents qui décrivent les paquetages d'événements indiquent clairement que de tels exemples sont pour information et ne sont pas normatifs, avec des instructions auxquelles les développeurs font référence dans le corps principal du document pour les détails exacts du protocole.

4.4.13 Utilisation des URI pour restituer l'état

Certains types de paquetages d'événements peuvent définir des informations d'état qui sont potentiellement trop grandes pour les envoyer raisonnablement dans un message SIP. Pour atténuer ce problème, les paquetages d'événements peuvent inclure la capacité à convoier un URI au lieu des informations d'état ; cet URI sera alors utilisé pour restituer les informations d'état réelles.

Le mécanisme précis pour convoier de tels URI sort du domaine d'application du présent document.

5 Considérations pour la sécurité

5.1 Contrôle d'accès

La capacité à accepter des abonnements devrait être sous le contrôle direct de l'utilisateur notificateur, car de nombreux types d'événements peuvent être considérés comme sensibles pour les besoins de la confidentialité. De même, le notificateur devrait avoir la capacité de rejeter de façon sélective les abonnements sur la base de l'identité de l'abonné (d'après les listes de contrôle d'accès) en utilisant les mécanismes SIP standard d'authentification. Les méthodes de création et de distribution de telles listes de contrôle d'accès sortent du domaine d'application du présent document.

5.2 Mécanisme de confidentialité du notificateur

Le simple acte de retourner des réponses 200 ou certaines réponses 4xx et 6xx aux demandes SUBSCRIBE peut, dans certaines circonstances, créer des problèmes de confidentialité en révélant des informations sensibles de politique. Dans ce cas, le notificateur DEVRAIT toujours retourner une réponse 202. Alors que le message NOTIFY qui suit peut ne pas convoier le véritable état, il DOIT paraître contenir un ensemble de données potentiellement correct du point de vue de l'abonné, indistinguishable d'une réponse valide. Les informations sur le point de savoir si un usager est autorisé à s'abonner à l'état demandé n'est jamais renvoyé à l'utilisateur d'origine dans ces circonstances.

Les paquetages individuels, et les documents qui s'y rapportent, pour lesquels un tel mode de fonctionnement a un sens peuvent décrire plus avant comment et pourquoi générer de telles données potentiellement correctes. Par exemple, un tel mode de fonctionnement est rendu obligatoire par la RFC 2779 [6] pour les informations de présence de l'utilisateur.

5.3 Attaques de déni de service

Le modèle actuel (une demande SUBSCRIBE déclenche une réponse SUBSCRIBE et une ou plusieurs demandes NOTIFY) est un établissement classique pour l'utilisation d'un nœud amplificateur dans une attaque de stroumf.

Aussi, la création d'un état à réception d'une demande SUBSCRIBE peut être utilisée par des attaquants pour consommer des ressources sur la machine d'une victime, la rendant inutilisable.

Pour réduire les chances de succès d'une telle attaque, les mises en œuvre de notificateurs DEVRAIENT exiger l'authentification. Les questions d'authentification sont exposées dans SIP [1].

5.4 Attaques en répétition

La répétition de SUBSCRIBE ou de NOTIFY peut avoir des effets dommageables.

Dans le cas des messages SUBSCRIBE, les attaquants peuvent être capables d'installer tout abonnement arbitraire dont ils ont été témoins de l'installation à un moment dans le passé. Répéter des messages NOTIFY peut être utilisé pour usurper de vieilles informations d'état (bien qu'un bon mécanisme de gestion des versions dans le corps des messages NOTIFY puisse aider à atténuer une telle attaque). Noter que la prohibition de l'envoi de messages NOTIFY aux nœuds qui n'ont pas souscrit à un événement aide aussi à atténuer les effets d'une telle attaque.

Pour empêcher de telles attaques, les mises en œuvre DEVRAIENT exiger l'authentification avec une protection anti répétition. les questions d'authentification sont exposées dans SIP [1].

5.5 Attaques par intrusion

Même avec l'authentification, des attaques par intrusion utilisant SUBSCRIBE peuvent se faire pour installer des abonnements arbitraires, capturer des abonnements existants, terminer des abonnements en cours, ou modifier la ressource à laquelle un abonnement a été fait. Pour empêcher de telles attaques, les mises en œuvre DEVRAIENT fournir la protection d'intégrité à travers les en-têtes "Contact", "Route", "Expires", "Event", et "To" des messages SUBSCRIBE, au minimum. Si les corps SUBSCRIBE sont utilisés pour définir des informations complémentaires sur l'état de l'appel, ils DEVRAIENT être inclus dans le schéma de protection d'intégrité.

Les attaques par intrusion peuvent aussi tenter d'utiliser les messages NOTIFY pour usurper des informations d'état arbitraires et/ou mettre fin à des abonnements en cours. Pour empêcher de telles attaques, les mises en œuvre DEVRAIENT fournir la protection de l'intégrité à travers les en-têtes "Call-ID", "CSeq", et "État-d'abonnement" et les corps de messages NOTIFY.

La protection d'intégrité des en-têtes de message et des corps est exposée dans SIP [1].

5.6 Confidentialité

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

Quoique moins vraisemblable, il est aussi possible que les informations contenues dans un message SUBSCRIBE contiennent des informations que les usagers pourraient ne pas souhaiter voir révélées. Les mises en œuvre PEUVENT chiffrer de telles informations pour assurer leur confidentialité.

Pour permettre à la partie distante de cacher les informations qu'elle considère comme sensibles, toutes les mises en œuvre DEVRAIENT être capable de traiter les messages SUBSCRIBE et NOTIFY chiffrés.

Les mécanismes pour fournir la confidentialité sont détaillées dans SIP [1].

6 Considérations relatives à l'IANA

Le présent document définit un espace de nom de type d'événement qui exige un corps de coordination central. Le corps choisi pour cette coordination est l'Autorité d'allocation des numéros de l'Internet (IANA, *Internet Assigned Numbers Authority*).

Il y a deux différents types de type d'événement : les paquetages d'événements normaux, et les gabarits de paquetage d'événement ; voir le paragraphe 4.2. Pour éviter toute confusion, les noms de gabarit de paquetage et les noms de paquetages partagent le même espace de nom ; en d'autres termes, un gabarit de paquetage d'événement NE DOIT PAS partager un nom avec un paquetage.

Suivant les politiques soulignées dans les "Lignes directrices pour la rédaction d'une section de Considérations relatives à l'IANA dans les RFC" [4], les jetons d'identification de paquetage d'événements normal sont alloués sur la base du premier arrivé, premier servi, et les jetons d'identification des gabarits de paquetage d'événement sont alloués sur la base du consensus de l'IETF.

Les enregistrements auprès de l'IANA DOIVENT inclure le jeton enregistré et si le jeton est un paquetage ou un gabarit de paquetage. De plus, les paquetages DOIVENT inclure les informations de contact pour la partie responsable de l'enregistrement et/ou du document publié qui décrit le paquetage d'événements. Les enregistrements de jetons de gabarit de paquetage d'événement DOIVENT inclure un pointeur sur la RFC publiée qui définit la gabarit de paquetage d'événement.

Les jetons enregistrés pour désigner les paquetages et les gabarits de paquetages NE DOIVENT PAS contenir le caractère ".", qui est utilisé pour séparer les gabarits de paquetages des paquetages.

6.1 Informations d'enregistrement

Comme le présent document ne spécifie aucun nom de paquetage ou de gabarit de paquetages, l'enregistrement initial de l'IANA pour les types d'événement sera vide. Le reste du texte de la présente section donne un exemple du type des informations à conserver par l'IANA ; il montre aussi les cinq permutations possibles de type de paquetage, de contact, et de référence.

Le tableau ci-dessous fait la liste des paquetages d'événements et gabarits de paquetages définis dans "Notification d'événement spécifique de SIP" [RFC3265]. Chaque nom est conçu comme un paquetage ou un gabarit de paquetage sous "Type".

Nom de paquetage	Type	Contact	Référence
exemple1	paquetage	[Roach]	
exemple2	paquetage	[Roach]	[RFC3265]
exemple3	paquetage	[Roach]	
exemple4	gabarit	[Roach]	[RFC3265]
exemple5	gabarit		[RFC3265]

PERSONNE

[Roach] Adam Roach <adam@dynamicsoft.com>

RÉFÉRENCES

[RFC3265] Roach, A., "Notification d'événement spécifique de SIP", RFC 3265, juin 2002.

6.2 Gabarit d'enregistrement

À : ietf-sip-events@iana.org

Sujet : Enregistrement de nouveaux paquetage d'événements SIP

Nom de paquetage : (Les noms de paquetage doivent se conformer à la syntaxe décrite au paragraphe 7.2.1.)

Est-ce un enregistrement d'un gabarit de paquetage : (indiquer oui ou non)

Spécification(s) publiée (s) : (Les gabarits de paquetages exigent une RFC publiée. Les autres paquetages peuvent faire référence à une spécification si c'est approprié).

Adresse personnelle et de messagerie électronique à contacter pour des informations complémentaires :

6.3 Noms de champ d'en-tête

Le présent document enregistre trois nouveaux noms de champ d'en-tête , décrits dans d'autres parties du document. Ces en-têtes sont définis par les informations suivantes, qui sont à ajouter au sous-registre d'en-tête sous <http://www.iana.org/assignments/sip-parameters>.

Nom d'en-tête : Allow-Events

Forme compacte : u

Nom d'en-tête : Subscription-State

Forme compacte : (aucune)

Nom d'en-tête : Event

Forme compacte : o

6.4 Codes de réponse

Le présent document enregistre deux nouveaux codes de réponse. Ces codes de réponse sont définis par les informations suivantes, qui sont à ajouter au sous-registre de méthode et code de réponse à l'adresse <http://www.iana.org/assignments/sip-parameters>.

Numéro de code de réponse : 202

Phrase de cause par défaut : Accepted

Numéro de code de réponse : 489

Phrase de cause par défaut : Bad Event

7 Syntaxe

La présente section décrit les extensions de syntaxe exigées pour la notification d'événement dans SIP. la sémantique est décrite à la section 3. Noter que les définitions de syntaxe formelle décrites dans le présent document sont exprimées dans le format ABNF utilisé dans SIP [1], et contiennent des références aux éléments qui y sont définis.

7.1 Nouvelles méthodes

Le présent document décrit deux nouvelles méthodes SIP : SUBSCRIBE et NOTIFY.

Ce tableau développe les tableaux 2 et 3 de SIP [1].

En-tête	où	SUB	NOT
Accept	R	o	o
Accept	2xx	-	-
Accept	415	o	o
Accept-Encoding	R	o	o
Accept-Encoding	2xx	-	-
Accept-Encoding	415	o	o
Accept-Language	R	o	o
Accept-Language	2xx	-	-
Accept-Language	415	o	o
Alert-Info	R	-	-
Alert-Info	180	-	-
Allow	R	o	o
Allow	2xx	o	o
Allow	r	o	o
Allow	405	m	m
Authentication-Info	2xx	o	o
Authorization	R	o	o
Call-ID	c	m	m

Contact	R	m	m
Contact	1xx	o	o
Contact	2xx	m	o
Contact	3xx	m	m
Contact	485	o	o
Content-Disposition		o	o
Content-Encoding		o	o
Content-Language		o	o
Content-Length		t	t
Content-Type		*	*
CSeq	c	m	m
Date		o	o
Error-Info	300-699	o	o
Expires		o	-
Expires	2xx	m	-
From	c	m	m
In-Reply-To	R	-	-
Max-Forwards	R	m	m
Min-Expires	423	m	-
MIME-Version		o	o
Organization		o	-
Priority	R	o	-
Proxy-Authenticate	407	m	m
Proxy-Authorization	R	o	o
Proxy-Require	R	o	o
RAck	R	-	-
Record-Route	R	o	o
Record-Route	2xx,401,484	o	o
Reply-To		-	-
Require		o	o
Retry-After	404,413,480,486	o	o
Retry-After	500,503	o	o
Retry-After	600,603	o	o
Route	R	c	c
RSeq	1xx	o	o
Server	r	o	o
Subject	R	-	-
Supported	R	o	o
Supported	2xx	o	o
Timestamp		o	o
To	c(1)	m	m
Unsupported ²	420	o	o
User-Agent		o	o
Via	c	m	m
Warning	R	-	o
Warning	r	o	o
WWW-Authenticate	401	m	m

7.1.1 Méthode SUBSCRIBE

"SUBSCRIBE" est ajouté à la définition de l'élément "Method" dans la grammaire de message SIP.

Comme tous les noms de méthode SIP, le nom de méthode SUBSCRIBE est sensible à la casse. La méthode SUBSCRIBE sert à demander la notification asynchrone d'un événement ou ensemble d'événements ultérieurs.

7.1.2 Méthode NOTIFY

"NOTIFY" est ajouté à la définition de l'élément "Method" dans la grammaire de message SIP.

La méthode NOTIFY sert à notifier à un nœud SIP qu'un événement qui a été demandé par une méthode SUBSCRIBE antérieure est survenu. Elle peut aussi donner des précisions sur l'événement.

7.2 Nouveaux en-têtes

Le présent tableau développe les tableaux 2 et 3 de SIP [1], tels qu'amendés par les changements décrits au paragraphe 7.1.

Champ d'en-tête	où	proxy ACK	BYE	CAN	INV	OPT	REG	PRA	SUB	NOT
Allow-Events	R	o	o	-	o	o	o	o	o	o
Allow-Events	2xx	-	o	-	o	o	o	o	o	o
Allow-Events	489	-	-	-	-	-	-	-	m	m
Event	R	-	-	-	-	-	-	-	m	m
Subscription-State	R	-	-	-	-	-	-	-	-	m

7.2.1 En-tête "Event"

Event est ajouté à la définition de l'élément "message-header" dans la grammaire de message SIP.

Pour les besoins de la confrontation des réponses et messages NOTIFY avec les messages SUBSCRIBE, la portion type d'événement de l'en-tête "Event" est comparée octet par octet, et le jeton de paramètre "id" (s'il est présent) est comparé octet par octet. Un en-tête "Event" contenant un paramètre "id" ne correspond jamais à un en-tête "Event" sans paramètre "id". Aucun autre paramètre n'est pris en compte lors d'une comparaison.

Noter que le texte précédent signifie que "Event: foo; id=1234" correspondrait à "Event: foo; param=abcd; id=1234", mais pas "Event: foo" (l'identifiant ne correspond pas) ou "Event: Foo; id=1234" (la portion événement ne correspond pas).

Le présent document ne définit pas de valeurs pour les types d'événement. Ces valeurs seront définies par les paquets d'événements individuels, et DOIVENT être enregistrées auprès de l'IANA.

Il DOIT y avoir exactement un type d'événement listé par en-tête d'événement. Plusieurs événements par message n'est pas permis.

7.2.2 En-tête "Allow-Events"

Allow-Events est ajouté à la définition de l'élément "general-header" dans la grammaire de message SIP. Son usage est décrit au paragraphe 3.3.7.

7.2.3 En-tête "Subscription-State"

Subscription-State est ajouté à la définition de l'élément "request-header" dans la grammaire de message SIP. Son usage est décrit au paragraphe 3.2.4.

7.3 Nouveaux codes de réponse

7.3.1 Code de réponse "202 Accepté"

La réponse 202 est ajoutée à la définition du champ d'en-tête "Success". "202 Accepté" a la même signification que celle définie dans HTTP/1.1 [3].

7.3.2 Code de réponse "489 Mauvais événement"

La réponse d'événement 489 est ajoutée à la définition de champ d'en-tête "Client-Error". "489 Mauvais événement" sert à indiquer que le serveur n'a pas compris le paquetage d'événements spécifié dans un champ d'en-tête "Event".

7.4 Définitions de BNF augmenté

Les définitions de BNF augmenté pour les divers éléments de syntaxe nouveaux et modifié figurent ci-après. La notation est celle utilisée dans SIP [1], et tout élément non défini dans ce paragraphe est tel que défini dans SIP et les documents auxquels il se réfère.

SUBSCRIBEm = %x53.55.42.53.43.52.49.42.45 ; SUBSCRIBE in caps
 NOTIFYm = %x4E.4F.54.49.46.59 ; NOTIFY in caps
 extension-method = SUBSCRIBEm / NOTIFYm / token

Event = ("Event" / "o") HCOLON event-type
 *(SEMI event-param)

event-type = event-package *("." event-template)
 event-package = token-nodot
 event-template = token-nodot
 token-nodot = 1*(alphanum / "-" / "!" / "%" / "*" / "_" / "+" / "/" / "" / "~")
 event-param = generic-param / ("id" EQUAL token)

Allow-Events = ("Allow-Events" / "u") HCOLON event-type
 *(COMMA event-type)

Subscription-State = "Subscription-State" HCOLON substate-value
 *(SEMI subexp-params)

substate-value = "active" / "pending" / "terminated" / extension-substate
 extension-substate = token
 subexp-params = ("reason" EQUAL event-reason-value)
 / ("expires" EQUAL delta-seconds)
 / ("retry-after" EQUAL delta-seconds)
 / generic-param

event-reason-value = "deactivated"
 / "probation"
 / "rejected"
 / "timeout"
 / "giveup"
 / "noresource"
 / event-reason-extension

event-reason-extension = token

8 Références normatives

- [1] [RFC3361] H. Schulzrinne, "[Option du protocole de configuration dynamique d'hôte](#) (DHCP-pour-IPv4) pour les serveurs du protocole d'initialisation de session (SIP)", août 2002. *(P.S.)*
- [2] [RFC2848] S. Petrack, L. Conroy, "[Protocole de service PINT](#) : extensions à SIP et SDP pour l'accès IP aux services de téléphone", juin 2000. *(P.S.)*
- [3] [RFC2616] R. Fielding et autres, "[Protocole de transfert hypertexte](#) -- HTTP/1.1", juin 1999. *(D.S., MàJ par 2817, 6585)*
- [4] [RFC2434] T. Narten et H. Alvestrand, "Lignes directrices pour la rédaction d'une section Considérations relatives à l'IANA dans les RFC", BCP 26, octobre 1998. *(Rendue obsolète par la RFC5226)*
- [5] [RFC2119] S. Bradner, "[Mots clés à utiliser](#) dans les RFC pour indiquer les niveaux d'exigence", BCP 14, mars 1997. DOI 10.17487/RFC2119, *(MàJ par RFC8174)*
- [6] [RFC2779] M. Day et autres, "[Exigences des protocoles Messagerie instantanée / Presence](#)", février 2000. *(Information)*

9 Références pour information

- [7] [RFC4485] J. Rosenberg, H. Schulzrinne, "Lignes directrices pour les auteurs d'extensions au protocole d'initialisation de session (SIP)", mai 2006. *(Information)*
- [8] [RFC3841] J. Rosenberg, H. Schulzrinne, P. Kyzivat, "[Préférences de l'appelant](#) pour le protocole d'initialisation de session (SIP)", août 2004. *(P.S.)*

10 Remerciements

Merci aux participants aux événements BOF à la 48^{ème} réunion de l'IETF à Pittsburgh, ainsi qu'à ceux qui ont apporté des idées et des suggestions sur la liste de diffusion du SIP Events. Je tiens en particulier à remercier Henning Schulzrinne de la Columbia University pour son apport du schéma final d'identification d'événement en trois parties, Sean Olson pour ses nombreux conseils, Jonathan Rosenberg pour sa révision scrupuleuse du projet -00, et les auteurs du document "SIP Extensions for Presence" pour leurs apports à la signification des demandes SUBSCRIBE et NOTIFY.

11 Notice concernant les droits de propriété intellectuelle

L'IETF a reçu notification de revendications de droits de propriété intellectuelle à l'égard de tout ou partie des spécifications contenues dans la présent document. Pour des précisions, consulter la liste en ligne à