Équipe d'ingénierie de l'Internet (IETF)
Request for Comments : 7423
BCP 193
Catégorie : Bonnes pratiques actuelles
ISSN : 2070-1721

L. Morand, Ed., Orange Labs
V. Fajardo, Fluke Networks
H. Tschofenig

novembre 2014
Traduction Claude Brière de L'Isle

Directives de conception pour les applications Diameter

Résumé
Le protocole de base Diameter fournit des facilités d'extensibilité de protocole qui permettrent la définition de nouvelles applications Diameter ou la modification d'applications existantes. Le présent document est un document d'accompagnement du protocole Diameter de base qui explique et précise les règles pour étendre Diameter. De plus, le présent document donne des lignes directrices pour les concepteurs d'applications Diameter qui réutilisent/définissent des applications Diameter ou créent des extensions génériques à Diameter.

Statut de ce mémoire
Ceci est un document des bonnes pratiques actuelles de l'Internet.

Le présent document a été produit par l'équipe d'ingénierie de l'Internet (IETF). Il représente le consensus de la communauté de l'IETF. Il a subi une révision publique et sa publication a été approuvée par le groupe de pilotage de l'ingénierie de l'Internet (IESG). Plus d'informations sur les normes de l'Internet sont disponibles à la Section 2 de la [RFC5741].

Les informations sur le statut actuel du présent document, tout errata, et comment fournir des réactions sur lui peuvent être obtenues à http://www.rfc-editor.org/info/rfc7423

Table des Matières

1. Introduction

Le protocole de base Diameter [RFC6733] est destiné à fournir un cadre d'authentification, autorisation et comptabilité (AAA, *Authentication, Authorization, and Accounting*) pour des applications comme l'accès réseau ou la mobilité IP dans des situations locales et d'itinérance. Ce protocole donne la capacité aux homologues Diameter d'échanger des messages portant des données sous la forme de paires d'attribut-valeur (AVP, *Attribute-Value Pair*).

Le protocole de base Diameter fournit des facilités pour étendre Diameter (voir le paragraphe 1.3 de la [RFC6733]) pour prendre en charge de nouvelles fonctionnalités. Dans le contexte du présent document, étendre Diameter signifie une des choses suivantes :

1. L'ajout d'une nouvelle fonctionnalité à une application Diameter existante sans définir une nouvelle application.
2. L'ajout d'une nouvelle fonctionnalité à une application Diameter existante qui exige la définition d'une nouvelle application.
3. La définition d'une application Diameter entièrement nouvelle pour offrir une fonctionnalité non prise en charge par les applications existantes.
4. La définition d'une nouvelle fonctionnalité générique qui peut être réutilisée sur différentes applications.

Toutes ces extensions sont des décisions de conception qui peuvent être réalisées par toute combinaison de réutilisation de commandes existantes ou de définition de nouvelles commandes, AVP, ou valeurs d'AVP. Cependant, les concepteurs d'application n'ont pas une liberté complète pour cela. Un certain nombre de règles ont été définies dans la [RFC6733] qui mettent des contraintes sur quand une extension exige l'allocation d'un nouvel identifiant d'application Diameter ou une nouvelle valeur de code de commande. L'objectif du présent document est le suivant :

- Préciser les règles d'extensibilité de Diameter comme définies dans le protocole de base Diameter.
- Discuter des choix de conception et fournir des directives pour la définition de nouvelles applications.
- Présenter des choix de compromis.

2. Terminologie

Le présent document réutilise la terminologie définie dans la [RFC6733]. De plus, les termes suivants sont utilisés ici :

Application : Extension du protocole de base Diameter [RFC6733] via l'ajout de nouvelles commandes ou AVP. Chaque application est identifiée de façon univoque par une valeur d'identifiant d'application allouée par l'IANA.

Commande : demande ou réponse Diameter portant des AVP entre des points d'extrémité Diameter. Chaque commande est identifiée de façon univoque par une valeur de code de commande allouée par l'IANA et est décrite par un format de code de commande (CCF, *Command Code Format*) pour une application.

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

3. Vue d'ensemble

Comme il est conçu, le protocole de base Diameter [RFC6733] peut être vu comme un protocole à deux couches. La couche inférieure est principalement chargée de gérer les connexions entre les homologues voisins et l'acheminement de messages. La couche supérieure est celle où résident les applications Diameter. Ce modèle est en ligne avec un nœud Diameter ayant une couche d'application et une couche de livraison d'homologue à homologue. Le document du protocole de base Diameter définit l'architecture et le comportement de la couche de livraison de message et fournit ensuite le cadre pour concevoir les applications Diameter sur la couche d'application. Ce cadre inclut les définitions des sessions d'application et de prise en charge de la comptabilité (voir les Sections 8 et 9 de la [RFC6733]). En conséquence, un nœud Diameter est vu dans le présent document comme une seule instance de couche de livraison de message Diameter et une ou plusieurs applications Diameter qui l'utilisent.

Le protocole de base Diameter est conçu pour être extensible et les principes sont décrits au paragraphe 1.3 de la [RFC6733]. En résumé, Diameter peut être étendu par :

1. la définition de nouvelles valeurs d'AVP
2. la création de nouvelles AVP
3. la création de nouvelles commandes
4. la création de nouvelles applications.

Comme principe directeur principal, les concepteurs d'application DEVRAIENT se conformer à la recommandation suivante : "essayer de réutiliser autant que possible !" Cela va réduire le temps nécessaire pour finaliser la rédaction des spécifications, et cela va conduire à un moindre effort de mise en œuvre ainsi qu'à réduire les besoins d'essais. En général, il est habile d'éviter de dupliquer les efforts lorsque possible.

Cependant, la réutilisation n'est pas appropriée lorsque la fonctionnalité existante ne correspond pas aux nouvelles exigences et/ou lorsque la réutilisation conduit à des ambiguïtés.

L'impact de l'extension d'applications existantes peut être catégorisé en deux groupes :

Extension mineure : l'amélioration de la portée fonctionnelle d'une application existante par l'ajout de caractéristiques facultatives pour la prendre en charge. De telles améliorations n'ont pas de problème de rétro compatibilité avec l'application existante. Un exemple typique serait la définition d'une nouvelle AVP facultative à utiliser dans une commande existante. Les mises en œuvre Diameter qui prennent en charge l'application existante mais pas la nouvelle AVP vont simplement l'ignorer, sans conséquence pour le traitement de message Diameter, comme décrit dans la [RFC6733]. L'effort de normalisation sera très faible.

Extension majeure : l'amélioration d'une application qui requiert la définition d'une nouvelle application Diameter. Une telle amélioration cause un problème de rétro compatibilité avec les mises en œuvre existantes qui prennent en charge l'application. Des exemples typiques seraient la création d'une nouvelle commande pour fournir une fonctionnalité non prise en charge par les applications existantes ou la définition d'une nouvelle AVP à porter dans une commande existante avec le bit M établi dans les fanions d'AVP (voir au paragraphe 4.1 de la [RFC6733] pour la définition du "bit M"). Pour une telle extension, un effort de spécification significatif est requis, et une approche prudente est recommandée.

4. Réutilisation d'applications Diameter existantes

Une application existante peut devoir être améliorée pour satisfaire à de nouvelles exigences, et ces modifications peuvent être au niveau de la commande et/ou au niveau de l'AVP. Les paragraphes qui suivent décrivent les possibles modifications à effectuer sur les applications existantes et leur impact.

4.1 Ajout d'une nouvelle commande

L'ajout d'une nouvelle commande à une application existante est considéré comme une extension majeure et exige qu'une nouvelle application Diameter soit définie, comme le déclare le paragraphe 1.3.4 de la [RFC6733]. Le besoin d'une nouvelle application est parce que un nœud Diameter qui n'est pas mis à niveau pour prendre en charge la ou les nouvelles commandes au sein de l'application (existante) va rejeter toute commande inconnue avec l'erreur de protocole DIAMETER_COMMAND_UNSUPPORTED et causer l'échec de la transaction. La nouvelle application assure que les nœuds Diameter ne reçoivent des commandes que dans le contexte d'applications qu'ils prennent en charge.

Ajouter une nouvelle commande signifie soit de définir une commande complètement nouvelle, soit d'importer la syntaxe de

format de code de commande de la commande d'une autre application par laquelle la nouvelle application hérite de tout ou partie des fonctionnalités de l'application d'où vient la commande. Dans le premier cas, la décision de créer une nouvelle application est directe, car c'est typiquement un résultat de l'ajout d'une nouvelle fonctionnalité qui n'existait pas encore. Pour le second cas, la décision de créer une nouvelle application va dépendre de si l'importation de la commande dans une nouvelle application convient mieux que de simplement utiliser l'application existante comme elle est, en conjonction avec toute autre application.

On considère par exemple l'application du protocole extensible d'authentification Diameter (EAP, *Extensible Authentication Protocol*) [RFC4072] et l'application de serveur d'accès réseau Diameter [RFC7155]. Lorsque l'authentification d'accès réseau utilisant EAP est requise, les commandes Diameter EAP (Diameter-EAP-Request/Diameter-EAP-Answer) sont utilisées ; autrement, l'application de serveur d'accès réseau Diameter sera utilisée. Quand l'application Diameter EAP est utilisée, les échanges de comptabilité définis dans le serveur d'accès réseau Diameter peuvent être utilisés.

Cependant, en général, il est difficile d'arriver à des directives précises, et donc une étude au cas par cas de chaque exigence d'application devrait être appliquée. Avant d'ajouter ou d'importer une commande, les concepteurs d'application devraient prendre en compte ce qui suit :

- la nouvelle fonctionnalité peut elle être satisfaite en créant une nouvelle commande indépendante de toute commande existante ? Dans ce cas, la nouvelle application résultante et l'application existante peuvent elles fonctionner indépendamment l'une de l'autre, mais en coopération ?
- La commande existante peut elle être réutilisée sans extension majeure et donc, sans qu'il soit besoin de définir une nouvelle application, par exemple, une nouvelle fonctionnalité introduite par la création de nouvelles AVP facultatives ?

Il est important de noter qu'importer des commandes de façon trop libérale pourrait résulter en une application monolithique et difficile à gérer prenant en charge trop de caractéristiques différentes.

4.2 Suppression d'une commande existante

Bien que ce processus ne soit pas courant, la suppression d'une commande d'une application exige qu'une nouvelle application Diameter soit définie, et ensuite elle est considérée comme une extension majeure. C'est dû au fait que la réception de la commande supprimée va systématiquement résulter en une erreur de protocole (c'est-à-dire, "commande Diameter non prise en charge").

Il est inhabituel de supprimer une commande existante d'une application juste pour le plaisir de la supprimer, elle ou la fonction qu'elle représente. Une exception pourrait être si l'intention de la suppression est de créer une nouvelle variante de la même application qui serait plus simple que l'application spécifiée initialement.

4.3 Réutilisation d'une commande existante

Ce paragraphe discute les règles pour l'ajout et/ou la suppression d'AVP d'une commande existante dans une application existante. Les cas décrits dans ce paragraphe peuvent ne pas nécessairement résulter en la création de nouvelles applications.

D'un point de vue historique , il vaut de noter qu'il y avait une forte recommandation en faveur de la réutilisation des commandes existantes dans la [RFC3588] pour prévenir l'épuisement rapide des valeurs de code disponibles pour les commandes spécifiques de fabricants. Cependant, la [RFC6733] a relâché la politique d'allocation et élargi la gamme des valeurs de code disponibles pour les applications spécifiques de fabricants. Bien que la réutilisation de commandes existantes soit toujours RECOMMANDÉE, les concepteurs de protocoles peuvent envisager de définir une nouvelle commande quand elle apporte une solution plus convenable que de torturer l'utilisation et l'application des commandes existantes.

4.3.1 Ajout d'AVP à une commande

Sur la base des règles de la [RFC6733], les AVP qui sont ajoutées à une commande existante peuvent être catégorisées :

- Comme des AVP obligatoires (de prise en charge). Comme défini dans la [RFC6733], ce sont des AVP avec le fanion du bit M établi (à 1) dans cette commande, ce qui signifie que le nœud Diameter qui les reçoit est obligé de comprendre non seulement leur valeur mais aussi leur sémantique. Manquer à le faire cause une erreur de traitement de message : soit un message d'erreur avec le code de résultat réglé à "AVP Diameter non acceptée" si l'AVP n'est pas comprise dans une demande, soit un traitement d'erreur spécifique d'application si l'AVP en question est dans une réponse.

- Comme des AVP facultatives (de prise en charge). Comme défini dans la [RFC6733], ce sont des AVP avec le fanion du bit M à zéro dans cette commande. Un nœud Diameter qui reçoit ces AVP peut simplement les ignorer si il ne les prend pas en charge.

Il est important de noter que les définitions données ci-dessus sont indépendantes du caractère obligatoire ou facultatif de ces AVP dans la commande comme spécifié par la syntaxe de CCF de la commande [RFC6733].

Note : Comme le déclare la [RFC6733], le réglage du bit M pour une certaine AVP relève d'une application et chaque commande au sein de cette application qui comporte l'AVP.

Les règles sont strictes dans le cas où les AVP à ajouter dans une commande existante sont de prise en charge obligatoire, c'est-à-dire, elles ont le bit M établi. Une AVP obligatoire NE DOIT PAS être ajoutée à une commande existante sans définir une nouvelle application Diameter, comme déclaré dans la [RFC6733]. Cela entre dans la catégorie des "extensions majeures". En dépit de la clarté de la règle, il reste une ambiguïté lors de l'évaluation de si une nouvelle AVP ajoutée devrait être obligatoire. Un concepteur d'application devrait examiner les questions suivantes pour décider du sort du bit M dans une nouvelle AVP :

- Sera t-il exigé que le côté receveur soit capable de traiter et comprendre l'AVP et son contenu ?
- Les nouvelles AVP vont elles changer l'automate à états de l'application ?
- La présence de la nouvelle AVP conduit elle à un nombre différent d'allers-retours, changeant effectivement l'automate à états de l'application ?
- La nouvelle AVP sera t-elle utilisée pour différencier les variantes anciennes et nouvelle de la même application lorsque les deux variantes ne sont pas rétro compatibles ?
- La nouvelle AVP aura t-elle une double signification, c'est-à-dire, sera t-elle utilisée pour porter des informations relatives à l'application ainsi que pour indiquer que le message est pour une nouvelle application ?

Si la réponse à la dernière question est "oui", le bit M DOIT alors être établi pour la nouvelle AVP, et une nouvelle application Diameter DOIT être définie. Cette liste de questions n'est pas exhaustive, et d'autres critères PEUVENT être pris en compte dans le processus de décision.

Si les concepteurs d'application envisagent plutôt l'utilisation d'AVP facultatives, c'est-à-dire, avec le bit M à zéro, il y a encore des pièges qui peuvent causer des problèmes d'interopérabilité ; donc, ils doivent être évités. Des exemples de ces pièges sont :

- L'utilisation d'AVP facultatives avec des significations qui se recoupent. Une AVP a partiellement le même usage et la même signification qu'une autre AVP. La présence des deux peut introduire une certaine confusion.
- Des AVP facultatives avec un double objet, c'est-à-dire, porter des données d'application ainsi que indiquer la prise en charge d'une ou plusieurs caractéristiques. Cela a tendance à introduire des problèmes d'interprétation.
- L'ajout d'une ou plusieurs AVP facultatives et l'indication (généralement au sein d'un texte descriptif pour la commande) que au moins une d'elles doit être comprise par le receveur de la commande. Cela serait équivalent à ajouter une AVP obligatoire, c'est-à-dire, une AVP avec le bit M établi, à la commande.

4.3.2 Suppression d'AVP d'une commande

Les concepteurs d'applications peuvent vouloir la réutilisation d'une commande existante, mais certaines des AVP présentes dans la spécification de syntaxe de CCF de la commande peuvent n'être pas pertinentes pour la fonction qu'il est prévu de prendre en charge par cette commande. Il peut alors être tentant de supprimer ces AVP de la commande.

Les impacts de la suppression d'une AVP d'une commande dépendent de sa spécification de format de code de commande et du réglage du bit M :

- Cas 1 : Suppression d'une AVP qui est indiquée comme AVP obligatoire (notée par{AVP}) dans la spécification de syntaxe de CCF de la commande (sans considération du réglage du bit M). Dans ce cas, un nouveau code de commande, et par suite, une nouvelle application Diameter, DOIT être spécifié.

- Cas 2 : Suppression d'une AVP qui a le bit M établi, et est indiquée comme AVP facultative (notée [AVP] dans le CCF de commande) dans la spécification de syntaxe de CCF de la commande. Dans ce cas, aucun nouveau code de commande n'a à être spécifié, mais la définition d'une nouvelle application Diameter est REQUISE.

- Cas 3 : Suppression d'une AVP, qui a le bit M à zéro, et est indiquée comme [AVP] dans la spécification de syntaxe de CCF de la commande. Dans ce cas, l'AVP peut être supprimée sans conséquences.

Les concepteurs d'applications DEVRAIENT tenter la réutilisation de la spécification de syntaxe de CCF de la commande sans modification et simplement ignorer (mais pas supprimer) toute AVP facultative qui ne sera pas utilisée. C'est pour conserver la compatibilité avec les applications existantes qui ne vont pas connaître la nouvelle fonctionnalité ainsi que pour conserver l'intégrité des dictionnaires existants.

4.3.3 Changement des réglages de fanions d'AVP dans les commandes existantes

Bien que ce soit inhabituel, les développeurs peuvent vouloir changer le réglage des fanions d'AVP d'une AVP utilisée dans une commande.

Dans une commande existante, une AVP qui était initialement définie comme AVP dont la compréhension est obligatoire, c'est-à-dire, une AVP avec le fanion bit M établi dans la commande PEUT être en toute sécurité transformée en une AVP facultative, c'est-à-dire, avec le bit M à zéro. Tout nœud qui prend en charge l'application existante va quand même comprendre l'AVP, quel que soit le réglage du bit M. Au contraire, une AVP initialement définie comme AVP facultative à comprendre, c'est-à-dire, une AVP avec le fanion bit M à zéro dans la commande NE DOIT PAS être changée en une AVP obligatoire avec le fanion bit M établi sans définir une nouvelle application Diameter. Établir le bit M pour une AVP qui était définie comme AVP facultative est équivalent à ajouter une nouvelle AVP obligatoire à une commande existante, et les règles données au paragraphe 4.3.1 s'appliquent.

Tous les autres fanions d'AVP (bit V, bit P, bits réservés) DOIVENT rester inchangés.

4.4 Réutilisation d'AVP existantes

Ce paragraphe discute des règles de réutilisation des AVP existantes lors de la réutilisation d'une commande existante ou de la définition d'une nouvelle commande dans une nouvelle application.

4.4.1 Réglage des fanions d'AVP

Lorsque ils réutilisent des AVP existantes dans une nouvelle application, les concepteurs d'application DOIVENT spécifier le réglage du fanion bit M pour une nouvelle application Diameter et, si nécessaire, pour chaque commande de l'application qui peut porter ces AVP. En général, pour les AVP définies en dehors du protocole de base Diameter, les caractéristiques d'une AVP sont liées à son rôle au sein d'une certaine application et aux commandes utilisées dans cette application.

Tous les autres fanions d'AVP (bit V, bit P, bits réservés) DOIVENT rester inchangés.

4.4.2 Réutilisation d'AVP de type Enumerated

Lorsque on réutilise une AVP de type Enumerated dans une commande pour une nouvelle application, il est RECOMMANDÉ d'éviter de modifier l'ensemble des valeurs valides définies pour cette AVP. Modifier l'ensemble des valeurs énumérées inclut d'ajouter une valeur ou de déconseiller l'utilisation d'une valeur définie initialement pour l'AVP. Modifier l'ensemble de valeurs va impacter l'application qui définit cette AVP et toutes les applications qui l'utilisent, causant de potentiels problèmes d'interopérabilité : une valeur utilisée par un homologue qui ne sera pas reconnue par tous les nœuds entre le client et le serveur va causer une réponse d'erreur avec l'AVP Result-Code réglée à DIAMETER_INVALID_AVP_VALUE. Lorsque la pleine gamme des valeurs définies pour cette AVP Enumerated ne convient pas pour la nouvelle application, il est RECOMMANDÉ qu'une nouvelle AVP soit définie pour éviter des problèmes de rétro compatibilité avec les mises en œuvre existantes.

5. Définition de nouvelles applications Diameter

5.1 Introduction

Cette section discute du cas où de nouvelles applications ont des exigences qui ne peuvent pas être satisfaites par les applications existantes et exigeraient une définition de commandes, d'AVP et/ou de valeurs d'AVP, complètement nouvelles. Normalement, il y a peu d'ambiguïté sur la décision de créer ces types d'applications. Des exemples sont les interfaces définies pour le sous système multimédia IP du 3GPP, par exemple, Cx/Dx ([TS29.228] et [TS29.229]), Sh ([TS29.328] et [TS29.329]), etc.

Les concepteurs d'applications DEVRAIENT essayer d'importer les AVP existantes et les valeurs d'AVP pour toute nouvelles commandes définies. Dans certains cas où la comptabilité va être utilisée, les modèles décrits au paragraphe 5.10 DEVRAIENT aussi être considérés.

Les considérations supplémentaires sont décrites dans les paragraphes suivants.

5.2 Définition de nouvelles commandes

Comme recommandation générale, les commandes NE DEVRAIENT PAS être définies à partir de rien. Il est plutôt RECOMMANDÉ de réutiliser une commande existante offrant des fonctions similaires et de l'utiliser comme point de départ.

La réutilisation de code conduit à un plus petit effort de mise en œuvre et réduit les besoins d'essais.

De plus, la spécification de syntaxe de CCF de la nouvelle commande DEVRAIT être définie avec soin lorsque on considère l'applicabilité et l'extensibilité de l'application. Si la plupart des AVP contenues dans la commande sont indiquées comme fixées ou exigées, il pourrait être difficile de réutiliser la même commande et donc, la même application dans un environnement légèrement changé. Définir une commande dont la plupart des AVP sont indiquées comme facultatives est considéré comme un bon choix de conception dans de nombreux cas, en dépit de la souplesse que cela introduit dans le protocole. Les concepteurs de protocoles DOIVENT déclarer clairement les raisons pour lesquelles ces AVP facultatives pourraient être ou non présentes et définir de façon appropriée le comportement correspondant des nœuds Diameter lorsque ces AVP sont absentes de la commande.

Note : comme conseil aux concepteurs de protocoles, il n'est pas suffisant de juste regarder la spécification de la syntaxe de CCF de la commande. Il est aussi nécessaire de lire attentivement le texte d'accompagnement de la spécification.

De la même façon, la spécification de syntaxe de CCF DEVRAIT être définie de façon telle qu'il soit possible d'ajouter toutes AVP facultatives arbitraires avec le bit M à zéro (incluant les AVP spécifiques de fabricant) sans modifier l'application. À cette fin, "* [AVP]" DEVRAIT être ajouté au CCF de la commande, ce qui permet l'ajout d'un nombre arbitraire d'AVP facultatives, comme décrit dans la [RFC6733].

5.3 Utilisation de l'identifiant d'application dans un message

Lorsque ils conçoivent de nouvelles applications, les concepteurs d'application DEVRAIENT spécifier que l'identifiant d'application porté dans tous les messages de niveau session est l'identifiant d'application de l'application qui utilise ces messages. Cela inclut les messages de niveau session définis dans le protocole de base Diameter, c'est-à-dire, Re-Auth-Request (RAR) / Re-Auth-Answer (RAA), Session-Termination-Request (STR) / Session-Termination-Answer (STA), Abort-Session-Request (ASR) / Abort-Session-Answer (ASA), et éventuellement Accounting-Request (ACR) / Accounting Answer (ACA) dans le modèle de comptabilité couplée ; voir au paragraphe 5.10. Certaines spécifications existantes ne respectent pas cette règle pour des raisons historiques. Cependant, cette directive DEVRAIT être suivie par les nouvelles applications pour éviter des problèmes d'acheminement.

Lorsque une nouvelle application a reçu un nouvel identifiant d'application et qu'elle réutilise aussi des commandes existantes avec ou sans modifications, la commande DEVRAIT utiliser le nouvel identifiant d'application alloué dans l'en-tête et dans toutes les AVP Application-Id pertinentes (Auth-Application-Id ou Acct-Application-Id) présentes dans le corps de message de commande.

De plus, les concepteurs d'application qui utilisent une AVP Application-Id spécifique de fabricant NE DEVRAIENT PAS utiliser l'AVP Vendor-Id pour encore partager ou différencier l'identifiant d'application de spécification de fabricant. L'acheminement Diameter ne se fonde pas sur le Vendor Id. À ce titre, le Vendor Id NE DEVRAIT PAS être utilisé comme entrée supplémentaire pour les messages d'acheminement ou de livraison. L'AVP Vendor-Id est seulement une AVP d'information qui n'est conservée que pour des raisons de rétro compatibilité.

5.4 Automates à état de session spécifiques d'application

La Section 8 de la [RFC6733] donne des automates à états de session pour les services AAA, et ces automates à états de session ne sont pas destinés à couvrir des comportements hors de AAA. Si une nouvelle application ne peut pas être clairement catégorisée dans un de ces services AAA, il est RECOMMANDÉ que l'application définisse son propre automate à états de session. La prise en charge d'une demande initiée par un serveur est un exemple évident où un automate à états de session spécifique d'une application serait nécessaire, par exemple, l'interface Rw pour le modèle poussé de l'UIT-T (cf. [Q.3303.3]).

5.5 AVP Session-Id et gestion de session

Les applications Diameter sont généralement conçues dans le but de gérer des sessions d'utilisateur (par exemple, une application de serveur d'accès réseau (NAS, *Network Access Server*) Diameter [RFC4005]) ou une session d'accès à un service spécifique (par exemple, application SIP Diameter [RFC4740]). Dans le protocole de base Diameter, l'état de session est référencé en utilisant l'AVP Session-Id. Tous les messages Diameter qui utilisent le même Session-Id seront liés à la même session. La gestion de session fondée sur Diameter implique aussi que le client et le serveur Diameter (et potentiellement les agents mandataires le long du chemin) conservent les informations d'état de session.

Cependant, certaines applications peuvent n'avoir pas besoin de s'appuyer sur le Session-Id pour identifier et gérer des sessions parce que d'autres informations peuvent être utilisées à la place pour corréler les messages Diameter. Bien sûr, l'AVP User-Name ou toute autre AVP spécifique peut être présente dans tout message Diameter, et donc utilisée pour la corrélation des

messages. Certaines applications peuvent ne pas exiger du tout la notion de session Diameter. Pour de telles applications, l'AVP Auth-Session-State est généralement réglée à NO_STATE_MAINTAINED dans tous les messages Diameter, et ces applications sont donc conçues comme un ensemble de transactions autonomes. Même si une terminaison explicite de session d'accès est exigée, des commandes spécifiques de l'application sont définies et utilisées à la place du STR/STA ou ASR/ ASA définis dans le protocole de base Diameter [RFC6733]. Dans un tel cas, le Session-Id n'est pas significatif.

Sur la base de ces considérations, les concepteurs de protocoles devraient apprécier attentivement si l'application Diameter qu'ils définissent s'appuie sur la gestion de session spécifiée dans le protocole de base Diameter:

- Si elle l'est, la commande Diameter définie pour la nouvelle application DOIT inclure l'AVP Session-Id définie dans le protocole de base Diameter [RFC6733], et l'AVP Session-Id DOIT être utilisée pour la corrélation des messages relatifs à la même session. Des lignes directrices sur l'utilisation de l'AVP Auth-Session-State sont données dans le protocole de base Diameter [RFC6733].

- Autrement, parce que la gestion de session n'est pas exigée ou parce que l'application s'appuie sur son propre mécanisme de gestion de session, les commandes Diameter pour l'application n'ont pas besoin d'inclure l'AVP Session-Id. Si un concept spécifique de gestion de session est pris en charge par l'application, sa documentation DOIT clairement spécifier comment la session est traitée entre le client et le serveur (et éventuellement les agents Diameter dans le chemin). De plus, comme l'application ne conserve pas l'état de session au niveau du protocole de base Diameter, l'AVP Auth-Session-State DOIT être incluse dans toutes les commandes Diameter pour l'application et DOIT être réglée à NO_STATE_MAINTAINED.

5.6 Utilisation d'AVP de type Enumerated

Le type Enumerated a été initialement défini pour fournir une liste des valeurs valides pour une AVP avec leur interprétation respective décrite dans la spécification. Par exemple, les AVP de type Enumerated peuvent être utilisées pour fournir plus d'informations sur la raison de la terminaison d'une session ou sur une action spécifique à effectuer à réception de la demande.

Comme décrit au paragraphe 4.4.2, définir une AVP de type Enumerated présente certaines limitations en termes d'extensibilité et de réutilisabilité. Bien sûr, l'ensemble fini des valeurs valides mentionné dans la définition de l'AVP de type Enumerated ne peut pas être modifié en pratique sans causer des problèmes de rétro compatibilité avec les mises en œuvre existantes. Par conséquent, les AVP de type Enumerated NE DOIVENT PAS être étendues par l'ajout de nouvelles valeurs pour prendre en charge de nouvelles capacités. Les concepteurs de protocole Diameter DEVRAIENT, avant de définir une AVP Enumerated, considérer avec attention si l'ensemble des valeurs va rester inchangé ou si de nouvelles valeurs peuvent être requises dans le proche avenir. Si une telle extension est prévue ou ne peut pas être évitée, il est RECOMMANDÉ de définir des AVP de type Unsigned32 ou Unsigned64 dans lesquelles le champ de données va contenir un espace d'adresses représentant des "valeurs" qui auront le même usage que les valeurs de Enumerated. Tandis que seules les valeurs initiales mentionnées dans la définition de l'AVP de type Enumerated seront valides comme décrit au paragraphe 4.4.2, toute valeur provenant de l'espace d'adresses de 0 à 2^32 - 1 pour les AVP de type Unsigned32 ou de 0 à 2^64 - 1 pour les AVP de type Unsigned64 sera valide au niveau du protocole de base Diameter et ne causera pas de problème d'interopérabilité pour les nœuds intermédiaires entre clients et serveurs. Seuls les clients et serveurs seront capables de traiter les valeurs à la couche application.

Comme illustration, une AVP décrivant de possibles réseaux d'accès serait définie comme suit :

L'AVP Access-Network-Type (XXX) est du type Unsigned32 et contient un espace d'adresse de 32 bits représentant des types de réseaux d'accès. Cette application définit les classes suivantes de réseau d'accès, toutes identifiées par le chiffre des milliers en notation décimale :
- 1xxx (réseaux d'accès mobile)
- 2xxx (réseaux d'accès fixe)
- 3xxx (réseaux d'accès sans fil)

Les valeurs qui entrent dans la catégorie des réseaux d'accès mobile sont utilisés pour informer un homologue qu'une demande a été envoyée pour un utilisateur rattaché à un réseau d'accès mobile. Les valeurs suivantes sont définies dans cette application :

1001: 3GPP-GERAN
L'utilisateur est rattaché à un réseau d'accès radio du système mondial pour les communications mobiles (GSM, *Global System for Mobile communications*) aux débits binaires améliorés pour l'évolution du GSM (EDGE).

1002: 3GPP-UTRAN-FDD
L'utilisateur est rattaché à un réseau d'accès du système universel pour les télécommunications mobiles (UMTS, *Universal Mobile Telecommunications System*) qui utilise le duplexage à division de fréquence pour le duplexage.

À la différence d'une AVP Enumerated, toute nouvelle valeur peut être ajoutée dans l'espace d'adresses défini par cette AVP Unsigned32 sans modifier la définition de l'AVP. Il n'y a donc aucun risque de problème de rétro compatibilité, en particulier lorsque des nœuds intermédiaires peuvent être présents entre les points d'extrémité Diameter.

Dans la même veine, les AVP de type Enumerated sont trop souvent utilisées comme un simple fanion booléen, indiquant, par exemple, une permission ou capacité spécifique ; donc, seules trois valeurs sont définies, par exemple, VRAI/ FAUX, AUTORISÉ/NON AUTORISÉ, ou PRIS EN CHARGE/NON PRIS EN CHARGE. C'est un concept sous optimal car il limite l'extensibilité de l'application : toute nouvelle capacité/permission devrait être prise en charge par une nouvelle AVP ou une nouvelle valeur Enumerated de l'AVP déjà définie, avec les problèmes de rétro compatibilité décrits ci-dessus. Au lieu d'utiliser une AVP Enumerated comme un fanion booléen, les concepteurs de protocole DEVRAIENT utiliser des AVP de type Unsigned32 ou Unsigned64 dans lesquels le champ de données serait défini comme un gabarit binaire dont le réglage des bits serait décrit dans la spécification d'application Diameter pertinente. De telles AVP peuvent être réutilisées et étendues sans impact majeur sur l'application Diameter. Le gabarit binaire DEVRAIT laisser de la place pour de futurs ajouts. Des exemples d'AVP qui utilisent des gabarits binaires sont l'AVP Session-Binding définie dans la [RFC6733] et l'AVP MIP6-Feature-Vector définie dans la [RFC5447].

5.7 Acheminement de message spécifique d'application

Comme décrit dans la [RFC6733], une demande Diameter qui doit être envoyés à un serveur de rattachement desservant un domaine spécifique, mais pas à un serveur spécifique (comme la première demande d'une série d'allers-retours) va contenir une AVP Destination-Realm et pas d'AVP Destination-Host.

Pour une telle demande, l'acheminement du message s'appuie généralement seulement sur l'AVP Destination-Realm et l'identifiant d'application présent dans l'en-tête du message de demande. Cependant, certaines applications peuvent devoir s'appuyer sur l'AVP User-Name ou toute autre AVP spécifique de l'application présente dans la demande pour déterminer la destination finale d'une demande, par exemple, pour trouver le serveur AAA cible qui héberge les informations d'autorisation pour un certain utilisateur lorsque plusieurs serveurs AAA sont adressables dans le domaine.

Dans un tel contexte, les mécanismes d'acheminement de base décrits dans la [RFC6733] ne conviennent pas complètement, et des mécanismes de niveau application supplémentaires DOIVENT être décrits dans la documentation de l'application pour fournir un tel acheminement fondé sur l'AVP spécifique. Une telle fonctionnalité sera hébergée normalement par un agent mandataire spécifique de l'application qui sera chargé des décisions d'acheminement sur la base des AVP spécifiques reçues.

Des exemples de telles fonctions d'acheminement spécifique d'application peuvent être trouvés dans les applications Cx/Dx ([TS29.228] et [TS29.229]) du sous système 3GPP IP multimédia, dans lequel l'agent mandataire (fonction de localisation d'abonné (SLF, *Subscriber Location Function*) utilise des identités spécifiques de l'application trouvées dans la demande pour déterminer la destination finale du message.

Quels que soit les critères utilisés pour établir l'acheminement de la demande, l'acheminement de la réponse DOIT suivre le chemin inverse de celui de la demande, comme décrit dans la [RFC6733], la réponse étant envoyée à la source de la demande reçue, en utilisant les états de transaction et la correspondance d'identifiant bond par bond. Cela assure que les agents mandataires ou de relais Diameter sur le chemin d'acheminement de la demande seront capables de libérer l'état de transaction à réception de la réponse correspondante, évitant d'inutiles reprise sur défaillance. De plus, en particulier dans les cas d'itinérance, les agents mandataires sur le chemin doivent être capables d'appliquer les politiques locales lorsque ils reçoivent la réponse du serveur durant l'authentification/autorisation et/ou les procédures de comptabilité, et de conserver à jour les informations d'état de session en gardant trace de toutes les sessions actives autorisées. Donc, les concepteurs d'application NE DOIVENT PAS modifier les principes d'acheminement de réponse décrits dans la [RFC6733] lorsque ils définissent une nouvelle application.

5.8 Agents de traduction

Comme défini dans la [RFC6733], un agent de traduction est un appareil qui assure l'inter fonctionnement entre Diameter et un autre protocole AAA, comme RADIUS.

Dans le cas de RADIUS, on pensait initialement que définir une fonction de traduction découlerait directement de l'adoption de quelques principes de base, par exemple, par l'utilisation d'une gamme partagée de valeurs de code pour les attributs RADIUS et les AVP Diameter. Des lignes directrices pour la mise en œuvre d'un agent de traduction RADIUS-Diameter ont été mises dans l'application de NAS Diameter [RFC4005].

Cependant, il a été reconnu qu'un tel mécanisme de traduction n'était pas si évident et une analyse plus profonde du protocole a été nécessaire pour s'assurer d'un inter fonctionnement efficace entre RADIUS et Diameter. De plus, les exigences d'inter

fonctionnement dépendent des fonctionnalités fournies par l'application Diameter à spécifier, et une analyse au cas par cas est nécessaire. Par conséquent, tout le matériel relatif à la traduction de RADIUS en Diameter est retiré de la nouvelle version de la spécification d'application de NAS Diameter [RFC7155], qui déconseille la [RFC4005].

Donc, les concepteurs de protocoles NE DEVRAIENT PAS supposer la disponibilité d'un agent de passerelle Diameter/RADIUS "standard" lorsque ils prévoient d'interopérer avec l'infrastructure RADIUS. Ils DEVRAIENT spécifier le mécanisme de traduction requis avec l'application Diameter, si nécessaire. Cette recommandation s'applique à toute sorte de traduction.

5.9 Échange de capacités d'application de bout en bout

Les applications Diameter peuvent s'appuyer sur les AVP facultatives pour échanger des capacités et dispositifs spécifiques de l'application. Ces AVP peuvent être échangées de bout en bout à la couche application. Des exemples peuvent en être trouvé dans l'AVP MIP6-Feature-Vector dans la [RFC5447] et dans l'AVP QoS-Capability dans la [RFC5777].

Des AVP de capacités de bout en bout peuvent être ajoutées aux applications existantes comme AVP facultatives avec le bit M à zéro, pour annoncer la prise en charge d'une nouvelle fonctionnalité. Les receveurs qui ne comprennent pas ces AVP, ou les valeurs d'AVP, peuvent simplement les ignorer, comme déclaré dans la [RFC6733]. Lorsque ils les prennent en charge, les receveurs de ces AVP peuvent découvrir la fonctionnalité supplémentaire prise en charge par le point d'extrémité Diameter qui a généré la demande et se comporter en conséquence lors du traitement de la demande. Les envoyeurs de ces AVP peuvent en toute sécurité supposer que le point d'extrémité receveur ne prend pas en charge une fonctionnalité portée par l'AVP si elle n'est pas présente dans la réponse correspondante. Ceci est utile dans les cas où des choix de déploiement sont offerts, et où la conception générique peut être rendue disponible pour un certain nombre d'applications.

Lorsque utilisées dans une nouvelle application, ces AVP de capacités de bout en bout DEVRAIENT être ajoutées comme des AVP facultatives dans le CCF des commandes utilisées par la nouvelle application. Les concepteurs de protocoles DEVRAIENT clairement spécifier cet échange de capacités de bout en bout et le comportement correspondant des nœuds Diameter qui prennent l'application en charge.

Il est aussi important de noter que cet échange de capacités de bout en bout s'appuyant sur l'utilisation d'AVP facultatives n'est pas destiné à être un mécanisme générique de prise en charge de l'extensibilité des applications Diameter de fonctionnalité arbitraire. Lorsque les caractéristiques ajoutées changent de façon drastique l'application Diameter ou lorsque les agents Diameter doivent mettre à niveau la prise en charge de la nouvelle caractéristique, une nouvelle application DEVRAIT être définie, comme recommandé dans la [RFC6733].

5.10 Prise en charge de la comptabilité Diameter

La comptabilité peut être traitée comme une application auxiliaire qui est utilisée en soutien d'autres applications. Dans la plupart des cas, la prise en charge de la comptabilité est exigée lors de la définition de nouvelles applications. Le présent document donne deux modèles possibles pour utiliser la comptabilité:

Modèle de comptabilité étalée :
Dans ce modèle, les messages de comptabilité vont utiliser l'identifiant d'application de comptabilité Diameter de base (de valeur 3). Cela implique pour la conception que la comptabilité soit traitée comme une application indépendante, en particulier pour l'acheminement Diameter. Cela signifie que les commandes de comptabilité émanant d'une application peuvent être acheminées séparément du reste des autres messages d'application. Cela peut aussi impliquer que les messages finissent dans un serveur de comptabilité central. Un modèle de comptabilité étalée est un bon choix de conception quand :

* L'application elle-même ne définit pas ses propres commandes de comptabilité.
* L'architecture globale du système permet l'utilisation d'une comptabilité centralisée pour une ou plusieurs applications Diameter.

Centraliser la comptabilité peut avoir des avantages, mais il y a aussi des inconvénients. Le modèle suppose que le serveur de comptabilité peut différencier les messages de comptabilité reçus. Comme les messages de comptabilité reçus peuvent être pour toute application et/ou service, le serveur de comptabilité DOIT avoir une méthode pour faire correspondre les messages de comptabilité aux applications et/ou services dont il tient la comptabilité. Cela peut parfois signifier de définir de nouvelles AVP, de vérifier la présence, l'absence, ou le contenu des AVP existantes, ou de vérifier le contenu de l'enregistrement comptable lui-même. Un de ces moyens pourrait être d'insérer dans la demande envoyée au serveur de comptabilité une AVP Auth-Application-Id contenant l'identifiant de l'application pour laquelle est envoyée la demande de comptabilité. Mais en général, il n'y a pas un schéma clair et générique pour trier ces messages. Donc, ce modèle NE DEVRAIT PAS être utilisé lorsque tous les messages de comptabilité reçus ne peuvent pas être clairement identifiés et triés. Dans la plupart des cas, l'utilisation du modèle de comptabilité couplée est RECOMMANDÉ.

Modèle de comptabilité couplée :
Dans ce modèle, les messages de comptabilité vont utiliser l'identifiant d'application de l'application qui utilise le service de comptabilité. Cette conception implique que les messages de comptabilité soient étroitement couplés à l'application elle-même, ce qui signifie que les messages de comptabilité seront acheminés comme les autres messages d'application. Il sera alors de la responsabilité du serveur d'application (entité d'application qui reçoit le message ACR) d'envoyer les enregistrements comptables portés par les messages de comptabilité au serveur de comptabilité approprié. Le serveur d'application est aussi chargé de la formulation d'une réponse appropriée (ACA). Un modèle de comptabilité couplé est un bon choix de conception lorsque :

* L'architecture ou le déploiement du système ne fournit pas un serveur de comptabilité qui prend en charge Diameter. Par conséquent, le serveur d'application DOIT être provisionné à utiliser différents protocoles pour accéder au serveur de comptabilité, par exemple, via le protocole léger d'accès à un répertoire (LDAP, *Lightweight Directory Access Protocol*), SOAP, etc. Ce cas inclut la prise en charge de plus anciens systèmes de comptabilité qui n'ont pas accès à Diameter.
* L'architecture ou le déploiement du système exige que le service comptable pour l'application spécifique soit traité par l'application elle-même.

Dans tous les cas ci-dessus, il n'y aura généralement pas d'accès Diameter direct au serveur de comptabilité.

Ces modèles donnent une base pour utiliser les messages de comptabilité. Les concepteurs d'applications peuvent évidemment s'écarter de ces modèles pourvu que les facteurs évoqués ici soient aussi bien pris en compte. À titre de recommandation générale, les concepteurs d'application NE DEVRAIENT PAS définir un nouvel ensemble de commandes pour porter des enregistrements de comptabilité spécifiques d'application.

5.11 Mécanismes de sécurité de Diameter

Comme spécifié dans la [RFC6733], l'échange de messages Diameter DEVRAIT être sécurisé entre les homologues du voisinage Diameter en utilisant la sécurité de la couche Transport (TLS, *Transport Layer Security*) / TCP ou la sécurité de couche transport de datagrammes (DTLS, *Datagram Transport Layer Security*) / protocole de transmission de contrôle de flux (SCTP, *Stream Control Transmission Protocol*). Cependant, IPsec PEUT aussi être déployé pour sécuriser la communication entre les homologues Diameter. Lorsque IPsec est utilisé à la place de TLS ou DTLS, les recommandations suivantes s'appliquent.

L'encapsulation de charge utile de sécurité (ESP, *Encapsulating Security Payload*) IPsec [RFC4301] en mode transport avec chiffrement non nul et des algorithmes d'authentification DOIT être utilisée pour assurer l'authentification par paquet, la protection de l'intégrité et de la confidentialité et pour prendre en charge les mécanismes de IPsec de protection contre la répétition. La version 2 du protocole d'échange de clés Internet (IKEv2, *Internet Key Exchange Protocol Version 2*) [RFC7296] DEVRAIT être utilisée pour effectuer l'authentification mutuelle et pour établir et maintenir les associations de sécurité (SA).

La version 1 de IKE (IKEv1), définie dans la [RFC2409], était initialement utilisée pour l'authentification des homologues, la négociation des associations de sécurité, et la gestion des clés dans la [RFC3588]. Pour une migration plus facile vers IKEv2 de la mise en œuvre obsolète fondée sur IKEv1, les signatures numériques RSA et les clés pré partagées DEVRAIENT être prises en charge dans IKEv2. Cependant, si IKEv1 est utilisée, les mises en œuvre DEVRAIENT suivre les lignes directrices du paragraphe 13.1 de la [RFC3588].

6. Définition d'extensions génériques pour Diameter

Les extensions Diameter génériques sont les AVP, les commandes, ou les applications qui sont conçues pour prendre en charge d'autres applications Diameter. Ce sont des applications auxiliaires destinées à améliorer le protocole Diameter lui-même ou des applications/fonctionnalités Diameter. Les exemples incluent les extensions pour prendre en charge la redirection fondée sur le domaine des demandes Diameter (voir la [RFC7075]) portant un ensemble spécifique de paramètres de priorité qui influencent la distribution des ressources (voir la [RFC6735]) et la prise en charge des AVP de qualité de service (voir la [RFC5777]).

Comme les extensions génériques peuvent couvrir de nombreux aspects de Diameter et des applications Diameter, il n'est pas possible d'énumérer tous les scénarios. Cependant, les considérations les plus communes sont les suivantes :

Rétro compatibilité : Lorsque ils définissent des extensions génériques conçues pour être prises en charge par les applications Diameter existantes, les concepteurs de protocoles DOIVENT considérer les impacts potentiels de l'introduction de la nouvelle extension sur le comportement du nœud qui ne serait pas encore mis à niveau pour prendre en charge/comprendre cette nouvelle extension. Les concepteurs DOIVENT aussi s'assurer que les nouvelles extensions ne bouleversent pas le comportement attendu de la couche de livraison de message.

Compatibilité vers l'avant : Les concepteurs de protocoles DOIVENT s'assurer que leur projet ne va pas introduire de restrictions indues sur de futures applications.

Compromis de signalisation : Les concepteurs peuvent avoir à choisir entre l'utilisation d'AVP facultatives portées sur des commandes existantes et définir de nouvelles commandes et applications. Les AVP facultatives sont plus simples à mettre en œuvre et peuvent ne pas imposer de changement aux applications existantes. Cependant, cela lie l'envoi de données d'extension à la transmission d'un message par l'application. Cela a des conséquences si l'application et les extensions ont des exigences de rythme différentes. L'utilisation des commandes et applications résout ce problème, mais le compromis porte sur la complexité supplémentaire de la définition et du déploiement d'une nouvelle application. Il appartient au concepteur de trouver un bon équilibre qui fasse un compromis entre les différentes exigences de l'extension.

En pratique, les extensions génériques utilisent souvent des AVP facultatives parce que elles sont simples et non intrusives pour l'application qui va les porter. Les homologues qui ne prennent pas en charge les extensions génériques n'ont pas besoin de comprendre ou reconnaître ces AVP facultatives. Cependant, il est RECOMMANDÉ que les auteurs de l'extension spécifient le contexte ou l'usage des AVP facultatives. Par exemple, dans le cas où l'AVP peut être utilisée seulement par un ensemble spécifique d'applications, la spécification DOIT énumérer ces applications et les scénarios où les AVP facultatives seront utilisées. Dans le cas où les AVP facultatives peuvent être portées par n'importe quelle application, il devrait suffire de spécifier ce cas d'utilisation et peut-être de fournir des exemples spécifiques d'applications qui les utilisent.

Dans la plupart des cas, ces AVP facultatives portées par les applications seront définies comme des AVP groupées, et elles vont encapsuler toutes les fonctionnalités de l'extension générique. En pratique, il n'est pas exceptionnel que l'AVP groupée encapsule une AVP existante qui avait été précédemment définie comme obligatoire (bit 'M' à 1), par exemple, les interfaces Cx/Dx de sous systèmes multimédia IP du 3GPP ([TS29.228] et [TS29.229]).

7. Lignes directrices pour l'enregistrement des valeurs Diameter

Comme résumé à la Section 3 du présent document et décrit plus en détails au paragraphe 1.3 de la [RFC6733], il y a quatre façons principales d'étendre Diameter. Le processus pour définir de nouvelles fonctionnalités varie légèrement selon les différentes extensions. Cette section donne aux concepteurs de protocoles des lignes directrices concernant la définition des valeurs de possibles extensions Diameter et les interactions nécessaires avec l'IANA pour enregistrer la nouvelle fonction.

a. Définir de nouvelles valeurs d'AVP

Les spécifications qui définissent les AVP et les valeurs des AVP DOIVENT fournir des indications pour définir de nouvelles valeurs et la politique correspondante pour l'ajout de ces valeurs. Par exemple, la [RFC5777] définit l'AVP Treatment-Action, qui contient une liste des valeurs valides correspondant a des actions prédéfinies (abandonner, formater, marquer, permettre). Cet ensemble de valeurs peut être étendu en suivant la politique de spécification exigée définie dans la [RFC5226]. Comme second exemple, la spécification de base Diameter [RFC6733] définit l'AVP Result-Code qui contient un espace d'adresse de 32 bits utilisé pour identifier de possibles erreurs. Selon le paragraphe 11.3.2 de la [RFC6733], de nouvelles valeurs peuvent être allouées par l'IANA via un processus de revue par l'IETF [RFC5226].

b. Création de nouvelles AVP

Deux différents types d'espaces de noms de code d'AVP peuvent être utilisés pour créer une nouvelle AVP :

* espace de noms de code d'AVP de l'IETF,
* espaces de noms de code d'AVP spécifique de fabricant.

Dans le dernier cas, un fabricant a besoin d'avoir d'abord un numéro d'entreprise privée alloué par l'IANA, qui peut être utilisé dans le champ Vendor-Id de l'AVP spécifique de fabricant. Ce numéro d'entreprise délimite un espace de noms privé dans lequel le fabricant est responsable de l'allocation de la valeur de code d'AVP spécifique de fabricant. L'absence d'un identifiant de fabricant ou une valeur de Vendor-Id de zéro (0) dans l'en-tête d'AVP identifie les AVP standard provenant de l'espace de noms de code d'AVP de l'IETF géré par l'IANA. L'allocation des valeurs de code de l'espace de noms géré par l'IANA est conditionné par revue d'expert de la spécification qui définit les AVP ou une revue de l'IETF si un bloc d'AVP doit être alloué. De plus, les bits restants du champ Fanions d'AVP de l'en-tête de l'AVP sont aussi alloués via action de normalisation si la création de nouveaux fanions d'AVP est désirée.

c. Création de nouvelles commandes

À la différence de l'espace de noms de code d'AVP, l'espace de noms de code de commandes est plat, mais la gamme des valeurs est subdivisée en trois tronçons avec des politiques d'enregistrement de l'IANA distinctes :

* une gamme de valeurs de code de commande standard qui sont allouées via revue de l'IETF ;
* une gamme de valeurs de code de commandes spécifique de fabricant qui sont allouées sur la base du premier arrivé, premier servi ;

* une gamme de valeurs réservées pour les seuls besoins d'expérience et d'essais.

Comme avec les fanions d'AVP, les bits restants du champ Fanions de commandes de l'en-tête Diameter sont aussi alloués via une action de normalisation pour créer de nouveaux fanions de commandes, si exigé.

d. Création de nouvelles applications

De même, pour l'espace de noms de code de commandes, l'espace de noms Application-Id est plat mais divisé en deux gammes distinctes :

* une gamme de valeurs réservées pour les identifiants d'application standard, alloués après revue d'expert de la spécification qui définit l'application standard,
* une gamme pour les valeurs des applications spécifiques de fabricant, allouées par l'IANA sur la base du premier arrivé, premier servi.

La page des paramètres AAA de l'IANA se trouve à <http://www.iana.org/assignments/aaa-parameters>, et la page des numéros d'entreprise de l'IANA est disponible à <http://www.iana.org/assignments/enterprise-numbers>. Des détails sur les politiques suivies par l'IANA sur la gestion des espaces de noms (par exemple, premier arrivé, premier servi; revue d'expert; revue par l'IETF; etc.) se trouvent dans la [RFC5226].

Note : Lorsque la même fonction/extension est utilisée par plus d'un fabricant, il est RECOMMANDÉ qu'une extension standard soit définie. De plus, une extension spécifique de fabricant DEVRAIT être enregistrée pour éviter des problèmes d'interopérabilité dans le même réseau. Dans ce but, la politique d'enregistrement d'une extension a été simplifiée avec la publication de la [RFC6733], et l'espace de noms réservé aux extensions spécifiques de fabricant est assez grand pour éviter son épuisement.

8. Considérations sur la sécurité

Le présent document fournit des lignes directrices et considérations pour l'extension de Diameter et des applications Diameter. Bien qu'une telle extension puisse avoir des rapports avec une fonctionnalité de sécurité, le document ne donne pas explicitement de directives supplémentaires pour améliorer Diameter à l'égard de la sécurité. Cependant, comme directive générale, il est recommandé que toute extension Diameter NE DEVRAIT PAS être en rupture avec le concept de sécurité donné dans la [RFC6733]. En particulier, on répète ici que toute commande définie ou réutilisée dans une nouvelle application Diameter DEVRAIT être sécurisée en utilisant TLS [RFC5246] ou DTLS/SCTP [RFC6083] et NE DOIT PAS être utilisée sans TLS, DTLS, ou IPsec [RFC4301]. Lors de la définition d'une nouvelle extension Diameter, tout impact possible des principes de sécurité existants décrits dans la [RFC6733] DOIT être évalué avec soin et documenté dans la spécification d'application Diameter.

9. Références

9.1 Références normatives

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

[RFC6733] V. Fajardo, J. Arkko, J. Loughney, G. Zorn, "Protocole de base Diameter", octobre 2012. (*Remplace les* RFC3588, RFC5719) *(P.S.)*

9.2 Références pour information

[Q.3303.3] Recommandation UIT-T Q.3303.3, "Protocole n° 3 de contrôle de ressources : protocoles à l'interface Rw entre l'entité physique de décision de politique (PD-PE) et une entité physique d'application de politique (PE-PE) : profil Diameter version 3", Union Internationale des Télécommunications, août 2008.

[RFC2409] D. Harkins et D. Carrel, "L'échange de clés Internet (IKE)", novembre 1998. *(Obsolète, voir la* RFC4306)

[RFC3588] P. Calhoun et autres, "Protocole fondé sur Diameter", septembre 2003. (*Remplacée par la* RFC6733) *(P.S.)*

[RFC4005] P. Calhoun et autres, "Application de serveur d'accès réseau Diameter", août 2005. *(P.S.) (Remplacée par* RFC7155)

[RFC4072] P. Eronen et autres, "Application Diameter du protocole d'authentification extensible (EAP)", août 2005. *(P.S. ; MàJ par* RFC8044)

[RFC4301] S. Kent et K. Seo, "Architecture de sécurité pour le protocole Internet", décembre 2005. *(P.S.) (Remplace la* RFC2401)

[RFC4740] M. Garcia-Martin et autres, "Application Diameter dans le protocole d'initialisation de session (SIP)", novembre 2006. *(P.S.)*

[RFC5226] T. Narten et H. Alvestrand, "Lignes directrices pour la rédaction d'une section Considérations relatives à l'IANA dans les RFC", BCP 26, mai 2008. (*Remplace* RFC2434 *; remplacée par* RFC8126)

[RFC5246] T. Dierks, E. Rescorla, "Version 1.2 du protocole de sécurité de la couche Transport (TLS)", août 2008. *(P.S.* ; *remplace* RFC3268, 4346, 4366 ; *MàJ* RFC4492 ; *rendue obsolète par la* RFC8446)

[RFC5447] J. Korhonen et autres, "Serveur IPv6 mobile Diameter : prise en charge de l'interaction entre serveur d'accès réseau et serveur Diameter", février 2009. *(P.S.)*

[RFC5777] J. Korhonen, H. Tschofenig, M. Arumaithurai, M. Jones, A. Lior, "Attributs de classification du trafic et de qualité de service pour Diameter", février 2010. *(P. S.)*

[RFC6083] M. Tyexen, R. Seggelmann et E. Rescorla, "Sécurité de la couche Transport de datagrammes (DTLS) pour le protocole de transmission de contrôle de flux (SCTP)", janvier 2011. *(P.S.)*

[RFC6735] K. Carlberg, T. Taylor, "Paires Attribut-valeur de priorité Diameter", octobre 2012. *(P.S.)*
[RFC7075] T. Tsou, R. Hao, T. Taylor, "Redirection fondée sur le domaine dans Diameter", nov. 2013. (*MàJ* RFC6733) *(P.S.)*

[RFC7155] G. Zorn, Ed., "Application de serveur d'accès réseau Diameter", avril 2014. (*Remplace* RFC4005) *(P.S.)*

[RFC7296] C. Kaufman, et autres, "Protocole d'échange de clé Internet version 2 (IKEv2)", octobre 2014. STD 79. *(MàJ par* RFC7670, RFC8247)

[TS29.228] 3rd Generation Partnership Project, "Technical Specification Group Core Network and Terminals; IP Multimedia IM) Subsystem Cx and Dx Interfaces; Signalling flows and message contents", 3GPP TS 29.228, septembre 2014, < http://www.3gpp.org/ftp/Specs/html-info/29228.htm >.

[TS29.229] 3rd Generation Partnership Project, "Technical Specification Group Core Network and Terminals; Cx and Dx interfaces based on the Diameter protocol; Protocol details", 3GPP TS 29.229, septembre 2014, <http://www.3gpp.org/ftp/Specs/html-info/29229.htm >.

[TS29.328] 3rd Generation Partnership Project, "Technical Specification Group Core Network and Terminals; IP Multimedia (IM) Subsystem Sh interface; Signalling flows and message contents", 3GPP TS 29.328, septembre 2014, <http://www.3gpp.org/ftp/Specs/html-info/29328.htm >.

[TS29.329] 3rd Generation Partnership Project, "Technical Specification Group Core Network and Terminals; Sh Interface based on the Diameter protocol; Protocol details", 3GPP TS 29.329, septembre 2014, <http://www.3gpp.org/ftp/Specs/html-info/29329.htm >.

Contributeurs

Le contenu du présent document a été influencé par l'équipe de conception créée pour revisiter les règles d'extensibilité de Diameter. L'équipe a été formée en février 2008 et a fini son travail en juin 2008. En plus des individus cités dans la section Adresse des auteurs, l'équipe comportait : Avi Lior, Glen Zorn, Jari Arkko, Jouni Korhonen, Mark Jones, Tolga Asveren, Glenn McGregor, Dave Frascone.

Nous tenons à remercier Tolga Asveren, Glenn McGregor, et John Loughney de leurs contributions comme co-auteurs des versions antérieures du présent document.

Remerciements

Nous avons beaucoup apprécié les conseils fournis par ceux qui ont mis en œuvre Diameter et qui ont souligné les problèmes et questions traités par le présent document. Les auteurs remercient aussi Jean Mahoney, Ben Campbell, Sebastien Decugis, et

Benoit Claise de leur précieuse relecture détaillée et de leurs commentaires sur ce document.

Adresse des auteurs

Lionel Morand (editor)
Orange Labs
38/40 rue du General Leclerc
Issy-Les-Moulineaux Cedex 9 92794
France
téléphone : +33145296257
mél : lionel.morand@orange.com

Victor Fajardo
Fluke Networks
mél : vf0213@gmail.com

Hannes Tschofenig
Hall in Tirol 6060
Austria
mél : Hannes.Tschofenig@gmx.net
URI : http://www.tschofenig.priv.at