

Groupe de travail Réseau
Request for Comments : 5057
 Catégorie : Information

R. Sparks, Estacado Systems
 novembre 2007
 Traduction Claude Brière de L'Isle

Dialogues à usages multiples dans le protocole d'initialisation de session

Statut de ce mémoire

Le présent mémoire donne des informations pour la communauté de l'Internet. Il ne spécifie aucune norme de l'Internet. La distribution du présent mémoire n'est soumise à aucune restriction.

Résumé

Dans le protocole d'initialisation de session (SIP, *Session Initiation Protocol*) plusieurs méthodes peuvent créer une association entre des points d'extrémité qu'on appelle un dialogue. Certaines de ces méthodes peuvent aussi créer une association différente, mais toujours en rapport avec un dialogue existant. Ces associations ou usages multiples de dialogues exigent un traitement soigneusement coordonné car elles ont des cycles de vie indépendants mais partagent un état de dialogue commun. Le traitement correct des usages multiples de dialogues n'est pas entièrement compris. Ce qui est compris est difficile à mettre en œuvre.

Le présent mémoire expose que les usages multiples de dialogues devraient être évités. Il discute des solutions de remplacement à leur utilisation et précise les comportements essentiels pour les éléments qui ne peuvent en fait les éviter.

Ce document est pour information et ne fait aucune sorte de déclaration normative.

Table des Matières

1. Généralités.....	1
2. Introduction.....	2
3. Exemples d'usages multiples.....	2
3.1 Transfert.....	2
3.2 Souscription réciproque.....	4
4. Création et destruction d'usage.....	5
4.1 Usages de Invite.....	5
4.2 Usages de Subscribe.....	6
5. Traitement approprié d'usages multiples.....	6
5.1 Survol de l'effet de réponses d'échec sur les usages et dialogues.....	6
5.2. Fin de temporisation de transaction.....	9
5.3 Confrontation des demandes aux usages.....	10
5.4 Demandes de rafraîchissement de cible.....	10
5.5 Rafraîchissement et terminaison d'usages.....	10
5.6 Refus de nouveaux usages.....	11
5.7 Remplacement d'usages.....	11
6. Évitement de multiples usages.....	11
7. Considérations sur la sécurité.....	14
8. Conclusion.....	15
9. Remerciements.....	15
10. Références pour information.....	15
Adresse de l'auteur.....	16
Déclaration complète de droits de reproduction.....	16

1. Généralités

Le présent document est pour information. Il ne fait aucune sorte de déclaration normative. Le présent document précise le concept d'un usage de dialogue dans le protocole d'initiation de session (SIP, *Session Initiation Protocol*) [RFC3261], et discute ce qui a conduit à son existence. Il explore l'ambiguïté associée au traitement de plusieurs usages de dialogue qui partagent un dialogue. En particulier, il examine l'effet des réponses d'échec SIP sur la transaction, l'usage du dialogue, et l'état du dialogue. Le présent document aidera les mises en œuvre à comprendre ce qui est exigé pour traiter correctement plusieurs usages de dialogue, et fournira des informations pour de futurs travaux sur la voie de la normalisation qui vont

préciser la RFC 3261 et autres documents en rapport. Finalement, le document explore des solutions de remplacement de dialogue à un seul usage (en utilisant les extensions à SIP) aux usages de dialogue multiples.

2. Introduction

Plusieurs méthodes dans SIP peuvent établir un dialogue. Quand elles le font, elles établissent aussi une association entre les points d'extrémité au sein de ce dialogue. Cette association a été appelée pendant un certain temps un "usage de dialogue" dans la communauté des développeurs. Un dialogue initié avec une demande INVITE a un usage d'invite. Un dialogue initié avec une demande SUBSCRIBE a un usage d'abonnement. Un dialogue initié avec une demande REFER a un usage d'abonnement.

Des dialogues avec plusieurs usages surviennent quand une action créant un usage se produit au sein d'un dialogue existant. De telles actions incluent d'accepter un REFER ou SUBSCRIBE produit dans un dialogue établi avec une demande INVITE. Plusieurs REFER au sein d'un dialogue créent plusieurs abonnements, dont chacun est un nouvel usage de dialogue qui partagent un état de dialogue commun. (Noter que tout REFER produit en utilisant le mécanisme de suppression d'abonnement spécifié dans la [RFC4488] ne crée pas de nouvel usage.) De même, un point d'extrémité dans un dialogue établi avec un INVITE pourrait s'abonner au langage de balisage à pression de touche (KPML, *Key Press Markup Language*) [RFC4730] de son homologue et produire ultérieurement un REFER, résultant en trois usages de dialogue partageant un état de dialogue commun.

L'état commun dans le dialogue partagé par tous les usages est exactement :

- o l'identifiant d'appel
- o l'étiquette locale
- o l'étiquette distante
- o le CSeq local
- o le CSeq distant
- o l'ensemble de chemins
- o le contact local
- o la cible distante
- o le fanion sécurisé

Les usages ont un état qui n'est pas partagé dans le dialogue. Par exemple, un abonnement a une durée, ainsi qu'un autre état spécifique de l'usage. Plusieurs abonnements dans le même dialogue ont chacun leur propre durée.

Un dialogue vient à l'existence avec la création du premier usage, et continue d'exister jusqu'à la fin du dernier usage (comptage de référence). Malheureusement, de nombreux aspects de la gestion d'usage de SIP, comme l'authentification, ont été à l'origine conçus avec l'hypothèse implicite qu'il avait un seul usage par dialogue. Les mécanismes résultants ont des effets mélangés, certains influençant l'usage, et certains influençant le dialogue entier.

Les spécifications actuelles définissent deux usages, invite et subscribe. Un dialogue peut partager jusqu'à un usage de invite et un nombre arbitraire de subscribe.

Parce que la [RFC3261] déclare que les agents d'utilisateur devraient réutiliser l'identifiant d'appel et incrémenter le CSeq sur une série de demandes d'enregistrement (et que les to-tag apparaissent dans les réponses register dans certains des exemples) certaines mises en œuvre ont traité REGISTER comme si il était dans un dialogue. Cependant, la RFC 3261 déclare explicitement que REGISTER ne crée pas un dialogue. Une série de demandes REGISTER ne crée aucun usage ou dialogue. De même, PUBLISH [RFC3903] ne crée aucun usage ou dialogue.

3. Exemples d'usages multiples

3.1 Transfert

Dans la Figure 1, Alice transfère à Carol un appel qu'elle a reçu de Bob. Un dialogue (et un usage de dialogue invite) entre Alice et Bob commence avec le 200 OK marqué F1. Un second usage (un abonnement à l'événement refer) commence avec le NOTIFY marqué F2. Ce second usage se termine quand l'abonnement est terminé par la transaction NOTIFY marquée F3. Le dialogue a encore un usage (l'usage invite) qui dure jusqu'à la transaction BYE marquée F4. À ce point, le dialogue n'a plus d'usage restant, et cesse donc d'exister. Les détails de chacun de ces messages sont montrés à la Figure 2.

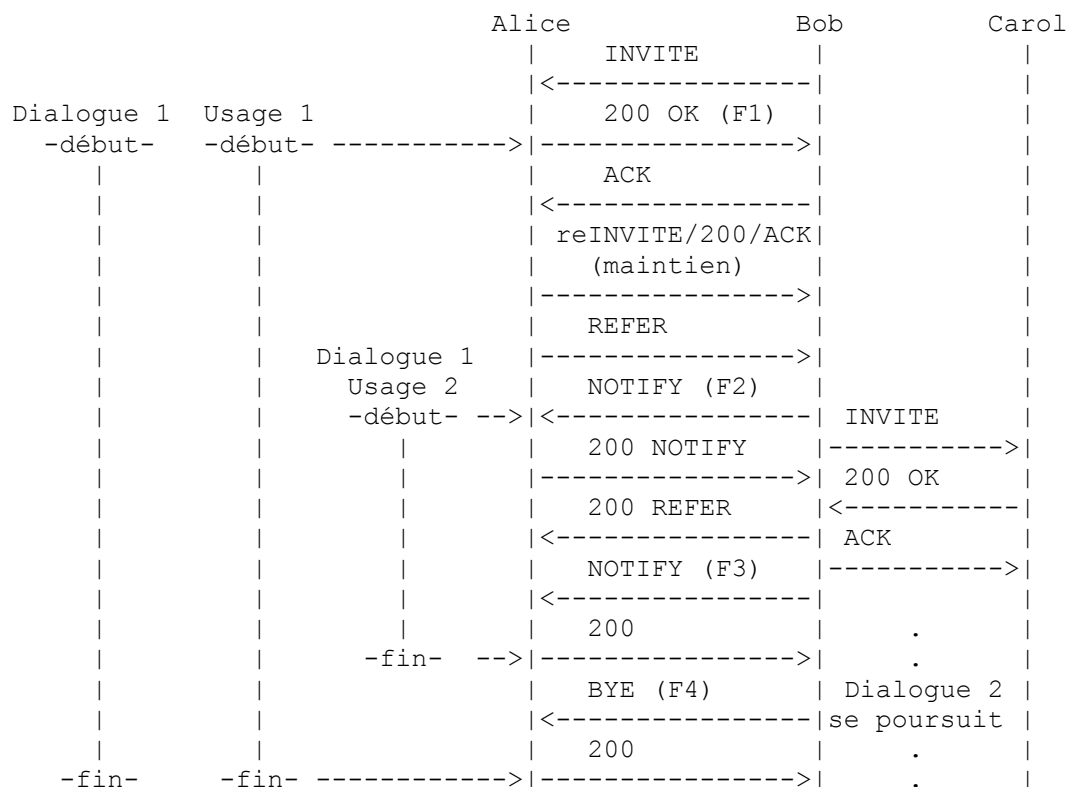


Figure 1

Détails de message (abrégés pour montrer seulement les détails de dialogue ou usage)

F1

```
SIP/2.0 200 OK
Call-ID: dialog1@bob.exemple.com
CSeq: 100 INVITE
To: <sip:Alice@alice.exemple.com>;tag=alicetag1
From: <sip:Bob@bob.exemple.com>;tag=bobtag1
Contact: <sip:aliceinstance@alice.exemple.com>
```

F2

```
NOTIFY sip:aliceinstance@alice.exemple.com SIP/2.0
Event: refer
Call-ID: dialog1@bob.exemple.com
CSeq: 101 NOTIFY
To: <sip:Alice@alice.exemple.com>;tag=alicetag1
From: <sip:Bob@bob.exemple.com>;tag=bobtag1
Contact: <sip:bobinstance@bob.exemple.com>
```

F3

```
NOTIFY sip:aliceinstance@alice.exemple.com SIP/2.0
Event: refer
Subscription-State: terminé ; raison=pas de ressource
Call-ID: dialog1@bob.exemple.com
CSeq: 102 NOTIFY
To: <sip:Alice@alice.exemple.com>;tag=alicetag1
From: <sip:Bob@bob.exemple.com>;tag=bobtag1
Contact: <sip:bobinstance@bob.exemple.com>
Content-Type: message/sipfrag
SIP/2.0 200 OK
```

F4

```
BYE sip:aliceinstance@alice.exemple.com SIP/2.0
```

```

Call-ID: dialog1@bob.exemple.com
CSeq: 103 BYE
To: <sip:Alice@alice.exemple.com>;tag=alicetag1
From: <sip:Bob@bob.exemple.com>;tag=bobtag1
Contact: <sip:bobinstance@bob.exemple.com>

```

Figure 2

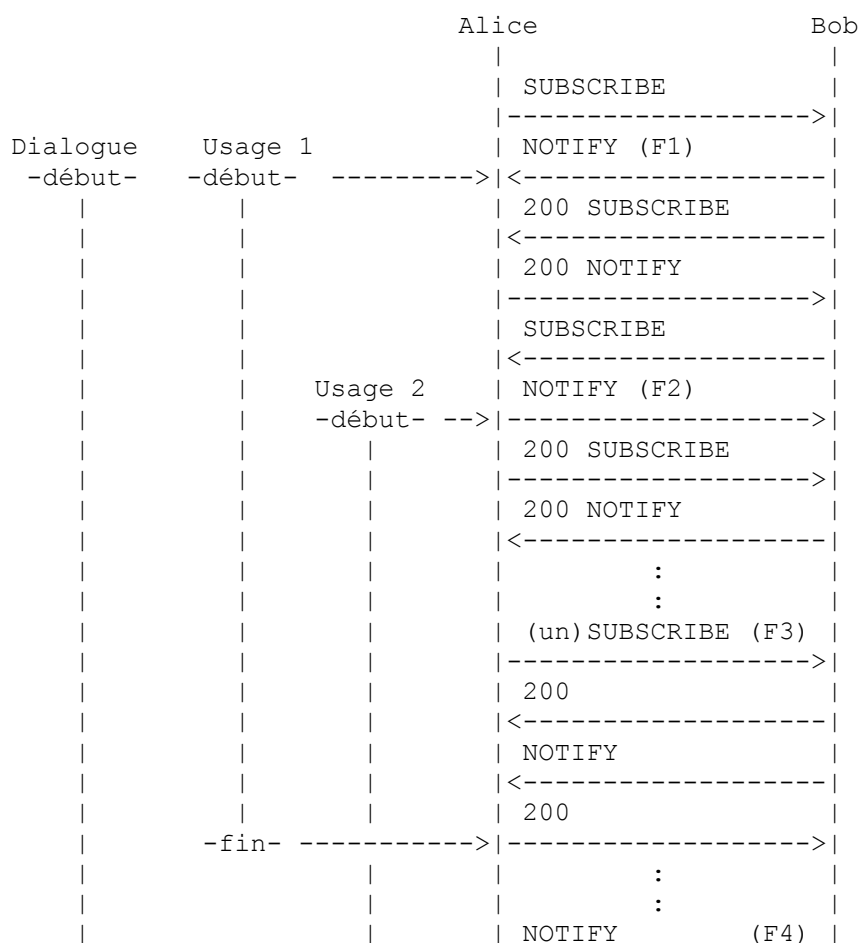
3.2 Souscription réciproque

Dans la Figure 3, Alice s'abonne à la présence de Bob. Pour simplifier, on suppose que Bob et Alice servent tous deux leur présence à partir de leurs points d'extrémité au lieu d'un serveur de présence. Pour se concentrer sur les points essentiels, la figure laisse de côté toute signalisation de rendez-vous par laquelle Alice découvre le point d'extrémité de Bob.

Bob est aussi intéressé à la présence d'Alice, de sorte qu'il s'abonne à Alice (dans la plupart des systèmes déployés de présence/IM, les gens se regardent les uns les autres). Il décide de sauter l'étape rendez-vous car il est déjà dans un dialogue avec Alice, et il envoie son SUBSCRIBE dans ce dialogue (quelques clients de SIMPLE se conduisaient exactement comme cela).

Le dialogue et son premier usage commence en F1, qui établit l'abonnement de Alice à Bob. Son second usage commence à F2, qui établit l'abonnement de Bob à Alice. Ces deux abonnements sont indépendants ; ils ont des moments d'expirations distincts et différents, mais partagent tous l'état de dialogue.

Le premier usage se termine quand Alice décide de se désabonner en F3. L'abonnement de Bob à Alice, et donc le dialogue, continuent d'exister. L'UA d'Alice doit maintenir cet état de dialogue même si l'abonnement qui a causé son existence en premier lieu est maintenant terminé. Le second usage se termine quand Alice décide de terminer l'abonnement de Bob en F4 (elle va probablement rejeter toute tentative de la part de Bob de se réabonner jusqu'à ce qu'elle soit prête à s'abonner de nouveau à Bob). Comme c'était le dernier usage, le dialogue se termine aussi. Les détails de ces messages sont montrés à la Figure 4.



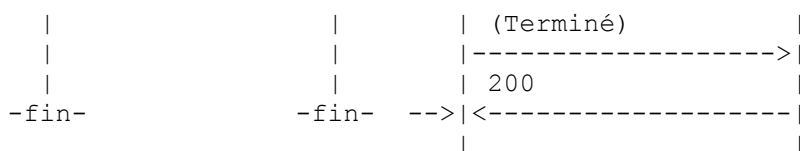


Figure 3

Détails de message (abrégés pour montrer seulement les détails de dialogue ou usage)

F1

```

NOTIFY sip:aliceinstance@alice.exemple.com SIP/2.0
Event: presence
Subscription-State: active;expires=600
Call-ID: alicecallid1@alice.exemple.com
From: <sip:Bob@bob.exemple.com>;tag=bobtag2
To: <sip:Alice@alice.exemple.com>;tag=alicetag2
CSeq: 100 NOTIFY
Contact: <sip:bobinstance@bob.exemple.com>

```

F2

```

NOTIFY sip:bobinstance@bob.exemple.com SIP/2.0
Event: presence
Subscription-State: active;expires=1200
Call-ID: alicecallid1@alice.exemple.com
To: <sip:Bob@bob.exemple.com>;tag=bobtag2
From: <sip:Alice@alice.exemple.com>;tag=alicetag2
CSeq: 500 NOTIFY
Contact: <sip:aliceinstance@alice.exemple.com>

```

F3

```

SUBSCRIBE sip:bobinstance@bob.exemple.com SIP/2.0
Event: presence
Expires: 0
Call-ID: alicecallid1@alice.exemple.com
To: <sip:Bob@bob.exemple.com>;tag=bobtag2
From: <sip:Alice@alice.exemple.com>;tag=alicetag2
CSeq: 501 SUBSCRIBE
Contact: <sip:aliceinstance@alice.exemple.com>

```

F4

```

NOTIFY sip:bobinstance@bob.exemple.com SIP/2.0
Event: presence
Subscription-State: terminated;reason=deactivated
Call-ID: alicecallid1@alice.exemple.com
To: <sip:Bob@bob.exemple.com>;tag=bobtag2
From: <sip:Alice@alice.exemple.com>;tag=alicetag2
CSeq: 502 NOTIFY
Contact: <sip:aliceinstance@alice.exemple.com>

```

Figure 4

4. Création et destruction d'usage

Les dialogues viennent à l'existence avec leur premier usage. Les dialogues se terminent quand leur dernier usage est détruit. Les messages qui créent et détruisent les usages varient selon l'usage. Cette Section donne une catégorisation générale de ces messages. Elle ne tente pas d'explorer le pseudo-dialogue REGISTER.

4.1 Usages de Invite

Crée par : des réponses non 100 provisoires à INVITE ; une réponse 200 à INVITE

Détruit par : des réponses 200 à BYE ; certaines réponses d'échec à INVITE, UPDATE, PRACK, INFO, ou BYE ; tout ce qui détruit un dialogue et tous ses usages

4.2 Usages de Subscribe

Créé par : des réponses de classe 200 à SUBSCRIBE ; des réponses de classe 200 à REFER ; des demandes NOTIFY.

Détruit par : des réponses de classe 200 à NOTIFY-terminé ; une fin de temporisation de demande NOTIFY ou rafraîchissement-SUBSCRIBE ; certaines réponses d'échec à NOTIFY ou SUBSCRIBE ; expiration sans rafraîchissement si des problèmes de réseau empêchent le NOTIFY terminal d'arriver ; tout ce qui détruit un dialogue et tous ses usages

5. Traitement approprié d'usages multiples

Les exemples de la Section 3 montrent des cas directs où le moment de début et de fin du dialogue est tout à fait évident. Malheureusement, il y a de nombreux scénarios où ce n'est pas aussi clair. Par exemple, à la Figure 1, que voudrait dire une réponse 481 au NOTIFY (F2) ? Cela termine-t-il simplement l'abonnement refer, ou cela détruit-il le dialogue entier ? Cette Section explore les zones de problème avec plusieurs usages qui ont été identifiées à ce jour.

5.1 Survol de l'effet de réponses d'échec sur les usages et dialogues

Pour cette enquête, on considère un usage subscribe à l'intérieur d'un dialogue établi avec un usage invite. Sauf mention contraire, on va discuter de l'effet sur chaque usage et le dialogue quand un client qui produit un NOTIFY à l'intérieur de l'usagee subscribe reçoit une réponse d'échec (comme un transférant qui produit un NOTIFY pour référencer un événement). De plus, sauf mention contraire, les conclusions s'appliquent à des usages multiples arbitraires. Cette enquête est faite du point de vue d'un client qui reçoit la réponse d'erreur. L'effet sur les dialogues et les usages au serveur qui produit la réponse est le même.

Réponses 3xx : la redirection à mi-dialogue n'est pas bien comprise dans SIP, mais quel que soit l'effet, il a un impact égal sur le dialogue entier et tous ses usages. Dans notre exemple de scénario, les deux usages abonnement et invite vont être redirigés par cette seule réponse.

Pour les réponses d'échec avec le code 400 et au-dessus, il y a trois façons courantes dont l'échec peut affecter la transaction, l'usage, et l'état de dialogue.

Transaction seule : l'erreur affecte seulement la transaction, pas l'usage ni le dialogue où la transaction se produit (au delà d'affecter le CSeq local). Aucun autre usage du dialogue n'est affecté. L'erreur est une plainte sur cette transaction, pas sur l'usage ou le dialogue dans lequel la transaction se produit.

Détruit l'usage : l'erreur détruit l'usage, mais pas le dialogue. Aucun autre usage partageant ce dialogue n'est affecté.

Détruit le dialogue : l'erreur détruit le dialogue et tous les usages qui le partagent.

Les Tableaux 1 et 2 affichent comment les divers codes affectent la transaction, l'usage, ou l'état de dialogue. Des commentaires ou exceptions spécifiques du code de réponse suivent le tableau.

Transaction seule	Détruit l'usage	Détruit le dialogue
400 (ou 4xx inconnu)	405, 480	404, 410, 416
401, 402, 403, 406	481, 489	482, 483
407, 408, 412-415	501	484, 485
417, 420, 421, 422		502, 604
423, 428, 429		
436-438, 486, 487		
488, 491, 493, 494		
500 (ou 5xx inconnu)		
503, 504, 505		

513, 580
600 (ou 6xx inconnu)
603, 606

Tableau 1

Code	Raison	Impact	Notes
400/4xx	Mauvaise demande	Transaction	
401	Non autorisé	Transaction	
402	Payement exigé	Transaction	(1)
403	Interdit	Transaction	
404	Pas trouvé	Dialogue	(2)
405	Méthode non admise	Usage	(3)
406	Non acceptable	Transaction	
407	Authentification du mandataire exigée	Transaction	
408	Expiration de la demande	Transaction	(4)
410	Parti	Dialogue	(2)
412	Échec de demande conditionnelle	Transaction	
413	Entité de demande trop grande	Transaction	
414	URI de demande trop long	Transaction	
415	Type de support non pris en charge	Transaction	
416	Schéma d'URI non pris en charge	Dialogue	(2)
417	Priorité de ressource inconnue	Transaction	
420	Mauvaise extension	Transaction	
421	Extension exigée	Transaction	
422	Intervalle de session trop petit	Transaction	(5)
423	Intervalle trop bref	Transaction	
428	Utiliser l'en-tête Identity	Transaction	
429	Fournir l'identité du référent	Transaction	(6)
436	Mauvaises informations d'identité	Transaction	
437	Certificat non pris en charge	Transaction	
438	En-tête d'identité invalide	Transaction	
480	Temporairement indisponible	Usage	(7)
481	Appel/transaction non existant	Usage	(8)
482	Boucle détectée	Dialog	(9)
483	Trop de bonds	Dialogue	(10)
484	Adresse incomplète	Dialogue	(2)
485	Ambigu	Dialogue	(2)
486	Occupé ici	Transaction	(11)
487	Demande terminée	Transaction	
488	Non acceptable ici	Transaction	
489	Mauvais événement	Usage	(12)
491	Demande en instance	Transaction	
493	Indéchiffrable	Transaction	
494	Accord de sécurité exigé	Transaction	
500/5xx	Erreur interne du serveur	Transaction	(13)
501	Non mis en œuvre	Usage	(3)
502	Mauvaise passerelle	Dialogue	(14)
503	Service indisponible	Transaction	(15)
504	Fin de temporisation du serveur	Transaction	(16)
505	Version non prise en charge	Transaction	
513	Message trop long	Transaction	
580	Échec de précondition	Transaction	
600/6xx	Occupé partout	Transaction	(17)
603	Refus	Transaction	
604	N'existe nulle part	Dialogue	(2)
606	Non acceptable	Transaction	

Tableau 2

- (1) 402 Paiement exigé : ce code de réponse est réservé. Si il est rencontré, il devrait être traité comme un 4xx non reconnu.
- (2) 404 Non trouvé
410 Parti
416 Schéma d'URI non pris en charge
484 Adresse incomplète
485 Ambigu
604 N'existe nulle part
- L'URI de demande qui est rejeté est la cible distante établie par le contact fourni par l'homologue. Cette réponse signifie que quelque chose ne va fondamentalement pas dans l'état de dialogue.
- (3) 405 Méthode non admise
501 Non mis en œuvre
Une de ces réponses va être aberrante dans notre exemple de scénario car la prise en charge de la méthode NOTIFY est exigée par l'usage. Dans ce cas, l'UA sait que la condition est irrécupérable et devrait arrêter d'envoyer des NOTIFY sur l'usage. Tout rafraîchissement d'abonnement devrait être rejeté. En général, ces erreurs vont affecter l'usage. Si la demande n'était pas intégrée à l'usage (utilisée comme méthode inconnue, ou était une INFO à l'intérieur d'un usage INVITE, par exemple) seule la transaction va être affectée.
- (4) 408 Fin de temporisation de demande : recevoir un 408 va avoir le même effet sur les usages et dialogues qu'une réelle fin de temporisation de transaction comme décrit au paragraphe 5.2.
- (5) 422 Intervalle de session trop petit : cette réponse n'a pas de sens pour toute demande de mi-usage. Si elle est reçue, un élément dans le chemin de la demande viole le protocole, et le receveur devrait traiter cela comme une réponse 4xx inconnue.
- (6) 429 Fournir l'identité du référant : cette réponse ne va pas être retournée à un NOTIFY comme dans notre exemple de scénario, mais quand elle est retournée à un REFER, elle vise seulement la demande REFER elle-même.
- (7) 480 Temporairement indisponible : la RFC 3261 n'est pas claire sur ce que signifie cette réponse pour les demandes de mi-usage. De futures mises à jour à cette spécification sont attendues pour préciser que cette réponse affecte seulement l'usage dans lequel la demande se produit. Aucun autre usage n'est affecté. Si la réponse incluait un champ d'en-tête Retry-After, d'autres demandes dans cet usage ne devraient pas être envoyées avant que le temps indiqué soit passé. Les demandes dans d'autres usages peuvent quand même être envoyées à tout moment.
- (8) 481 Appel/transaction non existant : cette réponse indique que l'homologue a perdu sa copie de l'état d'utilisation de dialogue. Le dialogue lui-même ne devrait pas être détruit sauf si c'était le dernier usage.
Les effets d'un 481 sur un dialogue et ses usages sont les plus ambigus de toute réponse finale. Il y a des mises en œuvre qui ont choisi la signification recommandée ici, et d'autres qui détruisent le dialogue entier sans égard au nombre d'usages en instance. Aller plus loin dans cette clarification va permettre aux mises en œuvre déployées qui supposent seulement que l'usage a été détruit de fonctionner avec un plus grand nombre de mises en œuvre. Les mises en œuvre existantes qui détruisent tous les autres usages dans le dialogue vont continuer de fonctionner comme elles le font actuellement, sauf que les homologues qui suivent la recommandation vont tenter de faire des choses avec les autres usages et cet élément va retourner des 481 pour chacun d'eux jusqu'à ce qu'ils soient tous partis. Cependant, la précision nécessaire à la RFC 3261 doit rendre très clair que la capacité de terminer des usages indépendamment du dialogue global en utilisant un 481 n'est pas une justification pour concevoir de nouvelles applications qui comptent sur des usages multiples dans un dialogue.
La réponse 481 à une demande CANCEL doit être traitée différemment. Pour CANCEL, un 481 signifie que l'UAS ne peut pas trouver une transaction correspondante. Une réponse 481 à un CANCEL affecte seulement la transaction CANCEL. L'usage associé à l'INVITE n'est pas affecté.
- (9) 482 Boucle détectée : cette réponse est aberrante à mi-dialogue. Elle va seulement se produire si le champ d'en-tête Record-Route a été improprement construit par les mandataires impliqués dans l'établissement de l'usage initial du dialogue, ou si une demande de mi-dialogue fourche et fusionne (ce qui ne devrait jamais arriver). Les futures demandes qui utilisent cet état de dialogue vont aussi échouer.
Une condition limite existe durant la reprise sur défaillance de la RFC 3263 à l'élément qui envoie une demande, où la demande fourche effectivement sur de multiples destinations à partir du client. Certaines mises en œuvre augmentent les risques en entrant dans cette condition limite en essayant très rapidement la prochaine localisation potentielle comme

déterminé par la RFC 3263 si la première ne répond pas immédiatement. Dans une situation où un client envoie la même demande à plus d'un point d'extrémité, il doit être prêt à recevoir une réponse de chaque branche (et devrait choisir une "meilleure" réponse pour agir suivant les mêmes lignes directrices qu'un mandataire fourcheur). Dans cette condition de concurrence particulière, si plusieurs branches répondent, toutes sauf une vont très probablement retourner un 482 Demande fusionnée. Le client devrait choisir la réponse restante non 482 comme "meilleure" réponse.

- (10) 483 Trop de bonds : comme pour 482, recevoir cela à mi-dialogue est aberrant. À la différence de 482, la récupération doit être possible en augmentant Max-Forwards (en supposant que le demandeur a fait quelque chose d'étrange comme d'utiliser une plus petite valeur pour Max-Forwards dans des demandes à mi-dialogue que celle utilisée pour la demande initiale). Si la demande n'est pas essayée avec un Max-Forwards augmenté, alors l'agent devrait suivre les actions de destruction de dialogue.
- (11) 486 Occupé ici : cette réponse n'a pas de sens dans notre exemple de scénario, ou dans tout scénario où cette réponse viendrait dans un usage établi. Si elle arrive dans ce contexte, elle devrait être traitée comme une réponse 4xx inconnue.
- (12) 489 Mauvais événement : dans notre exemple de scénario, la [RFC3265] déclare que l'usage de souscription dans lequel le NOTIFY est envoyé est terminé. Cette réponse est seulement valide dans le contexte de SUBSCRIBE et NOTIFY. Le comportement de l'UA pour recevoir cette réponse à d'autres méthodes n'est pas spécifié, mais la traiter comme un 4xx inconnu est une pratique raisonnable.
- (13) 500 et 5xx réponses non reconnues : Si la réponse contient une valeur de champ d'en-tête Retry-After, le serveur pense que la condition est temporaire, et la demande peut être réessayée après l'intervalle indiqué. Si la réponse ne contient pas de valeur de champ d'en-tête Retry-After, l'UA peut décider de réessayer après un intervalle de son choix ou tenter de terminer l'usage en douceur. Terminer ou non les autres usages dépend de l'application. Si l'UA reçoit une réponse 500 (ou 5xx non reconnu) à une tentative de terminer cet usage en douceur, il peut traiter cet usage comme terminé. Si c'est le dernier usage du dialogue, celui-ci est aussi terminé.
- (14) 502 Mauvaise passerelle : cette réponse est aberrante à mi-dialogue. Elle va seulement se produire si le champ d'en-tête Record-Route a été incorrectement construit par les mandataires impliqués dans l'établissement de l'usage initial du dialogue. Les futures demandes utilisant cet état de dialogue vont aussi échouer.
- (15) 503 Service indisponible : selon la [RFC3263], la logique de traitement de la localisation des serveurs SIP pour les transactions peut traiter les demandes 503 (effectivement, fourcher séquentiellement au point d'extrémité sur la base des résultats du DNS). Si ce processus ne donne pas une meilleure réponse, un 503 peut être retourné à l'utilisateur de la transaction. Comme une réponse 500, l'erreur est une plainte sur cette transaction, non sur l'usage. Parce que cette réponse s'est produite dans le contexte d'un usage établi (donc d'un dialogue existant) l'ensemble de chemins a déjà été formé et toute opportunité d'essayer d'autres serveurs (comme recommandé dans la [RFC3261]) a été épuisée par la logique de la RFC3263.
- (16) 504 Fin de temporisation du serveur : on ne voit pas trop dans quelles circonstances cette réponse pourrait être retournée à une demande dans un dialogue existant.
- (17) 600 et 6xx Réponses non reconnues : à la différence du 400 Mauvaise demande, un code de réponse 600 dit quelque chose sur l'utilisateur receveur, non sur la demande qui est faite. Cet utilisateur final déclare sa volonté de ne pas communiquer. Si la réponse contient une valeur de champ d'en-tête Retry-After, l'usager indique qu'il veut communiquer plus tard et la demande peut être réessayée après l'intervalle indiqué. Cet usage, et tous autres usages partageant le dialogue sont non affectés. Si la réponse ne contient pas de valeur de champ d'en-tête Retry-After, l'UA peut décider de réessayer après un intervalle de son choix ou tenter de terminer l'usage en douceur. Terminer ou non les autres usages dépend de l'application. Si l'UA reçoit un 600 (ou un 6xx non reconnu) en réponse à une tentative de terminer en douceur cet usage, il peut traiter cet usage comme terminé. Si c'est le dernier usage qui partage le dialogue, celui-ci est aussi terminé.

5.2. Fin de temporisation de transaction

La [RFC3261] déclare qu'un UA devrait terminer un dialogue (par l'envoi d'un BYE) si aucune réponse n'est reçue pour un demande envoyée au sein d'un dialogue. Cette recommandation devrait avoir été limitée à l'usage invite au lieu de tout le dialogue. La [RFC3265] déclare qu'une fin de temporisation pour un NOTIFY supprime un abonnement, mais un SUBSCRIBE qui échoue avec autre chose qu'un 481 ne le fait pas. D'après ces déclarations, il n'est pas clair qu'un rafraîchissement de SUBSCRIBE produit dans un dialogue partagé avec un usage invite détruit l'un et l'autre usage ou le dialogue si il arrive en fin de temporisation.

Généralement, une fin de temporisation de transaction devrait affecter seulement l'usage dans lequel la transaction s'est produite. Les autres usages partageant le dialogue ne devraient pas être affectés. Dans le pire cas de fin de temporisation due à la défaillance totale du transport, il peut falloir plusieurs échecs de messages pour supprimer tous les usages d'un dialogue (au moins un par usage).

Il y a certains messages de mi-dialogue qui n'appartiennent jamais à aucun usage. Si ils arrivent en fin de temporisation, il n'auront pas d'effet sur le dialogue ou ses usages.

5.3 Confrontation des demandes aux usages

Pour de nombreuses demandes de mi-dialogue, identifier l'usage auquel elles appartiennent est évident. Un dialogue peut avoir au plus un usage invite, de sorte que toutes les demandes INVITE, UPDATE, PRACK, ACK, CANCEL, BYE, ou INFO lui appartiennent. L'usage (c'est-à-dire, l'abonnement particulier) auquel appartiennent les demandes SUBSCRIBE, NOTIFY, et REFER peut être déterminé à partir du champ d'en-tête Event de la demande. Les demandes REGISTER au sein d'un (pseudo)-dialogue appartiennent à l'usage d'enregistrement. (Comme mentionné précédemment, les mises en œuvre ne mélangent pas les usages d'enregistrement avec d'autres usages, de sorte que le présent document n'explore pas les conséquences de ce mauvais comportement).

Selon la [RFC3261], "une demande OPTIONS reçue au sein d'un dialogue génère une réponse 200 OK qui est identique à celle construite en dehors d'un dialogue et n'a aucun impact sur ce dialogue". Donc, OPTIONS n'appartient à aucun usage. Seuls les échecs discutés aux paragraphes 5.1 et 5.2 qui détruisent des dialogues entiers vont avoir un effet sur les usages qui partagent le dialogue avec une demande OPTIONS qui échoue.

Les demandes MESSAGE sont déconseillées à l'intérieur d'un dialogue. Il est interdit aux mises en œuvre de créer un usage pour porter une séquence de demandes MESSAGE (bien que certaines mises en œuvre l'utilisent de cette façon, contre la recommandation standard). Un échec de MESSAGE qui se produit au sein d'un dialogue existant va avoir des effets similaires sur le dialogue et ses usages qu'un échec de demande OPTIONS.

Les demandes de mi-dialogue avec des méthodes inconnues ne peuvent pas être confrontées à un usage. Les serveurs vont retourner une réponse d'échec (probablement un 501). L'effet sur le dialogue et ses usages chez le client ou le serveur devrait être similaire à celui d'un échec de demande OPTIONS.

Ces lignes directrices pour confronter les messages aux usages (ou déterminer qu'il n'y a pas d'usage) s'appliquent également à l'action d'un UAS, d'un UAC, ou de tout tiers retraçant l'état d'usage et de dialogue en inspectant tous les messages entre deux points d'extrémité.

5.4 Demandes de rafraîchissement de cible

Les demandes de rafraîchissement de cible mettent à jour la cible distante d'un dialogue quand elles réussissent à être traitées. Les demandes de rafraîchissement de cible actuellement définies sont INVITE, UPDATE, SUBSCRIBE, NOTIFY, et REFER [RFC3515]).

La cible distante fait partie de l'état de dialogue. Quand une demande de rafraîchissement de cible l'affecte, elle l'affecte pour TOUS les usages qui partagent ce dialogue. Si des usages subscription et invite partagent un dialogue, l'envoi d'un rafraîchissement de SUBSCRIBE avec un contact différent va causer des reINVITE de la part de l'homologue pour aller à ce contact différent.

Un UAS va seulement mettre à jour la cible distante si il envoie une réponse de classe 200 à une demande de rafraîchissement de cible. Un UAC va seulement mettre à jour la cible distante si il reçoit une réponse de classe 200 à une demande de rafraîchissement de cible. Là encore, toute mise à jour de la cible distante d'un dialogue affecte tous les usages de ce dialogue.

Il y a une ambiguïté connue sur les effets de réponses provisoires sur des cibles distantes qu'une future spécification tentera de clarifier. De plus, parce que la cible distante fait partie de l'état de dialogue, et pas d'un état d'usage, il est ambigu d'avoir des demandes de rafraîchissement de cible en instance sur plusieurs usages dans le même dialogue. Les concepteurs de mises en œuvre devraient examiner avec attention ces conditions.

5.5 Rafraîchissement et terminaison d'usages

Les usages d'abonnement et d'enregistrement expirent au fil du temps et doivent être rafraîchis (avec un rafraîchissement de SUBSCRIBE, par exemple). Cette expiration est celle de l'état d'usage, pas de l'état de dialogue. Si plusieurs abonnements partagent un dialogue, rafraîchir l'un d'eux n'a pas d'effet sur l'expiration des autres.

La termination normale d'un usage n'a pas d'effet sur les autres usages qui partagent un même dialogue. Par exemple, terminer un abonnement avec un NOTIFY/Subscription-State: terminated ne va pas terminer un usage invite qui partage son dialogue. De même, terminer un usage invite avec un BYE ne termine aucun abonnement Event: refer actif établi sur ce dialogue.

5.6 Refus de nouveaux usages

Comme le montre l'examen de l'effet des réponses d'échec, il faut faire attention quand on refuse un nouvel usage à l'intérieur d'un dialogue existant. Choisir le mauvais code de réponse va terminer le dialogue et tous ses usages. Généralement, retourner un 603 Refus est la façon la plus sûre de refuser un nouvel usage.

5.7 Remplacement d'usages

La [RFC3891] définit un mécanisme par lequel un usage peut en remplacer un autre. Il peut être utilisé, par exemple, pour associer les deux dialogues dans lesquels une cible de transfert est impliquée durant un transfert attendu. Il est écrit en utilisant le terme "dialog", mais son intention est seulement d'affecter l'usage invite du dialogue qu'il cible. Aucun autre usage dans ce dialogue n'est affecté. Pour certaines applications, les autres usages peuvent ne plus avoir de sens, et l'application peut aussi les terminer.

Cependant, les interactions entre les Replace et les usages multiples de dialogue n'ont pas été bien explorées. Une discussion de ce sujet est nécessaire. Les mises en œuvre devraient éviter complètement ce scénario.

6. Évitement de multiples usages

Le traitement correct d'usages multiples n'est pas complètement compris. Ce qui est compris est difficile à mettre en œuvre et va très probablement conduire à des problèmes d'interopérabilité. La meilleure façon d'éviter les ennuis qui découlent d'une telle complexité est de l'éviter complètement.

Quand on conçoit de nouvelles applications ou caractéristiques qui utilisent des dialogues SIP, on ne doit pas exiger des points d'extrémité qu'il construisent plusieurs usages pour participer à l'application ou utiliser la caractéristique. Quand on conçoit des points d'extrémité, traiter les scénarios existants d'usages multiples au mieux qu'il est possible. En dehors de ces scénarios, si un homologue tente de créer un second usage à l'intérieur d'un dialogue, refusez le.

Malheureusement, il y a des applications existantes, comme le transfert, qui entraînent actuellement des usages multiples, de sorte que la plus simple solution de "ne pas le faire" va exiger des travaux de transition. La présente Section examine les pressions qui conduisent aux usages multiples existants et suggère des solutions de remplacement.

Quand ils exécutent un transfert, le transféreur et le transféré partagent actuellement entre eux un usage invite et un usage subscription au sein du dialogue. C'est le résultat de l'envoi de la demande REFER au sein du dialogue établi par l'usage invite. Les mises en œuvre ont été conduites à ce comportement par les problèmes suivants :

1. Il n'y avait pas de moyen de s'assurer qu'un REFER sur un nouveau dialogue atteindrait le point d'extrémité particulier impliqué dans un transfert. De nombreux facteurs, incluant des détails de mise en œuvre et des changements d'acheminement de mandataire entre un INVITE et un REFER pourraient causer l'envoi du REFER au mauvais endroit. Envoyer le REFER sur le dialogue existant assurait qu'il irait au même point d'extrémité que celui avec lequel le dialogue a été établi.
2. La façon d'associer un usage invite existant avec un REFER arrivant sur un nouveau dialogue n'était pas claire, alors qu'il est complètement évident que l'association a été faite quand le REFER est venu sur le dialogue de l'usage invite.
3. Il y a des problèmes d'autorisation des REFER hors d'un dialogue. La politique d'autorisation pour REFER est dans la plupart des mises en œuvre portée sur la politique d'autorisation pour INVITE (qui est, dans la plupart des cas, fondée simplement sur "J'ai passé, ou répondu à, cet appel").

Les URI d'agent d'utilisateur mondialement acheminable (GRUU, *Globally Routable User Agent*) [RFC5627] ont été définis spécifiquement pour traiter le problème 1 en fournissant un URI qui va atteindre un agent d'utilisateur spécifique. Le champ d'en-tête Target-Dialog [RFC4538] a été créé pour traiter les problèmes 2 et 3. Ce champ d'en-tête permet à une demande d'indiquer les identifiants de dialogue d'un autre dialogue, fournissant une association avec l'autre dialogue qui peut être utilisée dans une décision d'autorisation.

Les mécanismes Join [RFC3911] et Replaces [RFC3891] peuvent aussi être utilisés pour traiter le problème 1. Quand on utilise cette technique, une nouvelle demande est envoyée en dehors de tout dialogue dans l'espoir qu'elle va fourcher sur éventuellement de nombreux points d'extrémité, incluant celui qui nous intéresse. Cette demande contient un champ d'en-tête qui fait la liste des identifiants de dialogue d'un dialogue en cours. Seul le point d'extrémité qui détient un dialogue correspondant à ces identifiants va accepter la demande. Les autres points d'extrémité sur lesquels la demande peut avoir fourché vont répondre avec une erreur. Ce mécanisme est raisonnablement robuste, échouant seulement quand la logique d'acheminement pour les demandes hors dialogue change de telle façon que la nouvelle demande n'arrive pas au point d'extrémité qui détient le dialogue concerné.

Les aspects d'accessibilité de l'utilisation d'un GRUU pour traiter le problème 1 peuvent être combinés avec l'association à d'autres aspects de dialogue des mécanismes Join/Replaces et Target-Dialog. Une demande REFER envoyée hors dialogue peut être envoyée vers un GRUU, et identifier un dialogue existant au titre du contexte que le receveur devrait utiliser. Le champ d'en-tête Target-Dialog peut être inclus dans le REFER qui fait la liste des dialogues auxquels ce REFER est associé. La Figure 5 montre comment cela pourrait être utilisé pour réaliser un transfert sans réutiliser un dialogue. Pour simplifier, le diagramme et les détails de message ne montrent pas le serveur à exemple.com qui serait impliqué dans l'acheminement du GRUU. Voir ces détails dans la [RFC5627].

Alice	Bob	Carol
F1 INVITE (AOR de Bob)		
Call-ID: (call-id un)		
Contact: (GRUU d'Alice)		
----->		
F2 200 OK		
To: <>;tag=totag1		
From: <>;tag=fromtag1		
Call-ID: (call-id un)		
Contact: (GRUU de Bob)		
<-----		
ACK		
----->		
:		
(Bob met Alice en garde)		
:		
	F3 INVITE (AOR de Carol)	
	Call-ID: (call-id deux)	
	Contact: (GRUU de Bob)	
	----->	
	F4 200 OK	
	To: <>;tag=totag2	
	From: <>;tag=fromtag2	
	Call-ID: (call-id deux)	
	Contact: (GRUU de Carol)	
	<-----	
	ACK	
	----->	
	:	
	(Bob met Carol en garde)	
	:	
F5 REFER (GRUU d'Alice)		
Call-ID: (call-id trois)		
Refer-To: (GRUU de Carol)		
Target-Dialog: (call-id un, totag1, fromtag1)		
Contact: (GRUU de Bob)		
<-----		
202 Accepté		

```

|----->|
| NOTIFY (GRUU de Bob) |
| Call-ID: (call-id trois) |
|----->|
| 200 OK |
|<-----|
|
| F6 INVITE (GRUU de Carol)
| Call-ID: (call-id quatre)
| Contact: (GRUU d'Alice)
|----->|
| 200 OK
| Contact: (GRUU de Carol)
|<-----|
| ACK
|----->|
|
| F7 NOTIFY (GRUU de Bob)
| Call-ID: (call-id trois)
|----->|
| 200 OK
|<-----|
| BYE (GRUU d'Alice)
| Call-ID: (call-id un)
|<-----| BYE (GRUU de Carol)
| Call-ID: (call-id deux)
| 200 OK |----->|
|----->| 200 OK
|<-----|
|

```

Figure 5 : Transfert sans réutilisation de dialogue

Dans le message F1, Alice invite Bob à indiquer la prise en charge des GRUU (et offre un GRUU pour elle-même) :

Message F1 (abrégé, détaille les champs pertinents)

```

INVITE sip:bob@exemple.com SIP/2.0
Call-ID: 13jfdwer230jsdw@alice.exemple.com
Supported: gruu
Contact: <sip:alice@exemple.com;gr=urn:uuid:(Alice's UA's bits)>

```

Message F2 portant le GRUU de Bob à Alice.

Message F2 (abrégé, détaille les champs pertinents)

```

SIP/2.0 200 OK
Supported: gruu
To: <sip:bob@exemple.com>;tag=totag1
From: <sip:alice@exemple.com>;tag=fromtag1
Contact: <sip:bob@exemple.com;gr=urn:uuid:(Bob's UA's bits)>

```

Bob décide d'essayer de transférer Alice à Carol, donc il met Alice en garde et envoie un INVITE à Carol. Carol et Bob négocient la prise en charge de GRUU semblable à ce qui est arrivé dans F1 et F2.

Message F3 (abrégé, détaille les champs pertinents)

```

INVITE sip:carol@exemple.com SIP/2.0
Supported: gruu
Call-ID: 23rasdnfoa39i4jnasdf@bob.exemple.com
Contact: <sip:bob@exemple.com;gr=urn:uuid:(Bob's UA's bits)>

```

Message F4 (abrégé, détaille les champs pertinents)

```
SIP/2.0 200 OK
Supported: gruu
To: <sip:carol@exemple.com>;tag=totag2
From: <sip:bob@exemple.com>;tag=fromtag2
Call-ID: 23rasdnfoa39i4jnasdf@bob.exemple.com
Contact: <sip:carol@exemple.com;gr=urn:uuid:(Carol's UA's bits)>
```

Après avoir consulté Carol, Bob la met en garde et réfère Alice à elle en utilisant le message F5. Remarquer que l'URI de Refer-To URI est le GRUU de Carol, et qu'il est sur un Call-ID différent du message F1. (l'URI dans l'en-tête Refer-To est renvoyé à la ligne pour la lisibilité dans ce document ; il ne serait pas valide de couper l'URI de cette façon dans un message réel.)

Message F5 (abrégé, détaille les champs pertinents)

```
REFER sip:aanewmr203raswdf@exemple.com SIP/2.0
Call-ID: 39fa99r0329493asdsf3n@bob.exemple.com
Refer-To: <sip:carol@exemple.com;g=urn:uuid:(bits de l'UA de Carol)
?Replaces=23rasdnfoa39i4jnasdf@bob.exemple.com;
to-tag=totag2;from-tag=fromtag2>
Target-Dialog: 13jfdwer230jsdw@alice.exemple.com;
local-tag=fromtag1;remote-tag=totag1
Supported: gruu
Contact: <sip:bob@exemple.com;gr=urn:uuid:(bits de l'UA de Bob)>
```

Alice utilise les informations du champ d'en-tête Target-Dialog pour déterminer que ce REFER est associé au dialogue qu'elle a déjà en place avec Bob. Alice est maintenant en position d'utiliser la même politique d'admission qu'elle a utilisée pour les REFER dans le dialogue : "Ai-je un appel avec cette personne ?". Elle accepte le REFER, envoie à Bob le NOTIFY obligatoire immédiat, et procède à l'INVITE à Carol avec le message F6.

Message F6 (abrégé, détaille les champs pertinents)

```

sip:carol@exemple.com;gr=urn:uuid:(bits de l'UA de Carol)
  \
   \
    v
INVITE
Call-ID: 4zsd9f234jasdfasn3jsad@alice.exemple.com
Replaces: 23rasdnfoa39i4jnasdf@bob.exemple.com;
to-tag=totag2;from-tag=fromtag2
Supported: gruu
Contact: <sip:alice@exemple.com;gr=urn:uuid:(Alice's UA's bits)>

```

Carol accepte l'invitation de Alice de remplacer son dialogue (usage invite) avec Bob, et lui notifie que le INVITE référencé a réussi avec F7:

Message F7 (abrégé, détaille les champs pertinents)

```
NOTIFY sip:boaiidfjereis@exemple.com SIP/2.0
Subscription-State: terminé ; raison = pas de ressource
Call-ID: 39fa99r0329493asdsf3n@bob.exemple.com
Contact: <sip:alice@exemple.com;gr=urn:uuid:(bits de l'UA d'Alice)>
Content-Type: message/sipfrag
SIP/2.0 200 OK
```

Bob termine alors ses usages invite avec Alice et Carol en utilisant des BYE.

7. Considérations sur la sécurité

Le traitement d'usages multiples au sein d'un seul dialogue est complexe et introduit des scénarios où la bonne chose à faire n'est pas claire. Les ambiguïtés décrites ici peuvent résulter en l'interruption inattendue de la communication si les codes de réponse ne sont pas choisis avec soin. De plus, ces ambiguïtés pourraient être exploitées, en particulier par des tiers qui injecteraient des demandes non authentifiées ou des réponses inappropriées. Les mises en œuvre qui choisissent de créer ou d'accepter des usages multiples au sein d'un dialogue devraient porter une extrême attention aux considérations de sécurité de la [RFC3261], en particulier celles concernant l'authenticité des demandes et le traitement des réponses.

Les mises en œuvre de services devraient considérer avec attention les effets sur leur service d'homologues qui font des choix différents dans ces zones d'ambiguïté. Un service qui exige de multiples usages doit prêter une attention particulière aux effets sur le service et l'utilisation du réseau quand un client échoue à détruire un dialogue dont le service estime qu'il devrait être supprimé. Un service qui interdit les usages multiples devrait considérer les effets sur les clients qui, par exemple, suppriment le dialogue entier quand seulement un usage devrait être supprimé. Dans le pire cas d'un service déployé dans un réseau avec un grand nombre de clients qui se conduisent mal en essayant de créer de multiples usages de façon automatique, une tempête d'essais similaire à une avalanche de redémarrages pourrait être produite.

8. Conclusion

Le traitement d'usages multiples au sein d'un seul dialogue est complexe et introduit des scénarios où la bonne chose à faire n'est pas claire. Les mises en œuvre devraient éviter d'entrer dans des usages multiples chaque fois que possible. Les nouvelles applications devraient être conçues pour ne jamais introduire d'usages multiples.

Il y a certaines pratiques acceptées dans SIP, incluant le transfert, qui exigent actuellement des usages multiples. Des travaux récents, notamment sur GRUU, rendent ces pratiques inutiles. La normalisation de ces pratiques et leurs mises en œuvre devraient être révisées aussitôt que possible pour utiliser seulement le dialogue à un seul usage.

9. Remerciements

Les idées contenues dans le présent document ont été affinées sur plusieurs réunions de l'IETF rassemblant de nombreux participants. Des contributions significatives ont été fournies par Adam Roach, Alan Johnston, Ben Campbell, Cullen Jennings, Jonathan Rosenberg, Paul Kyzivat et Rohan Mahy. Les membres du projet reSIPprocate ont aussi partagé leurs difficultés et découvertes lors de ma mise en œuvre des traitements de dialogue multi usages.

10. Références pour information

- [RFC3261] J. Rosenberg et autres, "SIP : [Protocole d'initialisation de session](#)", juin 2002. (*Mise à jour par* [3265](#), [3853](#), [4320](#), [4916](#), [5393](#), [6665](#), [8217](#), [8760](#))
- [RFC3263] J. Rosenberg, H. Schulzrinne, "Protocole d'initialisation de session (SIP) : [Localisation des serveurs SIP](#)", juin 2002. (*Remplace* [RFC2543](#)) (*P.S.* ; *MàJ par* [RFC7984](#), [RFC8898](#))
- [RFC3265] A.B. Roach, "[Notification d'événement spécifique](#) du protocole d'initialisation de session (SIP)", juin 2002. (*MàJ par* [RFC6446](#)) (*Remplacée par la* [RFC6665](#))
- [RFC3515] R. Sparks, "[Méthode Refer](#) du protocole d'initialisation de session (SIP)", avril 2003. (*MàJ par* [RFC8217](#))
- [RFC3891] R. Mahy, B. Biggs, R. Dean, "[En-tête "Replaces"](#) du protocole d'initialisation de session (SIP)", septembre 2004. (*P.S.*)
- [RFC3903] A. Niemi, "[Extension au protocole d'initialisation de session](#) (SIP) pour la publication d'état d'événement", octobre 2004.
- [RFC3911] R. Mahy, D. Petrie, "[En-tête "Join" du protocole](#) d'initialisation de session (SIP)", octobre 2004. (*P.S.*)

- [RFC4488] O. Levin, "[Suppression de l'abonnement implicite](#) de la méthode REFER du protocole d'initialisation de session (SIP)", mai 2006. (P.S.)
- [RFC4538] J. Rosenberg, "[Autorisation de demande par identification](#) de dialogue dans le protocole d'initialisation de session (SIP)", juin 2006. (P.S.)
- [RFC4730] E. Burger, M. Dolly, "[Paquetage d'événement du protocole d'initialisation de session](#) (SIP) pour stimulus par langage de balisage à pression de touche (KPML)", novembre 2006. (P.S.)
- [RFC5627] J. Rosenberg, "[Obtention et utilisation des URI](#) d'agent d'utilisateur mondialement acheminable (GRUU) dans le protocole d'initialisation de session (SIP)", octobre 2009. (P.S.)

Adresse de l'auteur

Robert J. Sparks
Estacado Systems
mél : RjS@estacado.net

Déclaration complète de droits de reproduction

Copyright (C) The Internet Society (2007)

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 contenues sont fournis 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, le 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 encloses ne viole 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 le 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.