Groupe de travail Réseau
Request for Comments : 5533
Catégorie : Sur la voie de la normalisation
Traduction Claude Brière de L'Isle

E. Nordmark, Sun Microsystems
M. Bagnulo, UC3M
juin 2009

Shim6 : protocole Shim de multi rattachement de niveau 3 pour IPv6

Résumé
Le présent document définit le protocole Shim6, un outil de couche 3 pour assurer la souplesse de localisateur en dessous des protocoles de transport, afin que le multi rattachements puisse être fourni pour IPv6 avec des propriétés de reprise sur défaillance et de partage de charge, sans supposer qu'un site multi rattachements va avoir un préfixe d'adresse IPv6 indépendant du fournisseur annoncé dans le tableau global d'acheminement IPv6. Les hôtes dans un site qui a plusieurs préfixes d'adresse IPv6 alloués par le fournisseur va utiliser le protocole Shim6 spécifié dans le présent document pour établir l'état avec les hôtes homologues afin que l'état puisse ultérieurement être utilisé pour la récupération sur défaillance sur une paire de localisateurs différente, si la paire originale cessait de fonctionner.

Table des matières

1. Introduction

Le présent document décrit une approche Shim de couche 3 et un protocole pour fournir l'agilité de localisateur en dessous des protocoles de transport, afin que le multi rattachements puisse être fourni pour IPv6 avec des propriétés de reprise sur défaillance et de partage de charge [RFC3582], sans supposer qu'un site multi rattachements va avoir une adresse IPv6 indépendante du fournisseur annoncée dans le tableau d'acheminement IPv6 global. Les hôtes dans un site qui a plusieurs préfixes d'adresse IPv6 alloués par le fournisseur va utiliser le protocole Shim6 spécifié dans le présent document pour établir l'état avec les hôtes homologues afin que cet état puisse être utilisé plus tard pour récupérer de défaillances sur une paire de localisateurs différente, si la paire originale devait cesser de fonctionner (le terme de localisateur est défini à la Section 2).

Le protocole Shim6 est une solution de multi rattachements de site au sens où il permet à une communication existante de continuer quand un site qui a plusieurs connexions à l'Internet subit une panne sur un sous ensemble de ces connexions ou plus loin en amont. Cependant, le traitement Shim6 est effectué sur des hôtes individuels plutôt que par des mécanismes à l'échelle du site.

On suppose que les attaques de redirection sont empêchées en utilisant des adresses fondées sur le hachage (HBA, *Hash-Based Address*) comme défini dans la [RFC5535].

Les mécanismes d'accessibilité et de détection de défaillance, y compris comment une nouvelle paire de localisateurs active est découverte après une défaillance, sont spécifiés dans la [RFC5534]. Le présent document alloue les types de message et les types d'option pour ce sous protocole, et laisse la spécification des formats de message et d'option, ainsi que le comportement du protocole, à la RFC 5534.

1.1 Objectifs

Les objectifs de cette approche sont de :

- Préserver les communications établies en présence de certaines classes de défaillances, par exemple, les connexions TCP et les flux UDP.
- Avoir un impact minimal sur les protocoles de couche supérieure en général et sur les protocoles et applications de transport en particulier.
- Traiter les menaces pour la sécurité de la [RFC4218] par une combinaison de l'approche HBA/CGA spécifiée dans la [RFC5535] et les techniques décrites dans le présent document.
- Ne pas exiger un aller-retour supplémentaire pour établir l'état spécifique de Shim. Elle permet plutôt au trafic de couche supérieure (par exemple, TCP) de s'écouler normalement et diffère l'établissement de l'état Shim jusqu'à ce qu'un certain nombre de paquets aient été échangés.
- Tirer parti de plusieurs localisateurs/adresses pour l'étalement de la charge afin que différents ensembles de communication à un hôte (par exemple, différentes connexions) pourraient utiliser différents localisateurs de l'hôte. Noter que ceci pourrait causer un étalement inégal de la charge ; donc, on utilise le terme "étalement de charge" plutôt que "équilibrage de charge". Cette capacité pourrait permettre certaines formes d'ingénierie du trafic, mais les détails de l'ingénierie du trafic, y compris quelles exigences peuvent être satisfaites, ne sont pas spécifiées dans ce document, et

font partie des extensions potentielles à ce protocole.

1.2 Non objectifs

Le problème qu'on essaye de résoudre est le multi rattachements de site, avec la capacité d'avoir l'ensemble de préfixes de site qui change au fil du temps à cause de la dénumérotation du site. De plus, on suppose que de tels changements de l'ensemble des préfixes de localisateur peuvent être relativement lents et gérables : assez lent pour permettre que les mises à jour du DNS se propagent (car le protocole défini dans le présent document dépend du DNS pour trouver les ensembles de localisateurs appropriés). Cependant, on notera que c'est un non objectif explicite de faire que la communication survive à un événement de renumérotage (qui cause le changement de tous les localisateurs d'un hôte en un nouvel ensemble de localisateurs). Cette proposition ne tente pas de résoudre le problème relatif à la mobilité de l'hôte. Cependant, il pourrait se trouver que le protocole Shim6 puisse être un composant utile pour de futures solutions de mobilité de l'hôte, par exemple, pour l'optimisation de chemin.

Finalement, cette proposition n'essaye pas non plus de fournir un nouvel espace de noms d'identifiant de niveau réseau ou de niveau transport distinct de l'espace de noms d'adresses IP actuel. Même si un tel concept serait utile aux protocoles de couche supérieure (ULP, *Upper-Layer Protocol*) et applications, en particulier si la charge de gestion d'un tel espace de noms était négligeable et si il y avait un mécanisme efficace et sûr pour transposer les identifiants en localisateurs, un tel espace de noms n'est pas nécessaire (et de plus ne semble pas servir à grand chose) pour résoudre le problème du multi rattachements.

La proposition Shim6 ne sépare pas complètement les fonctions d'identifiant et de localisateur qui ont traditionnellement été surchargées dans l'adresse IP. Cependant, dans ce document, le terme "identifiant" ou, plus précisément, d'identifiant de couche supérieure (ULID, *Upper-Layer IDentifier*) se réfère à la fonction d'identification d'une adresse IPv6. "Localisateur" se réfère à l'acheminement de couche réseau et aux propriétés de transmission d'une adresse IPv6.

1.3 Localisateurs comme identifiants de couche supérieure (ULID)

L'approche décrite dans ce document n'introduit pas de nouvel espace de nom d'identifiant mais utilise plutôt le localisateur qui est choisi dans le contact initial avec l'homologue distant comme identifiant de couche supérieure préservé. Bien qu'il puisse y avoir au fil du temps des changements ultérieurs des localisateurs de niveau réseau choisis (en réponse aux défaillances en utilisant le localisateur original), les éléments de la pile de protocole de niveau supérieur vont continuer d'utiliser sans changement cet identifiant de niveau supérieur.

Cela implique que le choix d'ULID est effectué comme la sélection d'adresse par défaut d'aujourd'hui comme spécifié dans la [RFC3484]. Certaines extensions sont nécessaires à la RFC 3484 pour essayer différentes adresses de source, que le protocole Shim6 soit utilisé ou non, comme souligné dans [9]. En dessous, et de façon transparente, l'ajustement de multi rattachements choisit les paires de localisateurrs actives avec la paire initiale de localisateurs qui est la paire ULID. Si la communication échoue en suite, le Shim peut tester et choisir des localisateurs de remplacement. Une section suivante discute les questions qui se posent quand l'ULID choisi n'est pas initialement actif, ce qui crée le besoin de commuter les localisateurs de front.

Utiliser un des localisateurs comme ULID a certains avantages pour les applications qui ont un état de session de longue durée ou qui effectuent des rappels ou des références, parce que le nom de domaine pleinement qualifié (FQDN, *Fully Qualified Domain Name*) et l'ULID de 128 bits fonctionnent tous deux comme des brides pour les applications.

Cependant, utiliser un seul ULID de 128 bits ne fournit pas une communication sans ruptures quand ce localisateur est inaccessible. Voir dans [18] une discussion sur les implications pour l'application.

Il y a eu des discussions sur l'utilisation d'adresses non acheminables, comme les adresses locales uniques (ULA, *Unique-Local Address*) [RFC4193], comme des ULID dans une solution de multi rattachements. Bien que le présent document ne spécifie pas tous les aspects de ce problème, on estime que l'approche peut être étendue pour traiter le cas de l'adresse non acheminable. Par exemple, le protocole a déjà besoin de traiter les ULID qui ne sont pas initialement accessibles. Donc, le même mécanisme peut traiter les ULID qui sont en permanence inaccessibles de l'extérieur de leur site. Le problème devient comment faire que le protocole ait de bonnes performances quand l'ULID est connu a priori comme étant inaccessible (par exemple, l'ULID est une ULA) par exemple, en évitant toute fin de temporisation et ré-essai dans ce cas. De plus, on va devoir comprendre comment les ULA vont être entrés dans le DNS pour éviter un impact sur les performances des hôtes IPv6 existants, sans capacité Shim6 qui pourraient essayer de communiquer avec l'ULA (inaccessible).

1.4 Diffusion groupée IP

La diffusion groupée IP exige que le champ Adresse IP de source contienne un localisateur topologiquement correct pour l'interface utilisée pour envoyer le paquet, car l'acheminement de diffusion groupée IP utilise à la fois l'adresse de source et le groupe de destination pour déterminer où transmettre le paquet. En particulier, l'acheminement de diffusion groupée IP doit être capable de faire la vérification de transmission sur le chemin inverse (RPF, *Reverse Path Forwarding*). (TCP n'est pas très différent de la situation avec le filtrage d'entrée largement mis en œuvre [RFC2827] pour l'envoi individuel.)

Alors qu'en théorie il serait possible d'appliquer la retransposition Shim des champs d'adresse IP entre les ULID et les localisateurs, le fait que tous les receveurs de diffusion groupée auraient besoin de connaître la transposition à effectuer rend une telle approche difficile en pratique. Donc, il est raisonnable d'avoir des ULP de diffusion groupée qui fonctionnent directement sur les localisateurs et de ne pas utiliser l'ajustement. C'est une disposition assez naturelle pour les protocoles qui utilisent RTP [RFC3550], car RTP a déjà un identifiant explicite sous la forme du champ de source de synchronisation (SSRC) dans les en-têtes RTP. Donc, les champs d'adresse IP réels ne sont pas importants pour l'application.
En résumé, la diffusion groupée IP n'a pas besoin de Shim pour retransposer les adresses IP.

Cela n'empêche pas le receveur de diffusion groupée de changer ses localisateurs, car le receveur n'est pas explicitement identifié ; l'adresse de destination est une adresse de diffusion groupée et non le localisateur d'envoi individuel du receveur.

1.5 Implications de dénumérotage

Comme déclaré ci-dessus, cette approche n'essaye pas de faire survivre la communication au dénumérotage dans le cas général.
Quand un hôte est dénuméroté, l'effet est que un ou plusieurs localisateurs deviennent invalides, et zéro, un ou plusieurs localisateurs sont ajoutés à l'interface réseau de l'hôte. Cela signifie que l'ensemble de localisateurs qui est utilisé dans le Shim vont changer, que le Shim peut traiter tant que tous les localisateurs originaux ne deviennent pas invalides en même temps ; la capacité de Shim à traiter cela dépend aussi du remps nécessaire pour mettre à jour le DNS et pour que ces mises à jour se propagent.
Mais les adresses IP sont aussi utilisées comme des ULID, et faire que la communication survive à l'invalidité des localisateurs peut potentiellement causer une certaine confusion aux couches supérieures. Le fait qu'un ULID pourrait être utilisé avec un localisateur différent au fil du temps ouvre la possibilité que la communication entre deux ULID pourrait continuer de fonctionner après qu'un ou les deux ULID ne soient plus accessibles comme localisateurs, par exemple, du fait d'un événement de dénumérotage. Cela ouvre la possibilité que l'ULID (ou au moins le préfixe sur lequel il se fonde) puisse être réalloué à un autre site alors qu'il est encore utilisé (avec un autre localisateur) pour une communication existante.
Dans le pire des cas, on pourrait finir avec deux hôtes séparés utilisant le même ULID alors que tous deux communiquent avec le même hôte.
Cette potentielle source de confusion est évitée en exigeant que toute communication utilisant un ULID DOIT être terminée quand l'ULID devient invalide (parce que le préfixe sous-jacent devient invalide). Ce comportement peut être réalisé par l'élimination explicite de l'état de Shim quand l'ULID devient invalide. Le mécanisme de récupération de contexte va alors avertir l'homologue que le contexte est parti et que l'ULID n'est plus présent au même localisateur.

1.6 Placement de l'ajustement

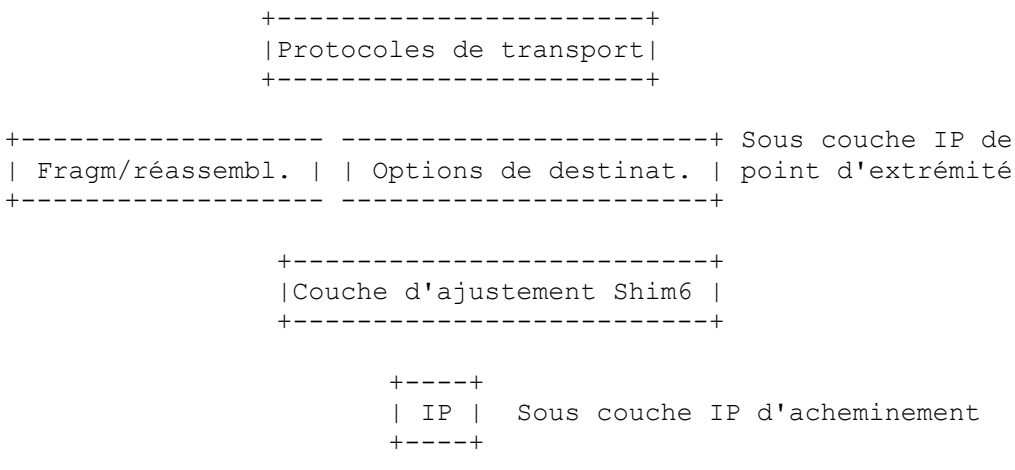


Figure 1 : Pile de protocoles

La proposition utilise une couche d'ajustement de multi rattachements au sein de la couche IP, c'est-à-dire, en dessous des ULP, comme le montre la Figure 1, afin d'assurer l'indépendance de l'ULP. La couche Shim de multi rattachements se comporte comme si elle était associée à un en-tête d'extension, qui serait placé après tout en-tête relatif à l'acheminement dans le paquet (comme toutes options bond par bond). Cependant, quand la paire de localisateurs est la paire d'ULID, il n'y a pas de données qui doivent être portées dans un en-tête d'extension ; donc, aucun n'est nécessaire dans ce cas.

Mettre en couches l'en-tête de fragmentation au dessus de l'ajustement de multi rattachements rend le réassemblage robuste dans le cas d'une rupture de l'acheminement multi-chemins qui résulte en utilisant différents chemins, et donc potentiellement des localisateurs de source différents, pour les différents fragments. Donc, la couche d'ajustement multi rattachements est placée entre la sous couche de point d'extrémité IP (qui traite la fragmentation et le réassemblage) et la sous couche d'acheminement IP (qui choisit le prochain bond et l'interface à utiliser pour envoyer les paquets).

Les applications et protocoles de couche supérieure utilisent les ULID que la couche Shim6 transpose de/en différents localisateurs. La couche Shim6 maintient l'état, appelé le contexte de paire d'ULID, par paire d'ULID (c'est-à-dire, cet état s'applique à toutes les connexions d'ULP entre la paire d'ULID) afin d'effectuer cette transposition. La transposition est effectuée de façon cohérente chez l'envoyeur et le receveur afin que les ULP voient les paquets qui apparaissent comme envoyés en utilisant des ULID de bout en bout. Cette propriété est maintenue même si les paquets voyagent à travers le réseau en contenant des localisateurs dans les champs d'adresse IP, et même si ces localisateurs peuvent être changés par la couche Shim6 qui les transmet.

L'état du contexte est maintenu par ULID distant, c'est-à-dire, approximativement par hôte homologue, et pas à une granularité plus fine. En particulier, l'état du contexte est indépendant des ULP et de toute connexion d'ULP. Cependant, la capacité de fourchement permet aux ULP à capacité Shim6 d'utiliser plus d'une paire de localisateurs à la fois pour une seule paire d'ULID.

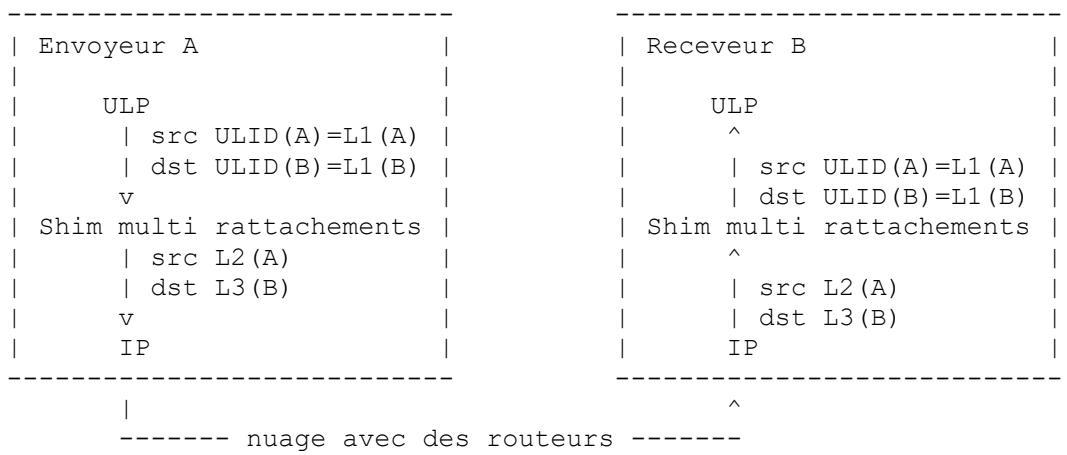


Figure 2 : Transposition avec changement de localisateurs

Le résultat de cette transposition cohérente est qu'il n'y a pas d'impact sur les ULP. En particulier, il n'y a pas d'impact sur les sommes de contrôle de pseudo en-tête et d'identification de connexion.

Conceptuellement, on pourrait voir cette approche comme si les ULID et les localisateurs étaient présents dans chaque paquet, et comme si un mécanisme de compression d'en-tête était appliqué qui supprime le besoin que les ULID soient portés dans les paquets une fois que l'état de compression a été établi. Afin que le receveur re-crée un paquet avec les ULID corrects, il est besoin d'inclure une "étiquette de compression" dans les paquets de données. Cela sert à indiquer le contexte correct à utiliser pour la décompression quand la paire de localisateurs dans le paquet est insuffisante pour identifier le contexte de façon univoque.

Il y a différents types d'interactions entre la couche Shim6 et les autres protocoles. Ces interactions sont influencées par l'usage des adresses dans ces autres protocoles et par l'impact de la transposition Shim6 sur ces usages. On trouvera une analyse détaillée des interactions des différents protocoles, incluant le protocole de transmission de commandes de flux (SCTP, *Stream Control Transmission Protocol*), IP mobile (MIP, *Mobile IP*), et le protocole d'identité d'hôte (HIP, *Host Identity Protocol*) dans la [RFC6629]. De plus, certaines applications peuvent nécessiter une interaction plus riche avec la sous couche Shim6. Pour permettre cela, une API [RFC6316] a été définie pour permettre un plus grand contrôle et échange d'informations pour les applications qui en ont besoin.

1.7 Ingénierie du trafic

Au moment de la rédaction, il n'apparaît pas clairement quelles exigences pour l'ingénierie du trafic ont un sens pour le protocole Shim6, car les exigences doivent résulter en un comportement utile tout en étant mises en œuvre en utilisant un mécanisme d'agilité de localisateur d'hôte à hôte comme Shim6.

Inhérente dans un mécanisme multi rattachements adaptable qui sépare la fonction de localisateur de l'adresse IP de la fonction d'identification de l'adresse IP est que chaque hôte finit avec plusieurs localisateurs. Cela signifie que, au moins pour le contact initial, c'est l'application de l'homologue distant (ou la couche qui travaille en son nom) qui a besoin de choisir l'ULID initial, qui devient automatiquement le localisateur initial. Dans le cas de Shim6, ceci est effectué en appliquant le choix d'adresse de la RFC 3484.

Ceci est assez différent du cas courant du multi rattachements IPv4 où le site a un seul préfixe d'adresse IP, car dans ce cas, l'homologue n'effectue pas de choix d'adresse de destination.

Donc, dans un "seul préfixe multi rattachements", le site (et dans de nombreux cas son FAI amont) peut utiliser BGP pour exercer un certain contrôle du chemin d'entrée utilisé pour accéder au site. Cette capacité n'existe pas par elle-même dans les approches de "multiples préfixes multi rattachements" comme Shim6. Il est concevable que pourraient être développées des extensions permettant des lignes directrice de site ou de fournisseur des mécanismes fondés sur l'hôte. Mais on devrait noter que l'ingénierie du trafic via BGP, MPLS, ou autres techniques similaires peut encore être appliquée pour le trafic sur chaque préfixe individuel ; Shim6 ne supprime pas cette capacité. Il fournit des capacités supplémentaires pour que les hôtes choisissent entre les préfixes.

Ces capacités comportent aussi des risques de comportement non optimal quand plus d'un mécanisme tente de corriger les problèmes en même temps. Cependant, on devrait noter que ceci n'est pas nécessairement une situation apportée par Shim6. Une forme plus contrainte de cette capacité existe déjà dans IPv6 lui-même, via sa prise en charge de préfixes multiples et des règles de choix d'adresse pour lancer de nouvelles communications. Même les hôtes IPv4 avec plusieurs interfaces peuvent avoir des capacités limitées de choisir les interfaces sur lesquelles ils communiquent. De même, les couhes supérieures peuvent choisir des adresses différentes.

En général, on s'attend à ce que Shim6 soit applicable dans des sites relativement petits et des hôtes individuels où des opérations d'ingénierie du trafic de style BGP sont indisponibles, peu probables, ou si il fonctionne avec un adressage indépendant du fournisseur, éventuellement dommageables, en considérant les taux de croissance du tableau d'acheminement global.

Le protocole fournit un bouche-trou, sous la forme de l'option Préférences de localisateur, qui peut être utilisée par les hôtes pour exprimer des valeurs de priorité et de pondération pour chaque localisateur. Cette option est simplement un bouche-trou quand elle vient fournir de l'ingénierie du trafic ; afin d'utiliser cela dans un grand site, il devrait y avoir un mécanisme par lequel l'hôte peut trouver quelles valeurs de préférence utiliser, soit statiquement (par exemple, une nouvelle option DHCPv6) soit dynamiquement.

Donc, l'ingénierie du trafic est mentionnée comme possible extension à l'Appendice A.

2. Terminologie

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],.

2.1 Définitions

Le présent document introduit les termes suivants :

Protocole de couche supérieure (ULP, *Upper-Layer Protocol*) : couche de protocole immédiatement au-dessus de IP. Des exemples sont des protocoles de transport tels que TCP et UDP, des protocoles de contrôle comme ICMP, des protocoles d'acheminement comme OSPF, et des protocoles Internet ou de couche inférieure qui sont "tunnelés" sur IP (c'est-à-dire, encapsulé dans) comme l'échange de paquets Internet (IPX, *Internet Packet Exchange*) AppleTalk, ou IP

lui-même.

Interface : rattachement d'un nœud à une liaison.

Adresse : nom de couche IP qui contient à la fois une signification topologique et agit comme identifiant univoque pour une interface. 128 bits. Le présent document utilise seulement le terme "adresse" dans le cas il n'est pas spécifié si c'est un localisateur ou un identifiant.

Localisateur : nom topologique de couche IP pour une interface ou un ensemble d'interfaces. 128 bits. Les localisateurs sont portés dans les champs d'adresse IP lorsque les paquets traversent le réseau.

Identifiant : nom de couche IP pour un point d'extrémité de couche IP. Le nom du point d'extrémité de transport est une fonction du protocole de transport et va normalement inclure l'identifiant IP plus un numéro d'accès.

Note : la présente proposition ne spécifie aucune nouvelle forme d'identifiant de couche IP, mais sépare quand même les propriétés d'identification et de localisation des adresses IP.

Identifiant de couche supérieure (ULID, *upper-layer identifier*) : adresse IP qui a été choisie pour la communication avec un homologue pour être utilisée par le protocole de couche supérieure. 128 bits. C'est utilisé pour le calcul de la somme de contrôle de pseudo en-tête et l'identification de connexion dans l'ULP. Différents ensembles de communication à un hôte (par exemple, différentes connexions) pourraient utiliser des ULID différents afin de permettre un étalement de charge. Comme l'ULID est juste un des localisateurs/adresses IP du nœud, il n'est pas besoin d'espace de noms et de mécanismes d'allocation séparés.

Champ Adresse : les champs Adresse de source et Adresse de destination dans l'en-tête IPv6. Comme IPv6 est spécifié actuellement, ces champs portent des "adresses". Si les identifiants et les localisateurs sont séparés, ces champs vont contenir des localisateurs pour les paquets dans le réseau.

FQDN *(Fully Qualified Domain Name)* : nom de domaine pleinement qualifié

Contexte de paire d'ULID : état que le Shim multi rattachements maintient entre une paire d'identifiants de couche supérieure. Le contexte est identifié par une étiquette de contexte pour chaque direction de la communication et aussi par une paire d'ULID et un identifiant d'instance fourchée (voir ci-dessous).

Étiquette de contexte : chaque extrémité du contexte alloue une étiquette de contexte pour le contexte. C'est utilisé pour associer de façon univoque les paquets de contrôle reçus et les en-têtes d'extension de charge utile Shim6 comme appartenant au contexte.

Paire de localisateurs courante : chaque extrémité du contexte a une paire de localisateurs courante qui est utilisée pour envoyer des paquets à l'homologue. Cependant, les deux extrémités pourraient utiliser des paires différentes de localisateurs courantes.

Contexte par défaut : à l'extrémité envoyeuse, le Shim utilise la paire d'ULID (passée de l'ULP) pour trouver le contexte pour cette paire. Donc, normalement, un hôte peut avoir au plus un contexte pour une paire d'ULID. On appelle cela le "contexte par défaut".

Fourchement de contexte : mécanisme qui permet aux ULP qui sont à capacité de localisateurs multiples d'utiliser des contextes séparés pour la même paire d'ULID, afin d'être capable d'utiliser différentes paires de localisateurs pour différentes communications au même ULID. Le fourchement de contexte cause la création de plus que juste le contexte par défaut pour une paire d'ULID.

Identifiant d'instance fourchée (FII, *Identifiant d'instance fourchée*) : afin de traiter le fourchement de contexte, un contexte est identifié par une paire d'ULID et un identifiant de contexte fourché. Le contexte par défaut a un FII de zéro.

Contact initial : on utilise ce terme pour se référer à la communication pré-Shim quand un ULP décide de commencer à communiquer avec un homologue en envoyant et recevant des paquets ULP. Normalement, cela ne va invoquer aucune opération dans le Shim, car le Shim peut différer l'établissement de contexte jusqu'à un instant arbitraire ultérieur.

Adresse fondée sur le hachage (HBA, *Hash-Based Address*) : forme d'adresse IPv6 où l'identifiant d'interface est déduit d'un hachage cryptographique de tous les préfixes alloués à l'hôte. Voir la [RFC5535].

Adresse générée cryptographiquement (CGA, *Cryptographically Generated Address*) : forme d'adresse IPv6 où l'identifiant d'interface est déduit d'un hachage cryptographique de la clé publique. Voir la [RFC3972].

Structure de données de paramètre (PDS, *Parameter Data Structure*) de CGA : informations que CGA et HBA échangent afin d'informer l'homologue de la façon dont l'identifiant d'interface a été calculé. Voir les [RFC3972] et [RFC5535].

2.2 Conventions de notation

A, B, et C sont des hôtes. X est un hôte potentiellement malveillant.

FQDN(A) est le nom de domaine pleinement qualifié de A.

Ls(A) est l'ensemble de localisateurs pour A, qui consiste en les localisateurs L1(A), L2(A), ... Ln(A). L'ensemble de localisateurs n'est pas ordonné d'une façon particulière autre que peut-être ce qui est retourné par le DNS. Un hôte pourrait former des ensembles de localisateurs différents contenant des sous réseaux différents d'adresses IP de l'hôte. C'est nécessaire dans certains cas pour des raisons de sécurité. Voir le paragraphe 16.1.

ULID(A) est un identifiant de couche supérieure pour A. Dans cette proposition, ULID(A) est toujours un membre de l'ensemble de localisateurs de A.

CT(A) est une étiquette de contexte allouée par A.

État se réfère à l'état spécifique de l'automate à états décrit au paragraphe 6.2

2.3 Variables conceptuelles

Le présent document utilise aussi des variables conceptuelles internes pour décrire le comportement du protocole et des variables externes qu'une mise en œuvre doit permettre aux administrateurs du système de changer. Les noms de variables spécifiques, comment leurs valeurs changent, et comment leur réglage influence le comportement du protocole sont fournis pour montrer le comportement du protocole. Une mise en œuvre n'est pas obligée de les avoir sous la forme exacte décrite ici, pour autant que son comportement externe est cohérent avec celui décrit dans ce document. Voir à la Section 6 une description de la structures des données conceptuelles.

3. Hypothèses

L'intention de la conception est de s'assurer que le protocole Shim6 est capable de traiter les défaillances de chemin indépendamment du nombre des adresses IP (localisateurs) disponibles aux deux hôtes communicants, et indépendamment de l'hôte qui détecte la condition de défaillance.

Considérons, par exemple, le cas dans lequel A et B ont tous deux un état Shim6 actif et où A a seulement un localisateur tandis que B a plusieurs localisateurs. Dans ce cas, il se pourrait que B essaye d'envoyer un paquet à A, et ait détecté une condition de défaillance avec la paire actuelle de localisateurs. Comme B a plusieurs localisateurs, il a probablement plusieurs FAI, et (par conséquent) a probablement des chemins de sortie de remplacement vers A. B ne peut pas faire varier l'adresse de destination (c'est-à-dire, le localisateur de A) car A a seulement un localisateur. Cependant, B peut avoir besoin de faire varier l'adresse de source afin d'assurer la livraison du paquet.

Dans de nombreux cas, le fonctionnement normal de l'acheminement IP peut être cause que les paquets suivent un chemin vers la sortie correcte (actuellement opérationnelle). Dans certains cas, il est possible qu'un chemin puisse être choisi sur la base de l'adresse de source, ce qui implique que B va devoir choisir une adresse de source correspondant à la sortie actuellement en fonction. Les détails de comment l'acheminement peut être réalisé sortent du domaine d'application de ce document.

Aussi, quand le FAI du site effectue le filtrage d'entrée sur la base des adresses de source des paquets, Shim6 suppose que les paquets envoyés avec des combinaisons différentes de source et destination ont des chances raisonnables de passer par les filtre d'entrée du FAI pertinent. Cela peut être réalisé de plusieurs façons (qui sortent toutes du domaine d'application de ce document) comme d'avoir les FAI qui relâchent leurs filtres d'entrée ou de choisir la sortie de telle façon qu'elle corresponde au préfixe IP de l'adresse de source. Dans le cas où un chemin de sortie est défaillant mais où un autre fonctionne correctement, il peut être nécessaire que la source du paquet (le nœud B du paragraphe précédent) choisisse une

adresse de source qui corresponde à la sortie opérationnelle, afin de passer les filtres d'entrée du FAI.

L'approche Shim6 suppose qu'il n'y a pas de NAT IPv6 à IPv6 sur les chemins, c'est-à-dire, que les deux extrémités peuvent échanger leur propre notion de leurs adresses IPv6 et que ces adresses vont aussi avoir un sens pour leur homologue.

La sécurité du protocole Shim6 repose sur l'usage des adresses fondées sur le hachage (HBA, *Hash-Based Address*) [RFC5535] et/ou des adresses générées cryptographiquement (CGA, *Cryptographically Generated Address*) [RFC3972]. Dans le cas d'utilisation de HBA, toutes les adresses allouées à l'hôte qui sont incluses dans le protocole Shim6 (comme localisateur ou comme ULID) doivent faire partie du même ensemble de HBA. Dans le cas d'utilisation de CGA, l'adresse utilisée comme ULID doit être une CGA, mais les autres adresses qui sont utilisées comme localisateurs n'ont pas besoin d'être des CGA ou des HBA. On devrait noter qu'il est parfaitement acceptable de faire fonctionner le protocole Shim6 entre un hôte qui a plusieurs localisateurs et un autre hôte qui a une seule adresse IP. Dans ce cas, l'adresse de l'hôte avec une seule adresse n'a pas besoin d'être une HBA ou une CGA.

4. Vue d'ensemble du protocole

Le protocole Shim6 opère en plusieurs phases dans le temps. La séquence suivante illustre les concepts :

- Une application sur l'hôte A décide de contacter une application sur l'hôte B en utilisant un protocole de couche supérieure. Il en résulte que l'ULP sur l'hôte A envoie des paquets à l'hôte B. On appelle cela le contact initial. En supposant que les adresses IP choisies par le choix d'adresse par défaut [RFC3484] et ses extensions [9] fonctionnent, il n'y a alors pas d'action pour l'ajustement à ce moment. Tout établissement du contexte Shim peut être différé.

- Une heuristique sur A ou B (ou les deux) détermine qu'il est approprié de subir les frais de Shim6 pour rendre cette communication d'hôte à hôte robuste contre les défaillances de localisateur. Par exemple, cette heuristique pourrait être que plus de 50 paquets ont été envoyés ou reçus, ou qu'il y a eu l'expiration d'un temporisateur pendant que l'échange de paquets avait lieu. Cela fait que Shim initiait l'échange d'établissement de contexte en quatre phases. L'objet de cette heuristique est d'éviter d'établir un contexte de Shim quand seulement un petit nombre de paquets est échangé entre deux hôtes.

  Par suite de cet échange, A et B vont tous deux connaître une liste de localisateurs l'un de l'autre.

  Si l'échange d'établissement de contexte échoue, l'initiateur va alors savoir que l'autre extrémité ne prend pas en charge Shim6, et va continuer avec le comportement standard (non Shim6) pour la session.

- La communication se continue sans aucun changement pour les paquets d'ULP. En particulier, il n'y a pas d'en-tête Extension Shim6 ajouté aux paquets ULP, car la paire d'ULID est la même que la paire de localisateurs. De plus, il pourrait y avoir des messages échangés entre les sous couches Shim pour la détection d'(in)accessibilité.

- À un moment, quelque chose échoue. Selon l'approche de la détection d'accessibilité, il pourrait y avoir un avis de l'ULP, ou la détection d'(in)accessibilité Shim pourrait découvrir qu'il y a un problème.

  À ce moment, une ou les deux extrémités de la communication ont besoin de sonder les différentes paires de localisateurs de remplacement jusqu'à trouver une paire qui fonctionne, et de passer à l'utilisation de cette paire de localisateurs.

- Une fois qu'une paire de localisateurs de remplacement qui fonctionne a été trouvée, Shim va réécrire les paquets en émission et étiqueter les paquets avec l'en-tête Extension de charge utile Shim6, qui contient l'étiquette de contexte du receveur. Le receveur va utiliser l'étiquette de contexte pour trouver l'état du contexte, qui va indiquer quelles adresses placer dans l'en-tête IPv6 avant de passer le paquet à l'ULP. Le résultat est que, du point de vue de l'ULP, le paquet passe non modifié de bout en bout, même si l'infrastructure d'acheminement IP envoie le paquet à un localisateur différent.

- La détection d'(in)accessibilité Shim va surveiller la nouvelle paire de localisateurs comme elle surveillait la paire de localisateurs originale, afin de pouvoir détecter des défaillances ultérieures.

- En plus des défaillances détectées sur la base des observations de bout en bout, un point d'extrémité pourrait tenir pour certain qu'un ou plusieurs de ses localisateurs ne fonctionnent pas. Par exemple, l'interface réseau pourrait être défaillante ou arrêtée (à la couche 2) ou une adresse IPv6 pourrait être devenue déconseillée ou invalide. Dans un tel

cas, l'hôte peut signaler à son homologue qu'essayer cette adresse n'est plus recommandé. Cela déclenche quelque chose de similaire au traitement d'une défaillance, et une nouvelle paire de localisateurs active doit être trouvée.

Le protocole a aussi la capacité d'exprimer d'autres formes de préférences de localisateur. Un changement de toute préférence peut être signalé à l'homologue, ce qui fera que l'homologue enregistre les nouvelles préférences. Un changement des préférences pourrait facultativement faire que l'homologue veuille utiliser une paire de localisateurs différente. Dans ce cas, l'homologue suit les mêmes procédure de changement de localisateurs qu'après une défaillance (en vérifiant que son homologue est bien présent au localisateur de remplacement, etc).

o Quand le Shim pense que l'état du contexte n'est plus utilisé, il peut éliminer l'état collecté ; il n'y a pas de coordination nécessaire avec l'homologue hôte avant la suppression de l'état. Il y a un message de récupération défini pour être capable de signaler quand il n'y a pas d'état de contexte, qui peut être utilisé pour détecter et récupérer aussi bien de l'élimination prématurée de l'état collecté que de la perte d'état complète (panne et réamorçage) d'un homologue.

Le mécanisme exact pour déterminer quand l'état du contexte n'est plus utilisé dépend de la mise en œuvre. Par exemple, une mise en œuvre pourrait utiliser l'existence de l'état d'ULP (lorsque il est connu de la mise en œuvre) comme une indication que l'état est encore utilisé, combinée avec un temporisateur (pour traiter l'état de l'ULP qui pourrait n'être pas connu de la sous couche Shim) pour déterminer quand l'état ne va probablement plus être utilisé.

Note : Les paquets d'ULP dans Shim6 peuvent être portés complètement non modifiés tant que la paire d'ULID est utilisée comme paire de localisateurs. Après un passage à une paire différente de localisateurs, les paquets sont "étiquetés" avec un en-tête Extension Shim6 afin que le receveur puisse toujours déterminer le contexte auquel ils appartiennent. Ceci est accompli en incluant un en-tête Extension de charge utile Shim6 de 8 octets avant les en-têtes (d'extension) qui sont traités par la sous couche IP de point d'extrémité et les ULP. Si, ensuite, les ULID originaux sont choisis comme paire de localisateurs active, alors l'étiquetage des paquets avec l'en-tête Extension Shim6 n'est plus nécessaire.

4.1 Étiquettes de contexte

Un contexte entre deux hôtes est en fait un contexte entre deux ULID. Le contexte est identifié par une paire d'étiquettes de contexte. Chaque extrémité doit allouer une étiquette de contexte, et une fois que le contexte est établi, la plupart des messages de contrôle Shim6 contiennent l'étiquette de contexte que le receveur du message a allouée. Donc, au minimum, la combinaison de <homologue ULID, ULID local, étiquette de contexte local> doit identifier de façon univoque un contexte. Mais, comme les en-têtes Extension de charge utile Shim6 sont démultiplexés sans regarder les localisateurs dans le paquet, le receveur va devoir allouer des étiquettes de contexte uniques pour tous ses contextes. L'étiquette de contexte est un nombre de 47 bits (le plus grand qui puisse tenir dans un en-tête d'extension de 8 octets) tout en préservant un bit pour différencier les messages de signalisation Shim6 de l'en-tête Shim6 inclus dans les paquets de données, permettant aux deux d'utiliser le même numéro de protocole.

Le mécanisme pour détecter une perte d'état de contexte chez l'homologue suppose que le receveur peut dire quels paquets ont besoin d'avoir le localisateur réécrit, même après qu'il a perdu tout état (par exemple, du fait d'une panne suivie d'un réamorçage). Ceci est réalisé parce que, après un événement de re-rattachement, les paquets qui ont besoin de réécrire le côté receveur portent l'en-tête Extension de charge utile Shim6.

4.2 Fourchement de contexte

Il a été affirmé qu'il serait important pour les futurs ULP -- en particulier, les futurs protocoles de transport -- d'être capables de contrôler quelles paires de localisateurs sont utilisées pour les différentes communications. Par exemple, l'hôte A et l'hôte B pourraient communiquer en utilisant tous deux le trafic de voix sur IP (VoIP, *Voice over IP*) et le trafic ftp, et ces communications pourraient bénéficier de l'utilisation de différentes paires de localisateurs. Cependant, le mécanisme Shim6 de base utilise une seule paire de localisateurs courante pour chaque contexte ; donc, un seul contexte ne peut pas accomplir cela.

Pour cette raison, le protocole Shim6 prend en charge la notion de fourchement de contexte. C'est un mécanisme par lequel un ULP peut spécifier (en utilisant une API non encore définie) qu'un contexte, par exemple, la paire d'ULID <A1, B2>, devrait être fourchée en deux contextes. Dans ce cas, le contexte fourché va recevoir un identifiant d'instance fourché différent de zéro, tandis que le contexte par défaut a un FII de zéro.

L'identifiant d'instance fourché (FII, *Forked Instance Identifier*) est un identifiant de 32 bits qui n'a pas de signification dans le protocole autre que de faire partie du triplet qui identifie le contexte. Par exemple, un hôte pourrait allouer des FII

comme numéros à la suite pour toute paire d'ULID donnée.

Aucune autre considération particulière n'est nécessaire dans le protocole Shim6 pour traiter les contextes fourchés.

Noter que le fourchement tel que spécifié NE permet PAS à A d'être capable de dire à B qu'un certain trafic (un quintuplet ?) devrait être fourché dans la direction inverse. Le mécanisme de fourchement Shim6 spécifié s'applique seulement à l'envoi de paquets d'ULP. Si un ULP veut fourcher dans les deux directions, il appartient à cet ULP d'établir cela et de donner des instructions au Shim à chaque extrémité de transmission en utilisant le contexte fourché.

4.3 Extensions d'API

Plusieurs extensions d'API ont été discutées pour Shim6, mais leur spécification réelle sort du domaine d'application du présent document. La plus simple serait d'ajouter une option de prise pour être capable de faire que le trafic outrepasse le Shim (de ne pas créer d'état et de ne pas utiliser d'état créé par d'autre trafic). Cela pourrait être une option de prise IPV6_DONTSHIM. Une telle option serait utile pour des protocoles, comme le DNS, où l'application a son propre mécanisme de reprise sur défaillance (plusieurs enregistrements NS dans le cas du DNS) et l'utilisation de Shim pourrait ajouter une latence supplémentaire sans aucun avantage.

D'autres extensions d'API sont discutées à l'Appendice A. Les extensions d'API actuelles sont définies dans la [RFC6316].

4.4 Sécurisation de Shim6

Les mécanismes sont sécurisés en utilisant une combinaison de techniques :

- La technique HBA [RFC5535] pour vérifier que les localisateurs empêchent un attaquant de rediriger le flux de paquets sur une autre destination.

- Exiger un sondage + réponse d'accessibilité (défini dans la [RFC5534]) avant qu'un nouveau localisateur soit utilisé comme destination, afin d'empêcher un tiers de lancer des attaques d'inondation.

- Le premier message ne crée aucun état chez le répondeur. Essentiellement, un échange en trois phases est exigé avant que le répondeur crée un état. Cela signifie que une attaque de déni de service fondée sur l'état (essayant d'utiliser toute la mémoire chez le répondeur) fournit au moins une adresse IPv6 que l'attaquant utilise.

- Les messages d'établissement de contexte utilisent des noms occasionnels pour empêcher les attaques en répétition et pour empêcher des attaquants hors chemin d'interférer avec l'établissement.

- Chaque message de contrôle du protocole Shim6, après l'établissement de contexte, porte l'étiquette de contexte allouée au contexte particulier. Cela implique qu'un attaquant a besoin de découvrir cette étiquette de contexte avant d'être capable d'usurper un message de contrôle Shim6. Une telle découverte exige probablement que tout attaquant potentiel soit le long du chemin afin de renifler la valeur de l'étiquette de contexte. Il en résulte que par cette technique, le protocole Shim6 est protégé contre les attaquants hors chemin.

4.5 Vue d'ensemble des messages de commande de Shim

L'établissement de contexte Shim6 est réalisé en utilisant quatre messages ; I1, R1, I2, R2. Normalement, ils sont envoyés dans cet ordre de l'initiateur et du répondeur, respectivement. Si les deux extrémités tentent d'établir l'état de contexte en même temps (pour la même paire d'ULID) alors leurs messages I1 pourraient se croiser en vol, et résulter en un message R2 immédiat. (Les noms de ces messages sont empruntés à HIP [RFC5201].)

Les messages R1bis et I2bis sont définis ; ils sont utilisés pour récupérer un contexte après qu'il a été perdu. Un message R1bis est envoyé quand un en-tête Extension de contrôle Shim6 ou de charge utile Shim6 arrive et qu'il n'y a pas de correspondance d'état de contexte chez le receveur. Quand un tel message est reçu, il va résulter en la re-création du contexte Shim6 en utilisant les messages I2bis et R2.

Les listes de localisateurs des homologues sont normalement échangées au titre de l'échange d'établissement de contexte. Mais l'ensemble de localisateurs pourrait être dynamique. Pour cette raison, il y a des messages de demande de mise à jour et d'accusé de réception de mise à jour ainsi qu'une option Liste de localisateurs.

Même quand la liste de localisateurs est fixe, un hôte pourrait déterminer que des préférences pourraient avoir changé. Par

exemple, il pourrait déterminer qu'il y a une défaillance visible localement qui implique que certains localisateurs ne sont plus utilisables. Cela utilise une option Préférences de localisateur dans le message Demande mise à jour.

Le mécanisme pour la détection d'(in)accessibilité est appelé la communication bidirectionnelle forcée (FBD, *Forced Bidirectional Communication*). FBD utilise un message Keepalive *(maintien en vie)* qui est envoyé quand un hôte a reçu des paquets de son homologue mais n'a pas encore envoyé de paquet de son ULP à l'homologue. Le type de message est réservé dans le présent document, mais le format de message et les règles de traitement sont spécifiées dans la [RFC5534].

De plus, quand le contexte est établi et qu'il y a ensuite une défaillance, il doit y avoir un moyen de sonder l'ensemble de paires de localisateurs pour trouver efficacement une paire qui fonctionne. Le présent document réserve un type de message Probe *(sonde)* avec le format de paquet et les règles de traitement spécifiés dans la [RFC5534].

Les messages Probe et Keepalive ci-dessus supposent qu'on a établi un contexte de paire d'ULID. Cependant, la communication pourrait échouer durant le contact initial (c'est-à-dire, quand l'application ou le protocole de transport essaye d'établir une communication). Ceci est traité en utilisant les mécanismes de l'ULP pour essayer différentes paires d'adresses comme spécifié dans la [RFC3484] et [9]. Dans les futures versions du protocole, et avec une API plus riche entre l'ULP et le Shim, le Shim pourrait être capable d'aider à optimiser la découverte d'une paire de localisateurs qui fonctionne durant le contact initial. C'est pour étude ultérieure.

4.6 Ordre des en-têtes d'extension

Comme l'ajustement est placé entre la sous couche de point d'extrémité IP et la sous couche d'acheminement IP, l'en-tête Shim va être placé avant tout en-tête Extension de point d'extrémité (en-têtes de fragmentation, en-tête Options de destination, AH, ESP) mais après tout en-tête relatif à l'acheminement (en-tête d'extensions bond par bond, en-tête d'acheminement, et en-tête Options de destinations, qui précède un en-tête d'acheminement). Quand le tunnelage est utilisé, que ce soit le tunnelage IP dans IP ou la forme spéciale de tunnelage qu'utilise IPv6 mobile (avec les options d'adresse de rattachement et le type 2 d'en-tête d'acheminement) il y a un choix de si l'ajustement s'applique dans le tunnel ou en dehors du tunnel, ce qui affecte la localisation de l'en-tête Shim6.

Dans la plupart des cas, les tunnels IP dans IP sont utilisés comme technique d'acheminement ; donc, il y a du sens à les appliquer sur les localisateurs, ce qui signifie que l'envoyeur va insérer l'en-tête Shim6 après toute encapsulation IP dans IP. C'est ce qui se passe naturellement quand les routeurs appliquent l'encapsulation IP dans IP. Donc, les paquets vont avoir :
- o l'en-tête IP externe
- o l'en-tête IP interne
- o l'en-tête Extension Shim6 (si nécessaire)
- o l'ULP

Mais l'ajustement peut aussi être utilisé pour créer des "tunnels Shim", c'est-à-dire, où un tunnel IP dans IP utilise l'ajustement pour être capable de commuter les adresses de point d'extrémité de tunnel entre différents localisateurs. Dans ce cas, les paquets auraient :
- o l'en-tête IP externe
- o l'en-tête Extension Shim6 (si nécessaire)
- o l'en-tête IP interne
- o l'ULP

Dans tous les cas, le comportement du receveur est bien défini ; un receveur traite les en-têtes Extension dans l'ordre. Cependant, l'interaction précise entre IPv6 mobile et Shim6 sera étudiée ultérieurement ; il pourrait y avoir du sens à ce que IPv6 mobile opère aussi sur les localisateurs, ce qui signifie que Shim serait mis en couche par dessus le mécanisme MIPv6.

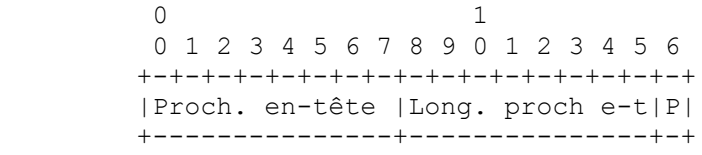
5. Formats de message

Les messages Shim6 sont tous portés en utilisant un nouveau numéro de protocole IP (140). Les messages Shim6 ont un en-tête commun (défini ci-dessous) avec des champs fixes, suivis par des champs spécifiques du type.

Les messages Shim6 sont structurés comme un en-tête Extension IPv6 car l'en-tête Extension de charge utile Shim6 est utilisé pour porter les paquets d'ULP après une commutation de localisateur. Les messages de contrôle Shim6 utilisent les mêmes formats d'en-tête d'extension de sorte qu'un seul "numéro de protocole" doit être permis à travers les pare-feu afin que Shim6 fonctionne à travers le pare-feu.

5.1 Format commun de message Shim6

Les 17 premiers bits de l'en-tête Shim6 sont communs pour l'en-tête Extension de charge utile Shim6 et pour les messages de contrôle. Il est comme suit :



Prochain en-tête : charge utile qui suit cet en-tête.

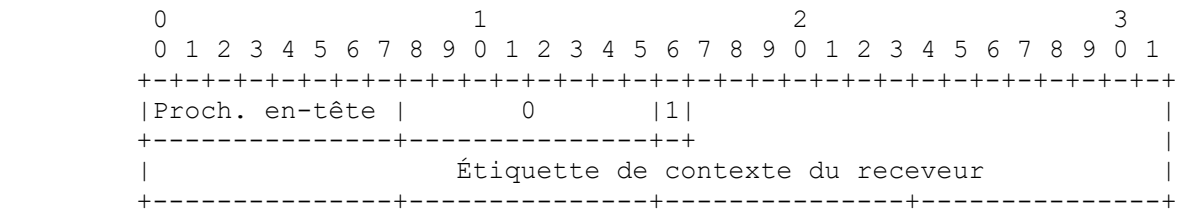
Longueur de prochain en-tête : entier non signé de 8 bits. Longueur de l'en-tête Shim6 en unités de 8 octets, non inclus les 8 premiers octets.

P : un seul bit pour distinguer les en-têtes d'extension de charge utile Shim6 des messages de contrôle.

Les paquets de signalisation Shim6 ne peuvent pas être de plus de 1280 octets, incluant l'en-tête IPv6 et tous les en-têtes intermédiaires entre l'en-tête IPv6 et l'en-tête Shim6. Une façon de satisfaire cette exigence est d'omettre une partie des informations d'adresse de localisateur si, avec ces informations incluses, le paquet ferait plus de 1280 octets. Une autre option est d'effectuer une ingénierie d'option, en divisant entre différents messages Shim6 les informations à transmettre. Une mise en œuvre peut imposer des restrictions administratives pour éviter des paquets Shim6 excessivement grands, comme une limitation du nombre de localisateurs à utiliser.

5.2 Format d'en-tête d'extension de charge utile Shim6

L'en-tête Extension de charge utile Shim6 est utilisé pour porter des paquets ULP où le receveur doit remplacer le contenu des champs Source et/ou Destination dans l'en-tête IPv6 avant de passer le paquet à l'ULP. Donc, cet en-tête d'extension est exigé quand la paire de localisateurs utilisée n'est pas la même que la paire d'ULID.



Prochain en-tête : charge utile qui suit cet en-tête.

Longueur d'extension d'en-tête : 0 (car l'en-tête fait 8 octets).

P : réglé à un. Un seul bit pour distinguer ceci des messages de contrôle Shim6.

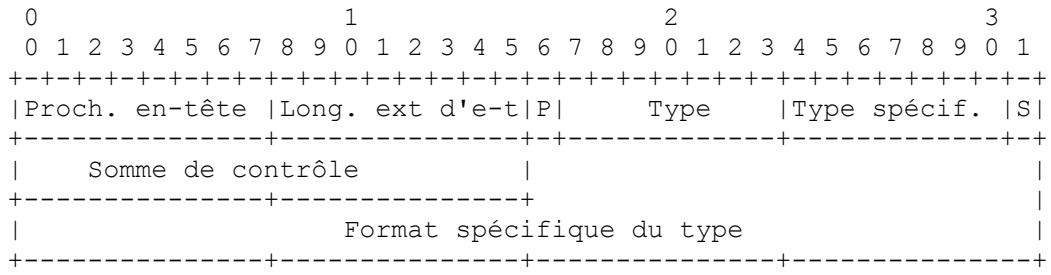
Étiquette de contexte du receveur : entier non signé de 47 bits. Alloué par le receveur pour identifier le contexte.

5.3 En-tête commun de commande Shim6

La partie commune de l'en-tête a un champ Prochain en-tête et un champ Longueur d'extension d'en-tête qui sont cohérents avec les autres en-têtes d'extension IPv6, même si la valeur de Prochain en-tête est toujours "NO NEXT HEADER" pour les messages de commande.

Les en-têtes Shim6 doivent être un multiple de 8 octets ; donc, la taille minimum est 8 octets.

L'en-tête commun de message de commande Shim6 est le suivant :



Prochain en-tête : sélecteur de 8 bits. Normalement réglé à NO_NXT_HDR (59).

Longueur d'extension d'en-tête : entier non signé de 8 bits. Longueur de l'en-tête Shim6 en unités de 8 octets, non inclus les 8 premiers octets.

P : réglé à zéro. Un seul bit pour distinguer cela de l'en-tête Extension de charge utile Shim6.

Type : entier non signé de 7 bits. Identifie le message actuel dans le tableau ci-dessous. Les codes de type de 0 à 63 ne vont pas déclencher de messages R1bis sur un contexte manquant, tandis que les codes 64 à 127 vont déclencher R1bis.

S : un seul bit réglé à zéro qui permet à Shim6 et HIP d'avoir un format d'en-tête commun tout en distinguant entre messages Shim6 et HIP.

Somme de contrôle : entier non signé de 16 bits. La somme de contrôle est le complément à un de 16 bits de la somme des compléments à un de l'en-tête entier de message Shim6, débutant au champ Prochain en-tête Shim6 et se terminant comme indiqué par la longueur d'en-tête d'extension. Donc, quand il y a une charge utile à la suite de l'en-tête Shim6, la charge utile N'EST PAS incluse dans la somme de contrôle Shim6. Noter que, à la différence de protocoles comme ICMPv6, il n'y a pas de partie somme de contrôle de pseudo en-tête de la somme de contrôle ; cela donne au localisateur de la souplesse sans avoir à changer la somme de contrôle.

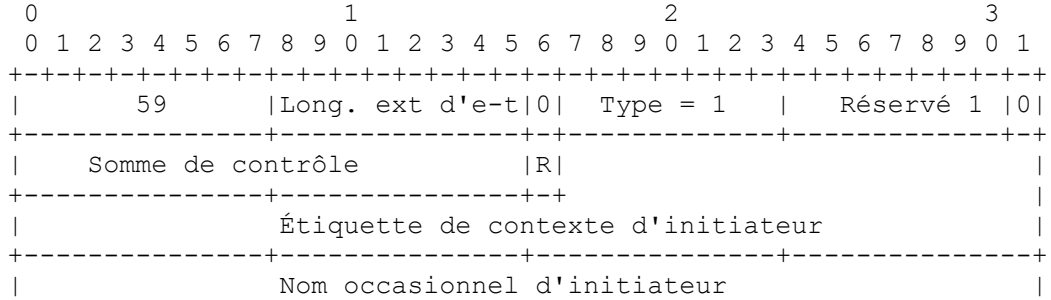
Type spécifique : partie du message qui est différente pour les différents types de messages.

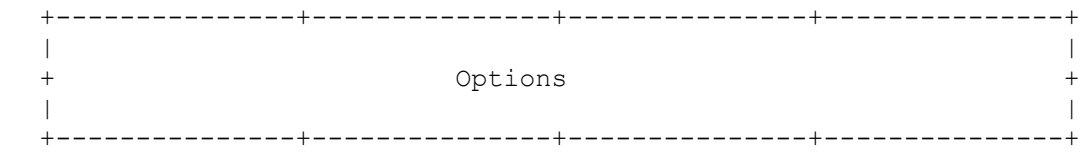
Type	**Message**
1	I1 (premier message d'établissement provenant de l'initiateur)
2	R1 (premier message d'établissement provenant du répondeur)
3	I2 (second message d'établissement provenant de l'initiateur)
4	R2 (second message d'établissement provenant du répondeur)
5	R1bis (réponse pour faire référence à un contexte non existant)
6	I2bis (réponse à un message R1bis)
64	Demande de mise à jour
65	Accusé de réception de mise à jour
66	Maintien en vie *(Keepalive)*
67	Message de sonde *(Probe)*
68	Message d'erreur *(Error)*

Tableau 1

5.4 Format de message I1

Le message I1 est le premier message dans l'échange d'établissement de contexte.





Prochain en-tête : NO_NXT_HDR (59).

Longueur d'extension d'en-tête : au moins 1, car l'en-tête fait 16 octets quand il n'y a pas d'options.

Type : 1

Réservé 1 : champ de 7 bits. Réservé pour utilisation future. Zéro à l'émission. DOIT être ignoré à réception.

R : champ de 1 bit. Réservé pour utilisation future. Zéro à l'émission. DOIT être ignoré à réception.

Étiquette de contexte d'initiateur : champ de 47 bits. L'étiquette de contexte que l'initiateur a alloué pour le contexte.

Nom occasionnel d'initiateur : entier non signé de 32-bits. Nombre aléatoire pris par l'initiateur, que le répondeur va retourner dans le message R1.

Les options suivantes sont définies pour ce message :

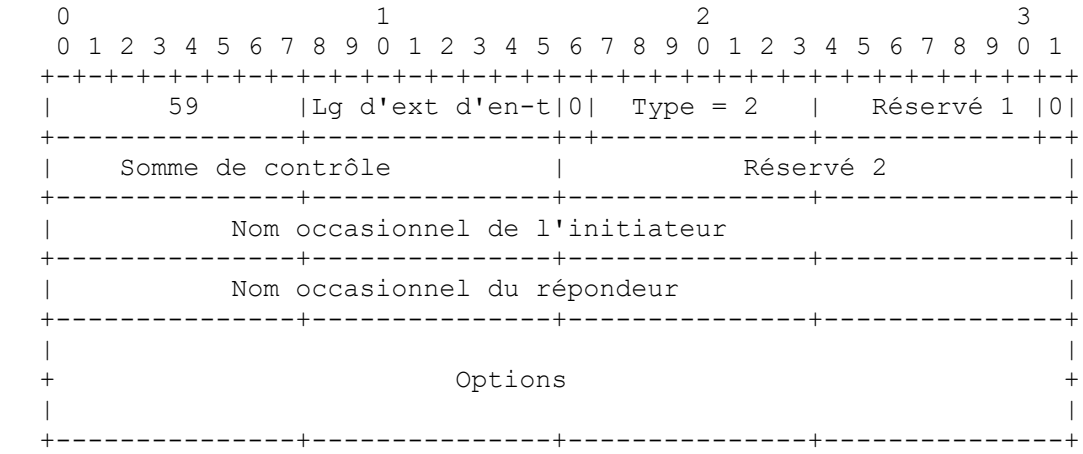
Paire d'ULID : quand les adresses IPv6 de source et de destination dans l'en-tête IPv6 ne correspondent pas à la paire d'ULID, cette option DOIT être incluse. Un exemple est quand on récupère d'un contexte perdu.

Identifiant d'instance fourchée : quand une autre instance d'un contexte existant avec la même paire d'ULID est créé, une option Identifiant d'instance fourchée DOIT être incluse pour distinguer cette nouvelle instance de l'existante.

De futures extensions de protocole pourraient définir des options supplémentaires pour ce message. Le bit C dans le format d'option définit comment une telle nouvelle option va être traitée par une mise en œuvre. Voir le paragraphe 5.15.

5.5 Format de message R1

Le message R1 est le second message dans l'échange d'établissement de contexte. Le répondeur envoie cela en réponse à un message I1, sans créer d'état spécifique chez l'initiateur.



Prochain en-tête : NO_NXT_HDR (59).

Longueur d'extension d'en-tête : au moins 1, car l'en-tête fait 16 octets quand il n'y a pas d'options.

Type : 2

Réservé 1 : champ de 7 bits. Réservé pour utilisation future. Zéro à l'émission. DOIT être ignoré à réception.

Réservé 2 : champ de 16 bits. Réservé pour utilisation future. Zéro à l'émission. DOIT être ignoré à réception.

Nom occasionnel de l'initiateur : entier non signé de 32 bits. Copié du message I1.

Nom occasionnel du répondeur : entier non signé de 32 bits. Nombre choisi par le répondeur, que l'initiateur va retourner dans le message I2.

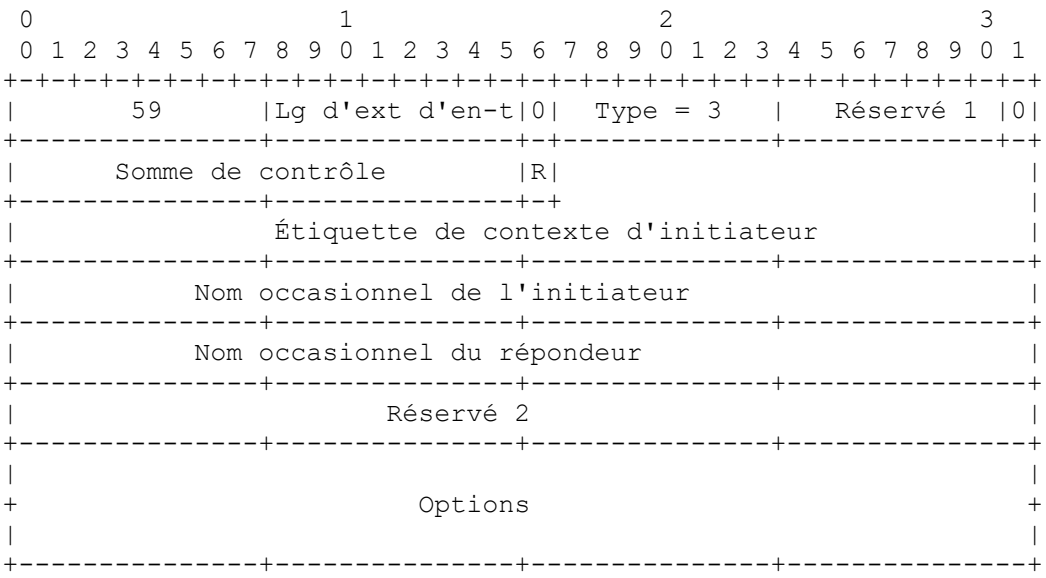
Les options suivantes sont définies pour ce message :

Valideur de répondeur : option de longueur variable. Cette option DOIT être incluse dans le message R1. Normalement, elle contient un hachage généré par le répondeur, qu'il utilise avec la valeur de nom occasionnel du répondeur pour vérifier qu'un message I2 est bien envoyé en réponse à un message R1, et que les paramètres dans le message I2 sont les mêmes que dans le message I1.

De futures extensions de protocole pourraient définir des options supplémentaires pour ce message. Le bit C dans le format d'option définit comment une telle nouvelle option va être traitée par une mise en œuvre. Voir le paragraphe 5.15.

5.6 Format de message I2

Le message I2 est le troisième message dans l'échange d'établissement de contexte. L'initiateur envoie cela en réponse à un message R1, après la vérification du nom occasionnel de l'initiateur, etc.



Prochain en-tête : NO_NXT_HDR (59).

Longueur d'extension d'en-tête : au moins 2, car l'en-tête fait 24 octets quand il n'y a pas d'options.

Type : 3

Réservé 1 : champ de 7 bits. Réservé pour utilisation future. Zéro à l'émission. DOIT être ignoré à réception.

R : champ de 1 bit. Réservé pour utilisation future. Zéro à l'émission. DOIT être ignoré à réception.

Étiquette de contexte d'initiateur : champ de 47 bits. L'étiquette de contexte que l'initiateur a alloué pour le contexte.

Nom occasionnel de l'initiateur : entier non signé de 32 bits. Nombre aléatoire pris par l'initiateur, que le répondeur va retourner dans le message R2.

Nom occasionnel du répondeur : entier non signé de 32 bits. Copié du message R1.

Réservé 2 : champ de 32 bits. Réservé pour utilisation future. Zéro à l'émission. DOIT être ignoré à réception. (Nécessaire pour faire que les options commencent sur une limite d'un multiple de 8 octets.)

Les options suivantes sont définies pour ce message :

Valideur de répondeur : option de longueur variable. Cette option DOIT être incluse dans le message I2 et DOIT être générée en copiant l'option Valideur de répondeur reçue dans le message R1.

Paire d'ULID : quand les adresses IPv6 de source et de destination dans l'en-tête IPv6 ne correspondent pas à la paire d'ULID, cette option DOIT être incluse. Un exemple est quand on récupère d'un contexte perdu.

Identifiant d'instance fourchée : quand une autre instance d'un contexte existant avec la même paire d'ULID est créée, une option Identifiant d'instance fourchée DOIT être incluse pour distinguer cette nouvelle instance de l'existante.

Liste de localisateurs : envoyée facultativement quand l'initiateur veut immédiatement dire au répondeur sa liste de localisateurs. Quand elle est envoyée, les informations de HBA/CGA nécessaires pour vérifier la liste de localisateurs DOIVENT aussi être incluses.

Préférences de localisateurs : envoyée facultativement quand les localisateurs n'ont pas tous les mêmes préférences.

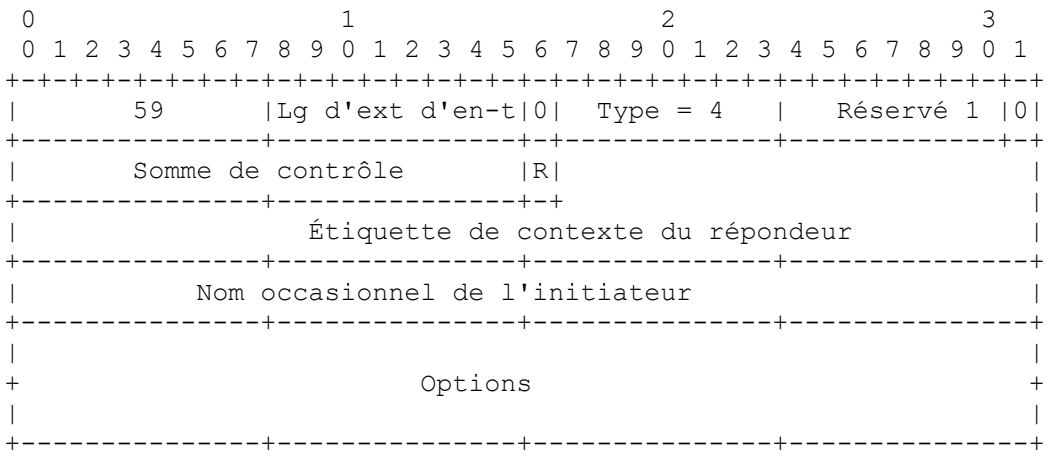
Structure de données de paramètres de CGA : cette option DOIT être incluse dans le message I2 quand la liste de localisateurs est incluse afin que le receveur puisse vérifier la liste de localisateurs.

Signature de CGA : cette option DOIT être incluse dans le message I2 quand certains des localisateurs de la liste utilisent des CGA (et non des HBA) pour la vérification.

De futures extensions de protocole pourraient définir des options supplémentaires pour ce message. Le bit C dans le format d'option définit comment une telle nouvelle option va être traitée par une mise en œuvre. Voir le paragraphe 5.15.

5.7 Format de message R2

Le message R2 est le quatrième message dans l'échange d'établissement de contexte. Le répondeur envoie cela en réponse à un message I2. Le message R2 est aussi utilisé quand les deux hôtes envoient des messages I1 en même temps et que les messages I1 se croisent en vol.



Prochain en-tête : NO_NXT_HDR (59).

Longueur d'extension d'en-tête : au moins 1, car l'en-tête fait 16 octets quand il n'y a pas d'options.

Type : 4

Réservé 1: champ de 7 bits. Réservé pour utilisation future. Zéro à l'émission. DOIT être ignoré à réception.

R : champ de 1 bit. Réservé pour utilisation future. Zéro à l'émission. DOIT être ignoré à réception.

Étiquette de contexte du répondeur : champ de 47 bits. L'étiquette de contexte que le répondeur a alloué pour le contexte.

Nom occasionnel d'initiateur : entier non signé de 32 bits. Copié du message I2.

Les options suivantes sont définies pour ce message :

Liste de localisateurs : envoyée facultativement quand le répondeur veut dire immédiatement à l'initiateur sa liste de localisateurs. Quand elle est envoyée, les informations de HBA/CGA nécessaires pour vérifier la liste des localisateurs DOIVENT aussi être incluses.

Préférences de localisateurs : envoyé facultativement quand les localisateurs n'ont pas tous la même préférence.

Structure de données de paramètres de CGA : Incluse quand la liste des localisateurs est incluse afin que le receveur puisse vérifier la liste des localisateurs.

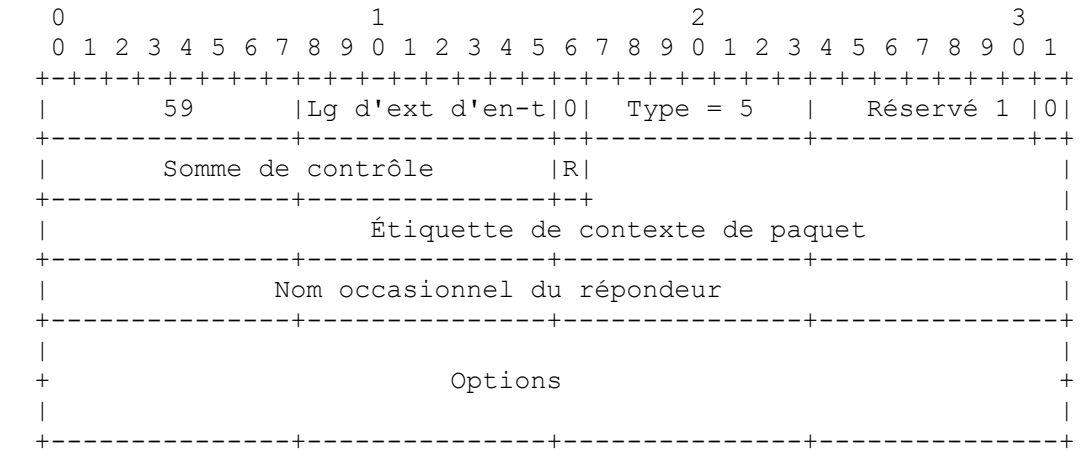
Signature de CGA : Incluse quand certains des localisateurs de la liste utilisent des CGA (et non des HBA) pour la vérification.

De futures extensions de protocole pourraient définir des options supplémentaires pour ce message. Le bit C dans le format d'option définit comment une telle nouvelle option va être traitée par une mise en œuvre. Voir le paragraphe 5.15.

5.8 Format de message R1bis

Si un hôte reçoit un paquet avec un en-tête Extension de charge utile Shim6 ou un message de contrôle Shim6 avec le code de type 64 à 127 (comme un message Update ou Probe) et si l'hôte n'a pas d'état de contexte pour l'étiquette de contexte reçue, il va alors générer un message R1bis.

Ce message permet à l'envoyeur du paquet de se référer au contexte non existant pour rétablir le contexte avec un échange d'établissement de contexte réduit. À réception du message R1bis, le receveur peut procéder au rétablissement du contexte perdu en envoyant directement un message I2bis.



Prochain en-tête : NO_NXT_HDR (59).

Longueur d'extension d'en-tête : au moins 1, car l'en-tête fait 16 octets quand il n'y a pas d'options.

Type : 5

Réservé 1 : champ de 7 bits. Réservé pour utilisation future. Zéro à l'émission. DOIT être ignoré à réception.

R : champ de 1 bit. Réservé pour utilisation future. Zéro à l'émission. DOIT être ignoré à réception.

Étiquette de contexte de paquet : entier non signé de 47 bits. L'étiquette de contexte contenue dans le paquet reçu qui a déclenché la génération du message R1bis.

Nom occasionnel de répondeur : entier non signé de 32 bits. Un nombre pris par le répondeur que l'initiateur va retourner dans le message I2bis.

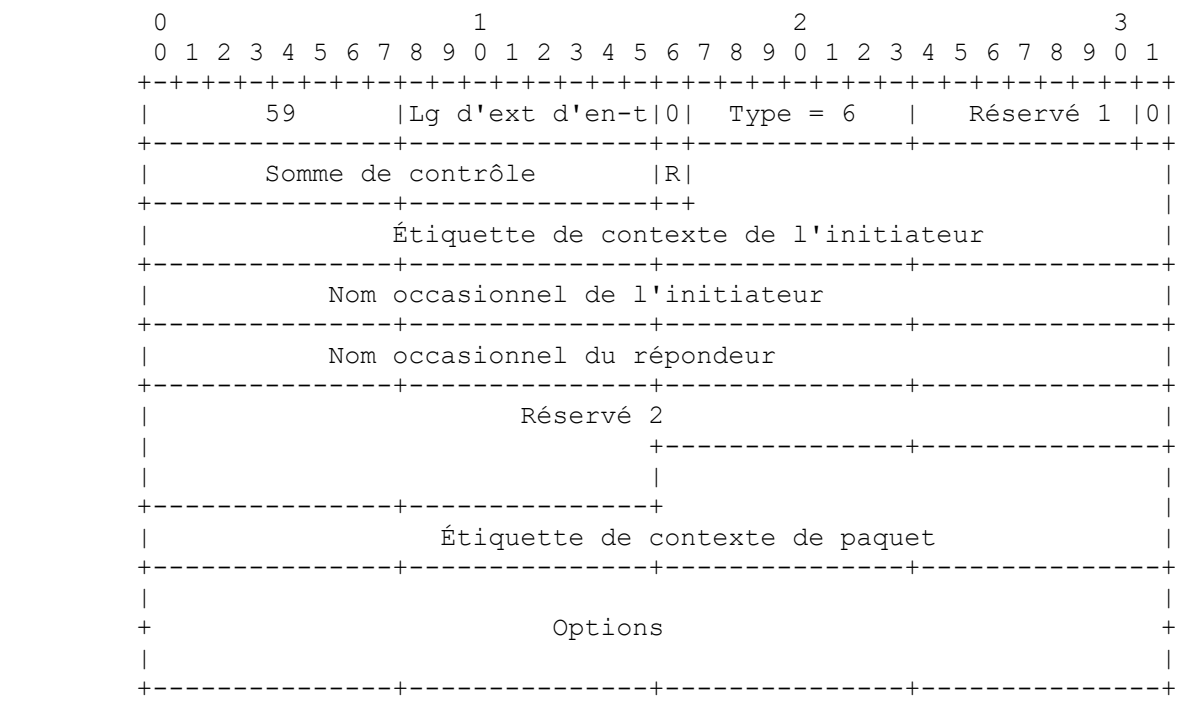
Les options suivantes sont définies pour ce message:

Valideur de répondeur : option de longueur variable. Normalement, un hachage généré par le répondeur, que le répondeur utilise avec la valeur de nom occasionnel de répondeur pour vérifier qu'un message I2bis est bien envoyé en réponse à un message.

De futures extensions de protocole pourraient définir des options supplémentaires pour ce message. Le bit C dans le format d'option définit comment une telle nouvelle option va être traitée par une mise en œuvre. Voir le paragraphe 5.15.

5.9 Format de message I2bis

Le message I2bis est le troisième message dans l'échange de récupération de contexte. Il est envoyé en réponse à un message R1bis, après avoir vérifié que le message R1bis se réfère à un contexte existant, etc.



Prochain en-tête : NO_NXT_HDR (59).

Longueur d'extension d'en-tête : au moins 3, car l'en-tête fait 32 octets quand il n'y a pas d'options.

Type : 6

Réservé 1 : champ de 7 bits. Réservé pour utilisation future. Zéro à l'émission. DOIT être ignoré à réception.

R : champ de 1 bit. Réservé pour utilisation future. Zéro à l'émission. DOIT être ignoré à réception.

Étiquette de contexte d'initiateur : champ de 47 bits. L'étiquette de contexte que l'initiateur a alloué pour le contexte.

Nom occasionnel d'initiateur : entier non signé de 32 bits. Nombre aléatoire pris par l'initiateur, que le répondeur va retourner dans le message R2.

Nom occasionnel de répondeur : entier non signé de 32 bits. Copié du message R1bis.

Réservé 2 : champ de 49 bits. Réservé pour utilisation future. Zéro à l'émission. DOIT être ignoré à réception. (Noter que 17 bits ne sont pas suffisants car les options doivent commencer sur une limite d'un multiple de 8 octets.)

Étiquette de contexte de paquet : entier non signé de 47 bits. Copié du champ Étiquette de contexte de paquet contenue dans le message R1bis reçu.

Les options suivantes sont définies pour ce message :

Valideur de répondeur : option de longueur variable. Juste une copie de l'option Valideur de répondeur dans le message R1bis.

Paire d'ULID : quand les adresses IPv6 de source et de destination dans l'en-tête IPv6 ne correspondent pas à la paire d'ULID, cette option DOIT être incluse.

Identifiant d'instance fourchée : quand une autre instance d'un contexte existant avec la même paire d'ULID est créée, une option Identifiant d'instance fourchée est incluse pour distinguer cette nouvelle instance de celle existante.

Liste de localisateurs : envoyé facultativement quand l'initiateur veut dire immédiatement au répondeur sa liste de localisateurs. Quand elle est envoyée, les informations de HBA/CGA nécessaires pour vérifier la liste des localisateurs DOIVENT aussi être incluses.

Préférences de localisateurs : envoyée facultativement quand les localisateurs n'ont pas tous les mêmes préférences.

Structure de données de paramètres de CGA : incluse quand la liste des localisateurs est incluse afin que le receveur puisse vérifier la liste des localisateurs.

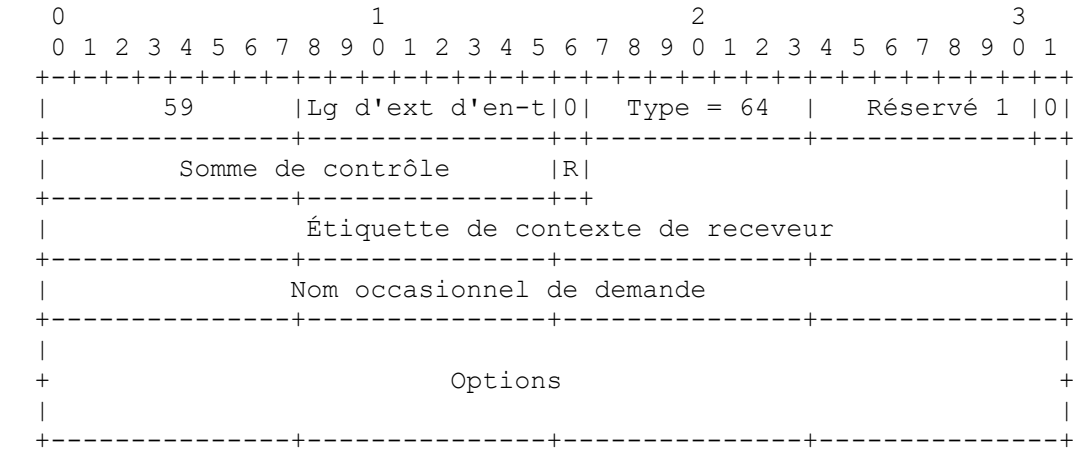
Signature de CGA : incluse quand certains des localisateurs de la liste utilisent des CGA (et pas des HBA) pour la vérification.

De futures extensions de protocole pourraient définir des options supplémentaires pour ce message. Le bit C dans le format d'option définit comment une telle nouvelle option va être traitée par une mise en œuvre. Voir le paragraphe 5.15.

5.10 Format de message demande de mise à jour

Le message Demande de mise à jour est utilisé pour mettre à jour la liste des localisateurs, les préférences de localisateur, ou les deux. Quand la liste des localisateurs est mise à jour, le message contient aussi la ou les options nécessaires pour que les HBA/CGA le sécurisent. La vérification de bonne santé de base qui empêche des attaquants hors chemin de générer des mises à jour boguées est l'étiquette de contexte dans le message.

Le message Demande de mise à jour contient des options (Liste de localisateurs et Préférences de localisateurs) qui, quand elles sont incluses, remplacent complètement respectivement la liste de localisateurs et les préférences de localisateur précédentes. Donc, il n'y a pas de mécanisme pour envoyer juste les différences de la liste des localisateurs.



Prochain en-tête : NO_NXT_HDR (59).

Longueur d'extension d'en-tête : au moins 1, car l'en-tête fait 16 octets quand il n'y a pas d'options.

Type : 64

Réservé 1 : champ de 7 bits. Réservé pour utilisation future. Zéro à l'émission. DOIT être ignoré à réception.

R : champ de 1 bit. Réservé pour utilisation future. Zéro à l'émission. DOIT être ignoré à réception.

Étiquette de contexte de receveur : champ de 47 bits. L'étiquette de contexte que le receveur a alloué pour le contexte.

Nom occasionnel de demande : entier non signé de 32 bits. Nombre aléatoire pris par l'initiateur, que l'homologue va retourner dans le message d'accusé de réception de mise à jour.

Les options suivantes sont définies pour ce message :

Liste de localisateurs : liste des localisateurs (nouveaux) de l'envoyeur. Les localisateurs pourraient être inchangés et seulement les préférences auraient changé.

Préférences de localisateur : envoyé facultativement quand les localisateurs n'ont pas tous la même préférence.

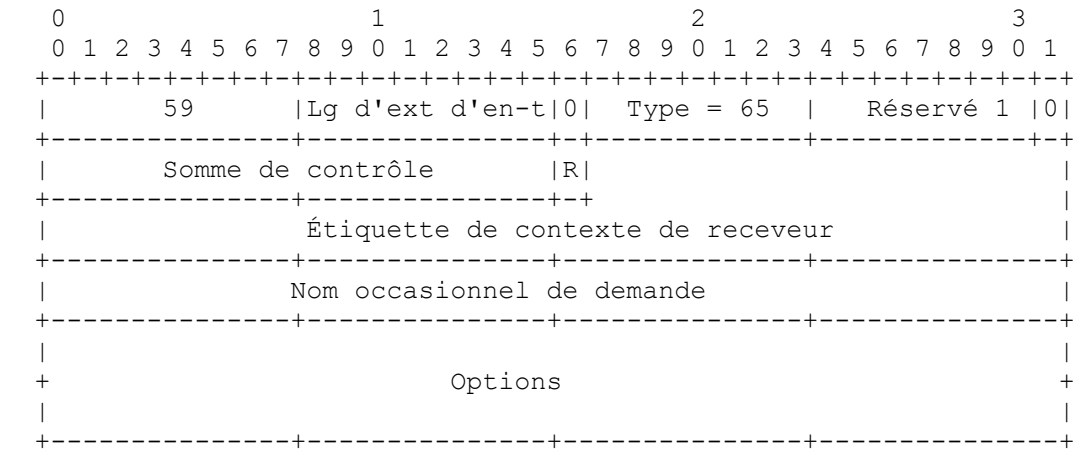
Structure de données de paramètres (PDS) de CGA : inclus quand la liste de localisateurs est incluse et que la PDS n'était pas incluse dans les messages I2/ I2bis/R2, afin que le receveur puisse vérifier la liste de localisateurs.

Signature de CGA : incluse quand certains des localisateurs dans la liste utilisent une CGA (et non une HBA) pour la vérification.

De futures extensions de protocole pourraient définir des options supplémentaires pour ce message. Le bit C dans le format d'option définit comment une telle nouvelle option va être traitée par une mise en œuvre. Voir le paragraphe 5.15.

5.11 Format de message accusé de réception de mise à jour

Ce message est envoyé en réponse à un message Demande de mise à jour. Il implique que la demande de mise à jour a été reçue et que tous les nouveaux localisateurs dans la demande de mise à jour peuvent maintenant être utilisés comme localisateurs de source des paquets. Mais cela n'implique pas que des (nouveaux) localisateurs ont été vérifiés pour être utilisés comme destination, car l'hôte pourrait différer la vérification d'un localisateur jusqu'à ce qu'il voit le besoin d'utiliser un localisateur comme destination.



Prochain en-tête : NO_NXT_HDR (59).

Longueur d'extension d'en-tête : au moins 1, car l'en-tête fait 16 octets quand il n'y a pas d'options.

Type : 65

Réservé 1 : champ de 7 bits. Réservé pour utilisation future. Zéro à l'émission. DOIT être ignoré à réception.

R : champ de 1 bit. Réservé pour utilisation future. Zéro à l'émission. DOIT être ignoré à réception.

Étiquette de contexte de receveur : champ de 47 bits. L'étiquette de contexte que le receveur a alloué pour le contexte.

Nom occasionnel de demande : entier non signé de 32 bits. Copié du message Demande de mise à jour.

Aucune option n'est actuellement définie pour ce message.

De futures extensions de protocole pourraient définir des options supplémentaires pour ce message. Le bit C dans le format d'option définit comment une telle nouvelle option va être traitée par une mise en œuvre. Voir le paragraphe 5.15.

5.12 Format de message Keepalive

Ce format de message est défini dans la [RFC5534].

Le message est utilisé pour s'assurer que quand un homologue envoie des paquets d'ULP dans un contexte, il reçoit toujours des paquets dans la direction inverse. Quand l'ULP envoie du trafic bidirectionnel, aucun paquet supplémentaire n'a besoin d'être inséré. Mais pour un schéma de trafic d'ULP unidirectionnel, Shim va renvoyer des messages Keepalive quand il reçoit des paquets d'ULP.

5.13 Format de message Probe

Ce message et sa sémantique sont définis dans la [RFC5534].

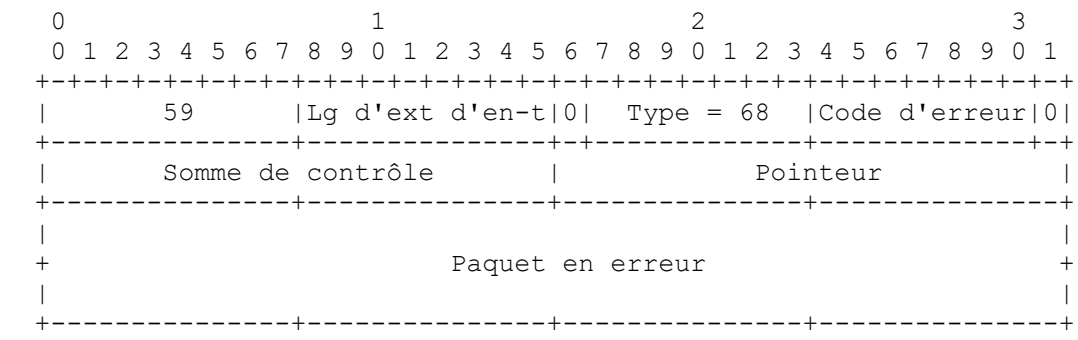
Le but de ce mécanisme est de vérifier si les paires de localisateur fonctionnent ou non dans le cas général. En particulier, ce mécanisme est capable de traiter le cas où une paire de localisateurs fonctionne de A à B et une autre paire de localisateurs fonctionne de B à A, mais où il n'y a pas de paire de localisateurs qui fonctionne dans les deux directions. Le mécanisme de protocole est que, lorsque A envoie des messages Probe à B, B va observer quelles paires de localisateurs il a reçu et va en faire rapport dans des messages Probe qu'il envoie à A.

5.14. Format de message Error

Le message d'erreur est généré par un receveur Shim6 à réception d'un message Shim6 contenant des informations critiques qui ne peuvent pas être traitées correctement.

Dans le cas où un nœud Shim6 reçoit un paquet Shim6 qui contient des informations critiques pour le protocole Shim6 et qui ne sont pas prises en charge par le receveur, il renvoie un message Erreur à l'origine du message Shim6. Il n'est pas fait d'accusé de réception au message d'erreur.

De plus, les messages d'erreur Shim6 définis dans cette section peuvent être utilisés pour identifier des problèmes avec les mises en œuvre de Shim6. Pour faire cela, une gamme de types de code d'erreur est réservée. En particulier, les mises en œuvre peuvent générer des messages d'erreur Shim6 avec des types de code dans cette gamme, au lieu d'éliminer en silence les paquets Shim6 durant le processus de débogage.



Prochain en-tête : NO_NXT_HDR (59).

Longueur d'extension d'en-tête : au moins 1, car l'en-tête fait 16 octets. Dépend des données d'erreur spécifiques.

Type : 68

Code d'erreur : champ de 7 bits décrivant l'erreur qui génère le message d'erreur. Voir la liste des codes d'erreur ci-dessous.

Pointeur : champ de16 bits. Identifie le décalage d'octets au sein du paquet où l'erreur a été détectée.

Paquet en erreur : autant du paquet en cause sans que le paquet de message d'erreur excède la MTU IPv6.

Les codes d'erreur suivants sont définis :

Code	Description
0	Type de message Shim6 inconnu
1	Option critique non reconnue
2	Échec de la méthode de vérification de localisateur (Pointeur sur un mauvais octet de méthode de vérification)
3	Nombre de génération de liste de localisateur hors de synchronisation.
4	Erreur du nombre de localisateurs dans une option Préférence de localisateurs.
120-127	Réservé aux fins de débogage.

Tableau 2

5.15 Formats d'option

Le format des options est une photographie du format d'option HIP actuel [RFC5201]. Cependant, il n'est pas prévu de changer quoi que ce soit au format d'option HIP, ni d'utiliser le même espace de noms pour les valeurs de type d'option. Mais utiliser le même format rendra plus facile d'importer les capacités HIP dans Shim6 comme extensions à Shim6, si cela devait se révéler utile.

Tous les paramètres de TLV ont une longueur (incluant les champs Type et Longueur) qui est un multiple de 8 octets. Quand nécessaire, un bourrage DOIT être ajouté à l'extrémité du paramètre afin que la longueur totale devienne un multiple de 8 octets. Cette règle assure un alignement approprié des données. Si un bourrage est ajouté, le champ Longueur NE DOIT PAS inclure le bourrage. Tous les octets de bourrage ajoutés DOIVENT être mis à zéro par l'envoyeur, et leur valeur NE DEVRAIT PAS être vérifiée par le receveur.

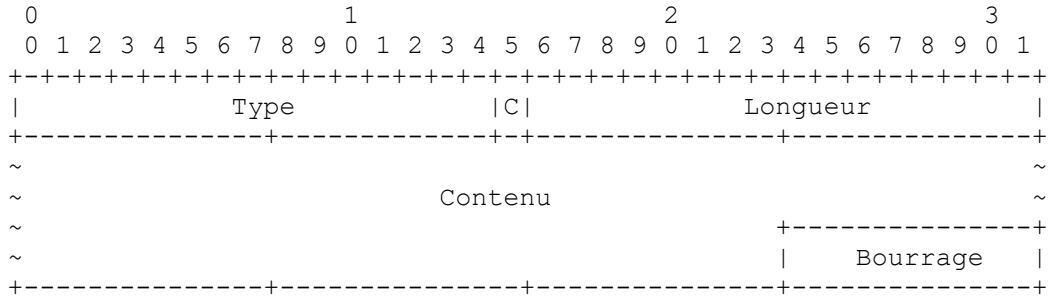
Par conséquent, le champ Longueur indique la longueur du champ Contenu (en octets). La longueur totale du paramètre de TLV (incluant Type, Longueur, Contenu, et Bourrage) est relative au champ Longueur selon la formule suivante :

Longueur totale = 11 + Longueur - (Longueur + 3) mod 8 ;

La longueur totale de l'option est le plus petit multiple de 8 octets qui permet les 4 octets de l'en-tête Option et l'option elle-même. La quantité de bourrage requise peut être calculée comme suit :

Bourrage = 7 - ((Longueur + 3) mod 8)

Longueur totale = 4 + Longueur + Bourrage



Type : identifiant de 15 bits du type d'option. Les options définies dans le présent document sont ci-dessous.

C : Critique. Un, si ce paramètre est critique et DOIT être reconnu par le receveur ; zéro autrement. Une mise en œuvre pourrait voir le bit C comme partie du champ Type en multipliant par deux les valeurs de type de cette spécification.

Longueur : longueur du contenu, en octets.

Contenu : spécifique du paramètre, défini par le type.

Bourrage : bourrage, de 0 à 7 octets, ajouté si nécessaire.

Type	Nom d'option
1	Valideur de répondeur
2	Liste de localisateurs
3	Préférences de localisateurs
4	Structure de données de paramètres de CGA
5	Signature de CGA
6	Paire d'ULID
7	Identifiant d'instance fourchée
10	Option de Fin de temporisation de maintien en vie

Tableau 3

De futures extensions du protocole pourraient définir des options supplémentaires pour les messages. Le bit C dans le format d'option définit comment de telles nouvelles options vont être traitées par une mise en œuvre.

De futures extensions de protocole pourraient définir des options supplémentaires pour les messages Shim6. Le bit C dans le format d'option définit comment une telle nouvelle option va être traitée par une mise en œuvre.

Si un hôte reçoit une option qu'il ne comprend pas (une option qui a été définie dans une future extension au présent protocole) ou qui n'est pas mentionnée comme option valide pour les différents types de message ci-dessus, alors le bit Critique dans l'option détermine le résultat.

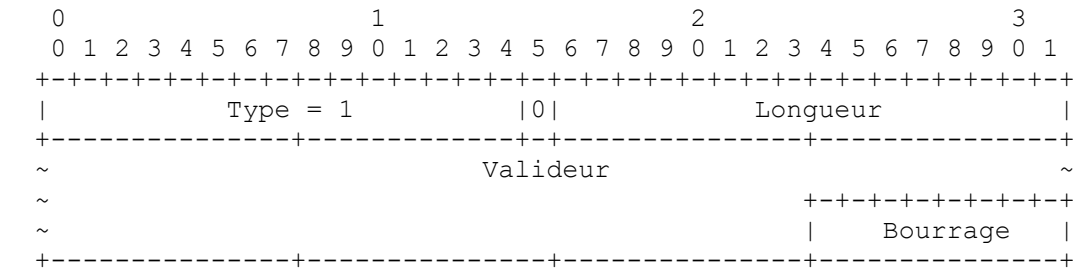
- Si C=0, l'option est alors ignorée en silence, et le reste du message est traité.

- Si C=1, l'hôte DEVRAIT alors renvoyer un message d'erreur Shim6 avec Code d'erreur=1, avec le champ Pointeur faisant référence au premier octet du champ Type d'option. Quand C=1, le reste du message NE DOIT PAS être traité.

5.15.1 Format d'option Valideur de répondeur

Le répondeur peut choisir exactement quelle entrée est utilisée pour calculer le valideur et quelle fonction unidirectionnelle (comme MD5 ou SHA1) il utilise, tant que le répondeur peut vérifier que le valideur qu'il reçoit en retour dans le message I2 ou I2bis est bien un qui :
1) a calculé,
2) a calculé pour le contexte particulier, et
3) n'est pas un message I2/I2bis répété.

Des suggestions sur la façon de générer les valideurs sont données aux paragraphes 7.10.1 et 7.17.1.



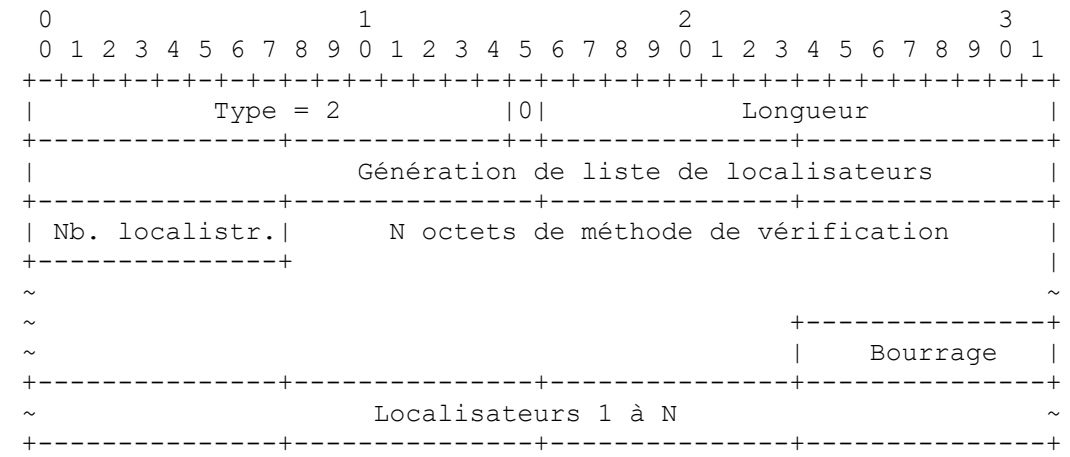
Valideur : contenu de longueur variable dont l'interprétation est locale pour le répondeur.

Bourrage : bourrage de 0 à 7 octets, ajouté si nécessaire. Voir le paragraphe 5.15.

5.15.2 Format d'option Liste de localisateurs

L'option Liste de localisateurs est utilisée pour porter tous les localisateurs de l'envoyeur. Noter que l'ordre des localisateurs est important, car l'option Préférences de localisateurs se réfère aux localisateurs en utilisant leur indice dans la liste.

Noter qu'on porte tous les localisateurs dans cette option même si certains d'entre eux peuvent être créés automatiquement à partir de la structure de données de paramètres de CGA.



Génération de liste de localisateurs : entier non signé de 32 bits. Indique une génération de nombre qui est augmentée de un pour chaque nouvelle liste de localisateurs. C'est utilisé pour s'assurer que l'indice dans les préférences de localisateurs se réfère à la bonne version de liste des localisateurs.

Nombre de localisateurs : entier non signé de 8 bits. Nombre de localisateurs qui sont inclus dans l'option. On appelle ce nombre "N" ci-dessous.

Méthode de vérification : N octets. Le ième octet spécifie la méthode de vérification pour le ième localisateur.

Bourrage : 0 à 7 octets, ajoutés si nécessaire afin que les localisateurs commencent sur une limite d'un multiple de 8 octets. Noter que pour cette option, il n'est jamais besoin de bourrer à l'extrémité car les localisateurs sont un multiple de 8 octets. Ce bourrage interne est inclus dans le champ Longueur.

Localisateurs : N localisateurs de 128 bits.

Les méthodes de vérification définies sont :

Valeur	**Méthode**
0	Réservé
1	HBA
2	CGA
3-200	Alloué en utilisant une action de normalisation
201-254	Utilisation expérimentale
255	Réservé

Tableau 4

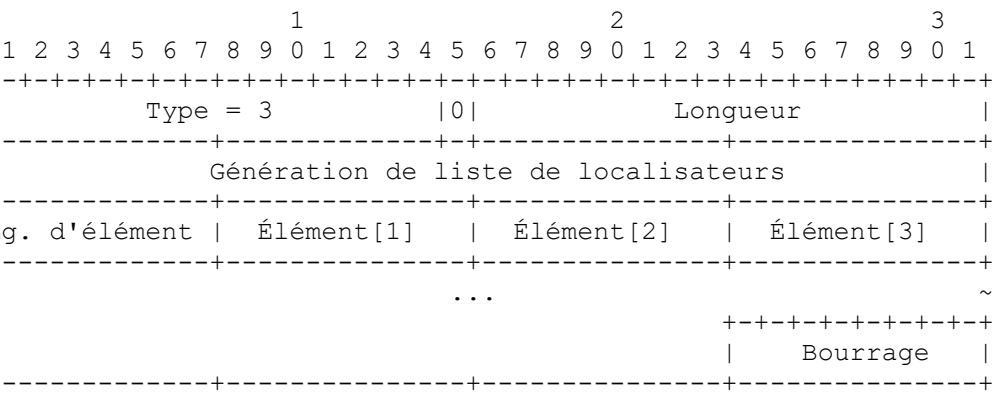
5.15.3 Format d'option Préférences de localisateurs

L'option Préférences de localisateurs peut avoir des fanions pour indiquer si un localisateur est ou non connu comme fonctionnant. De plus, l'envoyeur peut inclure une notion de préférences. On pourrait définir "préférences" comme une combinaison de priorité et de pondération, de la même façon que les enregistrements SRV du DNS ont de telles informations. La priorité va fournir un moyen de classer les localisateurs, et, dans une priorité donnée, la pondération va donner un moyen de faire une sorte de partage de charge. Voir dans la [RFC2782] comment un SRV définit l'interaction de la priorité et de la pondération.

La notion minimum de préférence dont on a besoin est d'être capable d'indiquer qu'un localisateur est "mort". On peut traiter cela en utilisant un fanion d'un seul octet pour chaque localisateur.

On peut étendre cela en portant un plus grand "élément" pour chaque localisateur. Le présent document définit présentement aussi des éléments de 2 et de 3 octets, et on peut ajouter plus d'informations en ayant des éléments encore plus grands si il en est besoin.

Les localisateurs ne sont pas inclus dans la liste de préférences. Le premier élément se réfère plutôt au localisateur qui était dans le premier élément dans l'option Liste de localisateurs. Le nombre de génération porté dans cette option et l'option Liste de localisateurs sont utilisés pour vérifier qu'ils se réfèrent à la même version de la liste des localisateurs.



Le cas de Longueur d'élément = 1 est décrit.

Génération de liste de localisateurs : entier non signé de 32 bits. Indique un nombre de génération pour la liste des localisateurs auxquels les éléments devraient s'appliquer.

Longueur d'élément : entier non signé de 8 bits. Longueur en octets de chaque élément. La présente spécification définit les cas où la longueur est 1, 2, ou 3.

Élément[i] : Champ avec un nombre d'octets défini par le champ Longueur d'élément. Donne les préférences pour le ième localisateur dans l'option Liste de localisateurs utilisée.

Bourrage : 0 à 7 octets, ajouté si nécessaire. Voir le paragraphe 5.15.

Quand Longueur d'élément est égal à un, l'élément consiste alors en seulement un champ Fanions d'un octet. L'ensemble de fanions actuellement définis est :

BROKEN : 0x01 *(cassé)*
TRANSIENT : 0x02 *(transitoire)*

L'intention du fanion BROKEN est d'informer l'homologue qu'un certain localisateur est connu pour ne pas fonctionner. L'intention de TRANSIENT est de permettre la distinction entre des adresses plus stables et des adresses moins stables quand Shim6 est combiné avec la mobilité IP, et quand on pourrait avoir des localisateurs de rattachement plus stables et des localisateurs d'entretien moins stables.

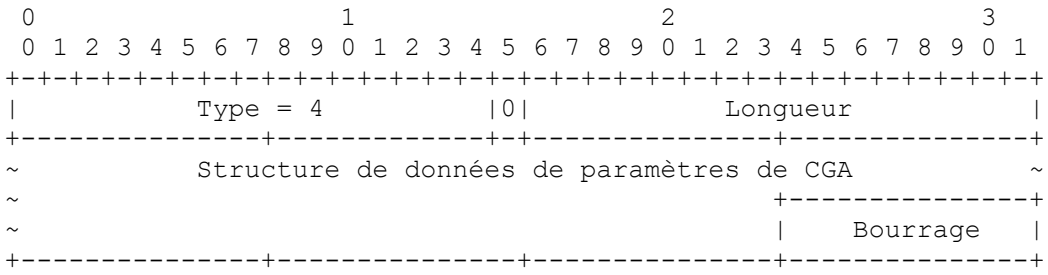
Quand Longueur d'élément est égal à deux, l'élément consiste alors en un champ Fanions de un octet suivi par un champ Priorité d'un octet. Ce champ Priorité a la même sémantique que le champ Priorité dans les enregistrements SRV du DNS.

Quand Longueur d'élément est égal à trois, l'élément consiste alors en un champ Fanions de un octet suivi par un champ Priorité d'un octet et un champ Pondération d'un octet. Ce champ Pondération a la même sémantique que le champ Pondération dans les enregistrements SRV du DNS.

Le présent document ne spécifie pas le format quand Longueur d'élément fait plus de trois, sauf que de tels formats DOIVENT être définis de telle façon que les trois premiers octets soient les mêmes que dans le cas ci-dessus, c'est-à-dire, un champ Fanions d'un octet suivi par un champ Priorité d'un octet, et un champ Pondération d'un octet.

5.15.4 Format d'option Structure de données de paramètres de CGA

Cette option contient la structure de données de paramètres de CGA (PDS). Quand HBA est utilisé pour vérifier les localisateurs, la PDS contient l'extension HBA multi préfixes en plus des champs obligatoires de PDS et autres extensions sans relation avec Shim6 que la PDS pourrait avoir. Quand CGA est utilisé pour vérifier les localisateurs, en plus de l'option PDS, l'hôte doit aussi inclure la signature sous la forme d'une option Signature de CGA.

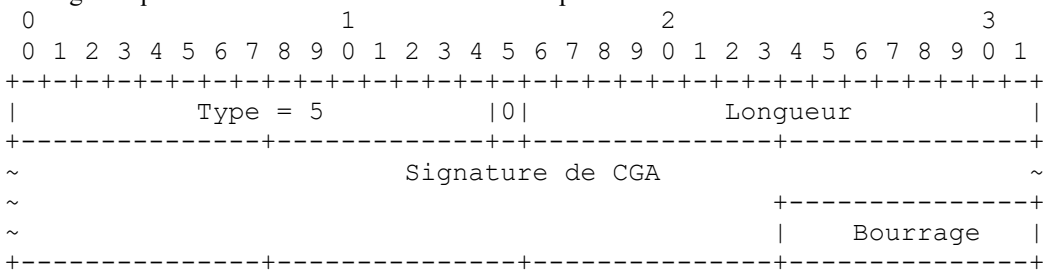


Structure de données de paramètres de CGA : contenu de longueur variable, défini dans les [RFC3972] et [RFC5535].

Bourrage : 0 à 7 octets, ajouté si nécessaire. Voir le paragraphe 5.15.

5.15.5 Format d'option Signature de CGA

Quand CGA est utilisé pour la vérification d'un ou plusieurs des localisateurs dans l'option Liste de localisateurs, le message en question va alors devoir contenir cette option.



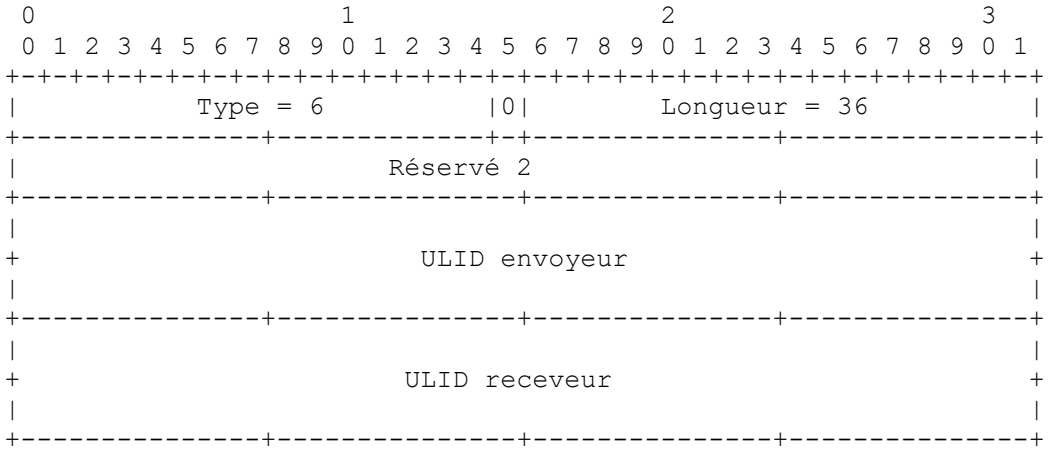
Signature de CGA : champ de longueur variable contenant une signature PKCS n°1 v1.5, construite en utilisant la clé privée de l'envoyeur sur la séquence d'octets suivante :

1. La valeur de l'étiquette de type de message CGA de 128 bits [CGA] pour Shim6 : 0x4A 30 5662 4858 574B 3655 416F 506A 6D48. (La valeur d'étiquette a été générée au hasard par l'éditeur de cette spécification.).
2. La génération de liste de localisateurs : nombre de l'option correspondante de liste de localisateurs.
3. Le sous ensemble de localisateurs inclus dans l'option correspondante de liste de localisateurs dont la méthode de vérification est réglée à CGA. Les localisateurs DOIVENT être inclus dans l'ordre dans lequel ils figurent sur la liste de l'option Liste de localisateurs.

Bourrage : 0 à 7 octets, ajouté si nécessaire. Voir le paragraphe 5.15.

5.15.6 Format d'option Paire d'ULID

Les messages I1, I2, et I2bis DOIVENT contenir la paire d'ULID ; normalement, c'est dans les champs Source et Destination IPv6. Dans le cas où l'ULID pour le contexte diffère de la paire d'adresses incluse dans les champs Adresse de source et Adresse de destination du paquet IPv6 utilisé pour porter le message I1/I2/I2bis, l'option Paire d'ULID DOIT être incluse dans le message I1/I2/I2bis.

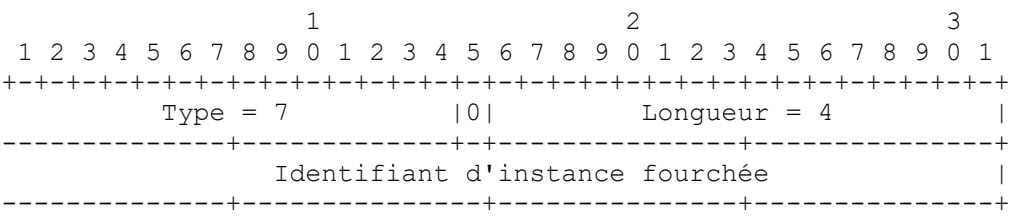


Réservé 2 : champ de 32 bits. Réservé pour utilisation future. Zéro à l'émission. DOIT être ignoré à réception. (Nécessaire pour faire commencer les ULID sur une limite d'un multiple de 8 octets.)

ULID envoyeur : adresse IPv6 de 128 bits.

ULID receveur : adresse IPv6 de 128 bits.

5.15.7 Format d'option Identifiant d'instance fourchée



Identifiant d'instance fourchée : champ de 32 bits qui contient l'identifiant de l'instance fourchée particulière.

5.15.8 Format d'option Fin de temporisation de maintien en vie

Cette option est définie dans la [RFC5534].

6. Modèle conceptuel d'un hôte

Cette section décrit un modèle conceptuel d'une organisation possible de structure de données que les hôtes vont maintenir pour les besoins de Shim6. L'organisation décrite est fournie pour faciliter l'explication de comment le protocole Shim6 devrait se comporter. Le présent document ne rend pas obligatoire que les mises en œuvre adhèrent à ce modèle pour autant que leur comportement externe est cohérent avec celui décrit dans ce document.

6.1 Structures de données conceptuelles

La structure clé de données conceptuelle pour le protocole Shim6 est le contexte de paire d'ULID. C'est une structure de données qui contient les informations suivantes :

- o L'état du contexte. Voir le paragraphe 6.2.
- o L'ULID de l'homologue : ULID(homologue).
- o L'ULID local : ULID(local).
- o L'identifiant d'instance fourchée : FII. C'est zéro pour le contexte par défaut, c'est-à-dire, quand il n'y a pas de fourchement.
- o La liste des localisateurs de l'homologue avec leurs préférences : Ls(homologue).
- o Le nombre de génération pour la plus récemment reçue, la liste de localisateurs d'homologue vérifiée.
- o Pour chaque localisateur de l'homologue, la méthode de vérification à utiliser (provenant de l'option Liste de localisateurs).
- o Pour chaque localisateur de l'homologue, un fanion spécifiant si il a été vérifié en utilisant HBA ou CGA, et un bit spécifiant si le localisateur a été sondé pour vérifier que l'ULID est présent à cette localisation.
- o Le localisateur de l'homologue actuel est le localisateur utilisé comme adresse de destination lors de l'envoi des paquets: Lp(homologue).
- o L'ensemble de localisateurs locaux et leurs préférences : Ls(local).
- o Le nombre de génération pour l'option Liste de localisateurs la plus récemment envoyée.
- o Le localisateur local actuel est le localisateur utilisé comme adresse de source lors de l'envoi des paquets : Lp(local).
- o L'étiquette de contexte utilisée pour transmette les messages de contrôle et les en-têtes Extension de charge utile Shim6 ; elle est allouée par l'homologue : CT(homologue).
- o Le contexte à attendre dans les messages de contrôle reçus et les en-têtes Extension de charge utile Shim6 ; il est alloué par l'hôte local : CT(local).
- o Les temporisateurs pour la retransmission des messages durant l'établissement de contexte et les messages de mise à jour.
- o Selon la façon dont une mise en œuvre détermine si un contexte est encore utilisé, il pourrait y avoir besoin de noter la

dernière fois qu'un paquet a été envoyé/reçu en utilisant le contexte.
- o L'état d'accessibilité pour les paires de localisateurs, comme spécifié dans la [RFC5534].
- o Durant l'exploration de paires, les informations sur les messages Probe qui ont été envoyés et reçus, comme spécifié dans la [RFC5534].
- o Durant la phase d'établissement de contexte, le nom occasionnel d'initiateur, le nom occasionnel de répondeur, le valeur de répondeur, et les temporisateurs relatifs aux différents paquets envoyés (I1,I2, R2), comme décrit à la Section 7.

6.2 ÉTATS de contexte

Les ÉTATS qui sont utilisés pour décrire le protocole Shim6 sont les suivants :

ÉTAT	Explication
IDLE	Début de l'automate à états
I1-SENT	Initiation de l'échange d'établissement de contexte
I2-SENT	Attente d'achèvement de l'échange d'établissement de contexte
I2BIS-SENT	Détection d'une potentielle perte de contexte
ESTABLISHED	Contexte SHIM établi
E-FAILED	Échec de l'échange d'établissement de contexte
NO-SUPPORT	Type de Prochain en-tête ICMP reçu non reconnu (type 4, code 1) indiquant que Shim6 n'est pas pris en charge

De plus, dans chaque ÉTAT susmentionné, les informations d'état suivantes sont mémorisées :

ÉTAT	Information
IDLE	aucune
I1-SENT	ULID(homologue), ULID(local), [FII], CT(local), INIT Nonce, Lp(local), Lp(homologue), Ls(local)
I2-SENT	ULID(homologue), ULID(local), [FII], CT(local), INIT Nonce, RESP Nonce, Lp(local), Lp(homologue), Ls(local), Valideur de répondeur
ESTABLISHED	ULID(homologue), ULID(local), [FII], CT(local), CT(homologue), Lp(local), Lp(homologue), Ls(local), Ls(homologue), INIT Nonce?(pour recevoir des R2 en retard)
I2BIS-SENT	ULID(homologue), ULID(local), [FII], CT(local), CT(homologue), Lp(local), Lp(homologue), Ls(local), Ls(homologue), CT(R1bis), RESP Nonce, INIT Nonce, Valideur de répondeur
E-FAILED	ULID(homologue), ULID(local)
NO-SUPPORT	ULID(homologue), ULID(local)

7. Établissement des contextes de paire d'ULID

Les contextes de paires d'ULID sont établis en utilisant un échange en quatre phases, qui permet au répondeur d'éviter de créer un état sur le premier paquet. Au titre de cet échange, chaque extrémité alloue une étiquette de contexte et partage cette étiquette de contexte et son ensemble de localisateurs avec l'homologue.

Dans certains cas, l'échange en quatre phases n'est pas nécessaire -- par exemple, quand les deux extrémités essayent d'établir le contexte en même temps, ou quand on récupère d'un contexte qui a été mal collecté ou perdu chez un des hôtes.

7.1 Unicité des étiquettes de contexte

Au titre de l'établissement d'un nouveau contexte, chaque hôte doit allouer une étiquette de contexte unique. Comme les en-têtes d'extension de charge utile Shim6 sont démultiplexés sur la seule base de la valeur de l'étiquette de contexte (sans utiliser les localisateurs) l'étiquette de contexte DOIT être unique pour chaque contexte.

Il est important que les étiquettes de contexte soient difficiles à deviner pour des attaquants hors chemin. Donc, si une mise en œuvre utilise une structure dans l'étiquette de contexte pour faciliter l'efficacité des recherches, au moins 30 bits de l'étiquette de contexte DOIVENT être non structurés et remplis de bits aléatoires ou pseudo-aléatoires.

De plus, afin de minimiser la réutilisation des étiquettes de contexte, l'hôte DEVRAIT circuler de façon aléatoire dans l'espace de noms d'étiquettes non structuré qui est réservé pour les valeurs d'étiquette de contexte allouées (par exemple, suivant les lignes directrices décrites dans la [RFC4086]).

7.2 Vérification de localisateur

Les localisateurs de l'homologue pourraient devoir être vérifiés durant l'établissement de contexte ainsi que lors du traitement des mises à jour de localisateurs de la Section 10.

Il y a deux aspects distincts de la vérification de localisateur. L'un d'eux est de vérifier que le localisateur est lié à l'ULID, c'est-à-dire, que l'hôte qui "possède" l'ULID est aussi celui qui revendique la "propriété" du localisateur. Le protocole Shim6 utilise les techniques de HBA ou CGA pour faire cette vérification. L'autre aspect est de vérifier que l'hôte est bien accessible au localisateur revendiqué. Une telle vérification est nécessaire non seulement pour s'assurer que la communication peut se faire mais aussi pour empêcher des attaques d'inondation par un tiers [RFC4218]. Ces différents aspects de la vérification de localisateur se passent à des moments différents car le premier pourrait devoir être effectué avant que des paquets puissent être reçus par l'homologue avec le localisateur de source en question, mais la dernière vérification est seulement nécessaire avant que les paquets soient envoyés au localisateur.

Avant qu'un hôte puisse utiliser un localisateur (différent de l'ULID) comme localisateur de source, il doit savoir que l'homologue va accepter des paquets avec ce localisateur de source au titre de ce contexte. Donc, la vérification HBA/CGA DEVRAIT être effectuée par l'hôte avant qu'il accuse réception du nouveau localisateur en envoyant un message d'accusé de réception de mise à jour ou un message R2.

Avant qu'un hôte puisse utiliser un localisateur (différent de l'ULID) comme localisateur de destination, il DOIT effectuer la vérification HBA/CGA si elle n'a pas été effectuée à réception de l'ensemble de localisateurs. De plus, il DOIT vérifier que l'ULID est bien présent à ce localisateur. Cette vérification est effectuée en faisant un essai d'acheminement de retour au titre du sous protocole de sonde [RFC5534].

Si la méthode de vérification dans l'option Liste de localisateurs n'est pas prise en charge par l'hôte, ou si la méthode de vérification n'est pas cohérente avec la structure de données de paramètres de CGA (par exemple, la structure de données de paramètre ne contient pas l'extension multi préfixes et la méthode de vérification dit d'utiliser HBA) alors l'hôte DOIT ignorer la liste de localisateurs et le message dans lequel elle est contenue. L'hôte DEVRAIT générer un message d'erreur Shim6 avec le code d'erreur =2 et avec le pointeur qui référence l'octet dans la méthode de vérification qui a été trouvé incohérent.

7.3 Établissement de contexte normal

L'établissement de contexte normal consiste en un échange de 4 messages dans l'ordre I1, R1, I2, R2, comme on peut le voir à la Figure 3.

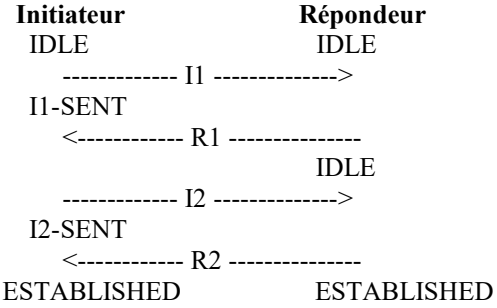


Figure 3 : Établissement de contexte normal

7.4 Établissement de contexte concurrent

Quand les deux extrémités essayent d'initier un contexte pour la même paire d'ULID, on pourrait alors finir par croiser les messages I1. Autrement, comme aucun état n'est créé quand il reçoit le I1, un hôte pourrait envoyer un I1 après avoir envoyé un message R1.

Comme un hôte se souvient qu'il a envoyé un I1, il peut répondre à un I1 provenant de l'homologue (pour la même paire d'ULID) avec un R2, résultant en l'échange de messages montré à la Figure 4. Un tel comportement est nécessaire pour des raisons comme de répondre correctement aux messages I1 retransmis, ce qui arrive quand le message R2 a été perdu.

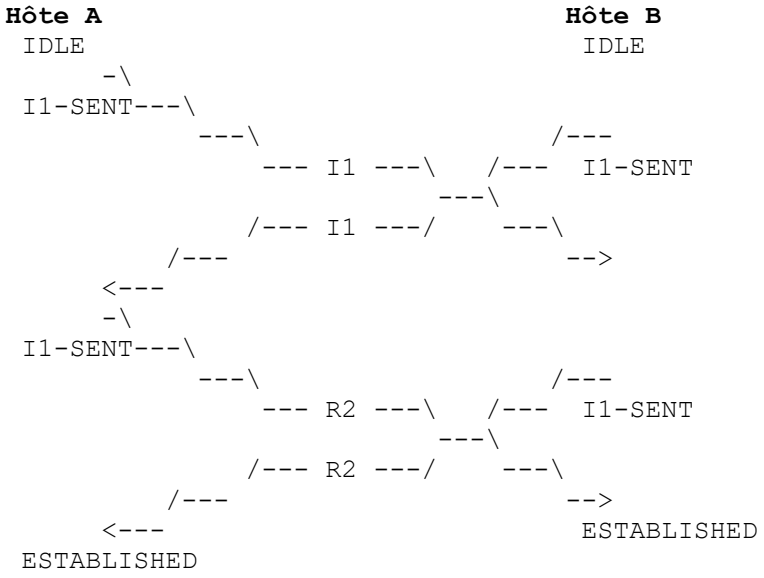


Figure 4 : Croisement des messages I1

Si un hôte a reçu un I1 et envoyé un R1, il n'a pas d'état pour s'en souvenir. Donc, si l'ULP sur l'hôte envoie des paquets, cela pourrait déclencher l'envoi par l'hôte lui-même d'un message I1. Donc, alors qu'une extrémité est en train d'envoyer un I1, l'autre envoie un I2, comme on peut le voir à la Figure 5.

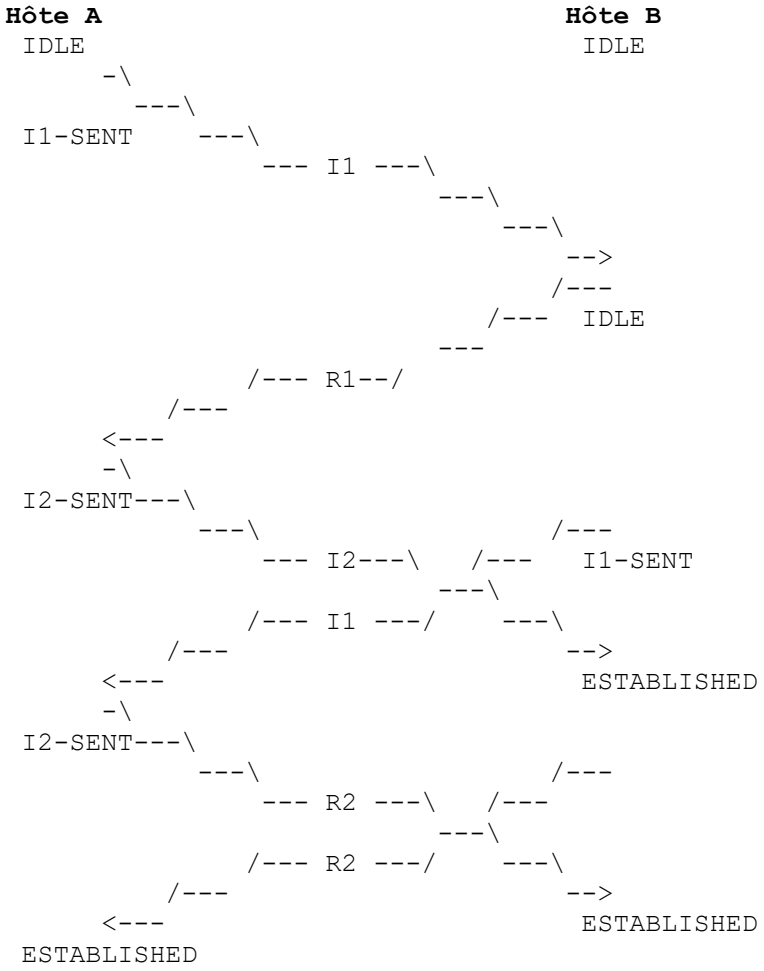


Figure 5 : Croisement de I2 et I1

7.5 Récupération de contexte

Du fait d'une mauvaise collecte, on peut finir avec une extrémité qui a et utilise l'état du contexte, et l'autre extrtémité qui n'a aucun état. On doit être capable de récupérer cet état à l'extrémité qui l'a perdu avant de pouvoir l'utiliser.

Ce besoin peut apparaître dans les cas suivants :

- La communication fonctionne en utilisant la paire d'ULID comme paire de localisateurs mais un problème se produit, et l'extrémité qui a conservé l'état du contexte décide de sonder des paires de localisateurs de remplacement.

- La communication fonctionne en utilisant une paire de localisateurs qui n'est pas la paire d'ULID ; donc, les paquets d'ULP envoyés d'un homologue qui a conservé l'état du contexte utilisent l'en-tête Extension de charge utile Shim6.

- L'hôte qui a conservé l'état envoie un message de contrôle (par exemple, un message Demande de mise à jour).

Dans tous les cas, le résultat est que l'homologue sans état reçoit un message Shim pour lequel il n'a pas de contexte pour l'étiquette de contexte.

On peut récupérer le contexte en faisant que le nœud qui n'a pas d'état de contexte renvoie un message R1bis, et ensuite en achevant la récupération avec un message I2bis et R2, comme on peut le voir à la Figure 6.

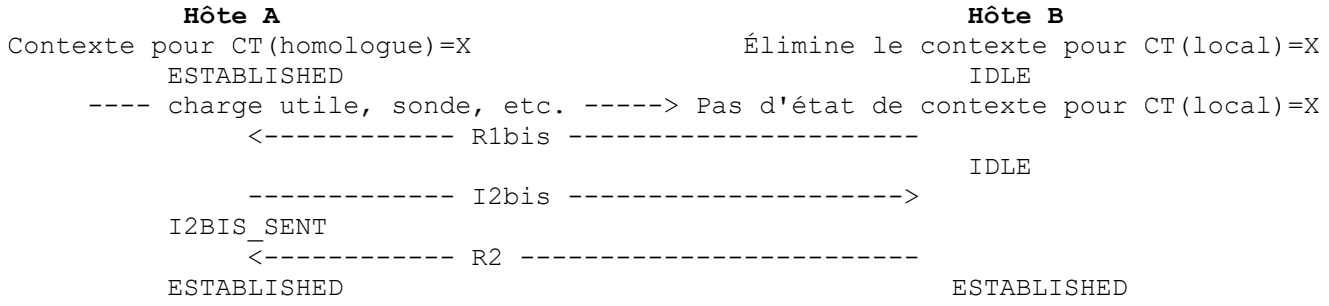


Figure 6 : Perte de contexte chez le receveur

Si une extrémité a mal collecté ou perdu l'état du contexte, elle pourrait essayer de créer un nouvel état de contexte (pour la même paire d'ULID) en envoyant un message I1. Dans ce cas, l'homologue (qui a encore l'état du contexte) va répondre avec un message R1, et l'échange en quatre phases complet va être effectué à nouveau, comme on peut le voir à la Figure 7.

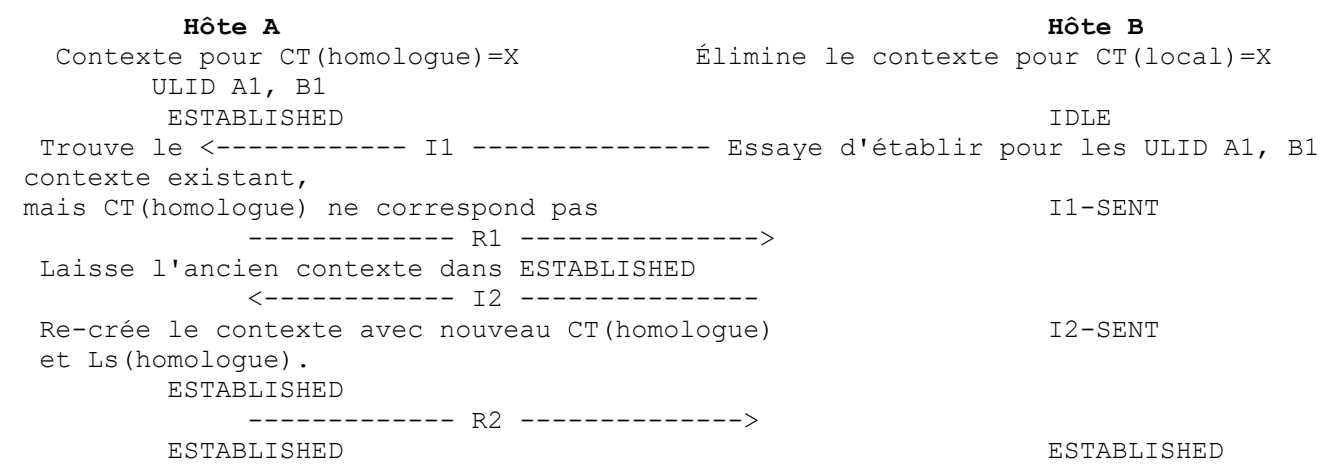


Figure 7 : Perte de contexte chez l'envoyeur

7.6 Confusion de contexte

Comme chaque extrémité pourrait mal collecter l'état du contexte, on peut avoir le cas où une extrémité a conservé l'état du contexte et essaye de l'utiliser, alors que l'autre extrémité a perdu l'état. On discute cela dans le paragraphe précédent sur la récupération. Mais, pour les mêmes raisons, quand un hôte conserve l'étiquette de contexte X comme CT(homologue) pour

la paire d'ULID <A1, B1>, l'autre extrémité pourrait finir par allouer cette étiquette de contexte comme CT(local) pour une autre paire d'ULID (par exemple, <A3, B1>) entre les mêmes hôtes. Dans ce cas, on ne peut pas utiliser les mécanismes de récupération car il faut avoir des étiquettes de contexte distinctes pour les deux paires d'ULID.

Ce type de "confusion" peut être observé dans deux cas (en supposant que c'est A qui a conservé l'état et B qui l' éliminé) :
- o B décide de créer un contexte pour la paire d'ULID <A3, B1>, alloue X comme étiquette de contexte pour elle, et envoie un I1 à A.
- o A décide de créer un contexte pour la paire d'ULID <A3, B1> et commence l'échange en envoyant un I1 à B. Quand B reçoit le message I2, il alloue X comme étiquette de contexte pour ce contexte.

Dans les deux cas, A peut détecter que B a alloué X pour la paire d'ULID <A3, B1> même si A a encore X comme CT(homologue) pour la paire d'ULID <A1, B1>. Donc, A peut détecter que B doit avoir perdu le contexte pour <A1, B1>.

La confusion peut être détectée quand I2/I2bis/R2 est reçu, car on exige que ces messages DOIVENT inclure un ensemble suffisamment grand de localisateurs dans une option Liste de localisateurs pour que l'homologue puisse déterminer si deux contextes ont ou non le même hôte que l'homologue en comparant si il y a des localisateurs communs dans Ls(homologue).

L'ancien contexte qui utilisait l'étiquette de contexte DOIT être supprimé ; il ne peut plus être utilisé pour envoyer des paquets. Donc, A va forcément supprimer l'état du contexte pour <A1, B1, X> afin qu'il puisse accepter le nouveau contexte pour <A3, B1, X>. Une mise en œuvre PEUT recréer un contexte pour remplacer celui qui a été supprimé -- dans ce cas, pour <A1, B1>. L'échange normal d'établissement I1, R1, I2, R2 va alors prendre des étiquettes de contexte uniques pour le contexte de remplacement. Cette recréation est FACULTATIVE, mais pourrait être utile quand une communication d'ULP utilise la paire d'ULID dont le contexte a été supprimé.

Noter qu'un message I1 avec une étiquette de contexte dupliquée ne devrait pas causer la suppression de l'ancien état de contexte ; cette opération doit être différée jusqu'à la réception du message I2.

7.7 Envoi des messages I1

Quand la couche Shim décide d'établir un contexte pour une paire d'ULID, elle commence par allouer et initialiser l'état du contexte pour son extrémité. À ce titre, elle alloue au hasard une étiquette de contexte au contexte qui n'est pas utilisé comme CT(local) par un autre contexte. Dans le cas où une nouvelle API est utilisée et où l'ULP demande un contexte fourché, la valeur de l'identifiant d'instance fourchée va être réglée à une valeur non zéro. Autrement, la valeur de FII est zéro. Alors l'initiateur peut envoyer un message I1 et établir l'état du contexte à I1-SENT. Le message I1 DOIT inclure la paire d'ULID -- normalement, dans les champs Source et Destination IPv6. Mais si la paire d'ULID pour le contexte n'est pas utilisée comme paire de localisateurs pour le message I1, alors une option ULID DOIT être incluse dans le message I1. De plus, si une valeur d'identifiant d'instance fourchée est non zéro, le message I1 DOIT inclure une option Identifiant d'instance de contexte contenant la valeur correspondante.

7.8 Retransmission des messages I1

Si l'hôte ne reçoit pas de message R1 ou R2 en réponse au message I1 après I1_TIMEOUT, alors il doit retransmettre le message I1. Les retransmissions devraient utiliser un temporisateur de retransmission avec un retard exponentiel binaire pour éviter de créer des problèmes d'encombrement au réseau quand un grand nombre d'hôtes effectuent des retransmissions de I1. Aussi, la valeur réelle de temporisation devrait être aléatoire entre 0,5 et 1,5 de la valeur nominale pour éviter l'auto synchronisation.

Si, après I1_RETRIES_MAX retransmissions, il n'y a pas de réponse, il est alors très probable que l'homologue ne met pas en œuvre le protocole Shim6 (ou il pourrait y avoir un pare-feu qui bloque le protocole). Dans ce cas, il est raisonnable que l'hôte se souvienne de ne pas essayer à nouveau d'établir un contexte avec cet ULID. Cependant, une telle mise en antémémoire négative devrait être conservée pendant au plus NO_R1_HOLDDOWN_TIME, afin d'être capable d'établir plus tard un contexte si le problème était que l'hôte n'était pas accessible du tout quand Shim a essayé d'établir le contexte.

Si l'hôte reçoit une erreur ICMP avec un type "Prochain en-tête non reconnu" (type 4, code 1) et si le paquet inclus est le message I1 qu'il vient juste d'envoyer, c'est alors une indication plus fiable que l'ULID de l'homologue ne met pas en œuvre Shim6. Là encore, dans ce cas, l'hôte devrait se souvenir de ne pas essayer à nouveau d'établir un contexte avec cet ULID. Une telle mise en antémémoire négative devrait être conservée pendant au plus ICMP_HOLDDOWN_TIME, qui devrait être significativement plus long que dans le cas précédent.

7.9 Réception des messages I1

Un hôte DOIT éliminer en silence tout message I1 reçu qui ne satisfait pas tous les essais de validité suivants en plus de ceux spécifiés au paragraphe 12.3 :

o Le champ Longueur d'extension d'en-tête est au moins 1, c'est-à-dire, la longueur est au moins 16 octets.

À réception d'un message I1, l'hôte extrait la paire d'ULID et l'identifiant d'instance fourchée du message. Si il n'y pas d'option Paire d'ULID, alors la paire d'ULID est prise des champs Source et Destination dans l'en-tête IPv6. Si il n'y pas d'option FII dans le message, alors la valeur de FII est prise comme zéro.

Ensuite, l'hôte cherche un contexte existant qui corresponde à la paire d'ULID et au FII.

Si aucun état n'est trouvé (c'est-à-dire, l'état est IDLE) alors l'hôte répond avec un message R1 comme spécifié ci-dessous.

Si un tel contexte existe dans l'état ESTABLISHED, l'hôte vérifie que le localisateur de l'initiateur est inclus dans Ls(homologue). (Cette vérification n'est pas nécessaire si il n'y a pas d'option Paire d'ULID dans le message I1.)

Si l'état existe comme ESTABLISHED et si les localisateurs ne rentrent pas dans les ensembles de localisateurs, alors l'hôte répond avec un message R1 comme spécifié ci-dessous. Cela achève le traitement de I1, avec l'état de contexte inchangé.

Si l'état existe comme ESTABLISHED et si les localisateurs entrent dans les ensembles, alors l'hôte compare CT(homologue) pour le contexte avec le CT contenu dans le message I1.

o Si les étiquettes de contexte correspondent, cela signifie probablement que le message R2 a été perdu et que ce I1 est une retransmission. Dans ce cas, l'hôte répond avec un message R2 contenant les informations disponibles pour le contexte existant.

o Si les étiquettes de contexte ne correspondent pas, cela signifie probablement que l'initiateur a perdu les informations de contexte pour ce contexte et qu'il essaye d'en établir un nouveau pour la même paire d'ULID. Dans ce cas, l'hôte répond avec un message R1 comme spécifié ci-dessous. Cela termine le traitement de I1, avec l'état de contexte inchangé.

Si l'état existe dans un autre état (I1-SENT, I2-SENT, I2BIS-SENT) on est dans une situation d'établissement de contexte concurrent, décrite au paragraphe 7.4. Dans ce cas, l'hôte laisse CT(homologue) inchangé et répond avec un message R2. Cela termine le traitement de I1, avec l'état de contexte STATE inchangé.

7.10 Envoi des messages R1

Quand l'hôte a besoin d'envoyer un message R1 en réponse au message I1, il copie le nom occasionnel d'initiateur provenant du message I1 dans le message R1, génère un nom occasionnel de répondeur, et calcule une option Valideur de répondeur comme suggéré au paragraphe suivant. Aucun état n'est créé chez l'hôte dans ce cas. (Noter que les informations utilisées pour générer le message de réponse R1 sont soit contenues dans le message I1 reçu, soit sont des informations globales (les valeurs S et de nom occasionnel de répondeur) qui ne sont pas associées au contexte particulier demandé.)

Quand l'hôte a besoin d'envoyer un message R2 en réponse au message I1, il copie le nom occasionnel d'initiateur du message I1 dans le message R2, et suit par ailleurs les règles normales pour former un message R2 (paragraphe 7.14).

7.10.1 Génération du valideur R1

Comme mentionné au paragraphe 5.15.1, le mécanisme de génération de valideur est un choix local car le valideur est généré et vérifié par le même nœud, c'est-à-dire, le répondeur. Cependant, afin de fournir la protection requise, le valideur doit être généré en satisfaisant les conditions décrites au paragraphe 5.15.1. Une façon pour que le répondeur génère correctement les valideurs est de garder un seul secret (S) et un compteur courant (C) pour le nom occasionnel de répondeur qui est incrémenté à des intervalles de temps fixés (cela permet au répondeur de vérifier l'âge d'un nom occasionnel de répondeur, indépendamment du contexte dans lequel il est utilisé).

Quand le valideur est généré pour être inclus dans un message R1 envoyé en réponse à un message I1 spécifique, le répondeur peut effectuer la procédure suivante pour générer la valeur de valideur :

D'abord, le répondeur utilise la valeur courante du compteur C comme nom occasionnel de répondeur.

Ensuite, il utilise les informations suivantes (enchaînées) comme entrées de la fonction unidirectionnelle :
- o Le secret S
- o Ce nom occasionnel de répondeur
- o L'étiquette de contexte d'initiateur provenant du message I1
- o Les ULID provenant du message I1
- o Les localisateurs provenant du message I1 (seulement strictement nécessaire si ils sont différents des ULID)
- o L'identifiant d'instance fourchée, si cette option était incluse dans le message I1

Enfin, il utilise le résultat de la fonction de hachage comme valeur de valideur incluse dans le message R1.

7.11 Réception des messages R1 et envoi des messages I2

Un hôte DOIT éliminer en silence tout message R1 reçu qui ne satisfait pas tous les essais de validité suivants en plus de ceux spécifiés au paragraphe 12.3 :

- o Le champ Longueur d'extension d'en-tête fait au moins 1, c'est-à-dire, la longueur est au moins 16 octets.

À réception d'un message R1, l'hôte extrait le nom occasionnel d'initiateur et la paire de localisateurs du message (cette dernière des champs Source et Destination dans l'en-tête IPv6). Ensuite, l'hôte cherche un contexte existant qui correspond au nom occasionnel d'initiateur et où les localisateurs sont contenus dans Ls(homologue) et Ls(local), respectivement. Si un tel contexte n'est pas trouvé, alors le message R1 est éliminé en silence.

Si un tel contexte est trouvé, l'hôte cherche alors l'état :
- o Si l'état est I1-SENT, alors il envoie un message I2 comme spécifié ci-dessous.
- o Dans tout autre état (I2-SENT, I2BIS-SENT, ESTABLISHED) alors l'hôte a déjà envoyé un message I2 et ceci est probablement une réponse à un message I1 retransmis, donc ce message R1 DOIT être éliminé en silence.

Quand l'hôte envoie un message I2, il inclut l'option Valideur de répondeur qui était dans le message R1. Le message I2 DOIT inclure la paire d'ULID -- normalement, dans les champs Source et Destination IPv6. Si une option Paire d'ULID était incluse dans le message I1, alors elle DOIT être incluse aussi dans le message I2. De plus, si la valeur d'identifiant d'instance fourchée pour ce contexte est non zéro, le message I2 DOIT contenir une option Identifiant d'instance fourchée portant la valeur d'identifiant d'instance fourchée. Par ailleurs, le message I2 contient un nom occasionnel d'initiateur. Il n'est pas exigé qu'il soit le même que celui inclus dans le précédent message I1.

Le message I2 peut aussi inclure la liste de localisateurs de l'initiateur. Si c'est le cas, alors il doit aussi inclure la structure de données de paramètres de CGA. Si CGA (et non HBA) est utilisé pour vérifier un ou plusieurs des localisateurs inclus dans la liste des localisateurs, alors l'initiateur doit aussi inclure une option Signature de CGA contenant la signature.

Quand le message I2 a été envoyé, l'état est réglé à I2-SENT.

7.12 Retransmission des messages I2

Si l'initiateur ne reçoit pas un message R2 après I2_TIMEOUT après l'envoi d'un message I2, il PEUT retransmettre le message I2, en utilisant un retard binaire exponentiel et des temporisateurs aléatoires. L'option Valideur de répondeur pourrait avoir une durée de vie limitée -- c'est-à-dire, l'homologue pourrait rejeter les options Valideur de répondeur qui sont plus anciennes que VALIDATOR_MIN_LIFETIME pour éviter des attaques en répétition. Dans le cas où l'initiateur décide de ne pas retransmettre les messages I2, ou dans le cas où l'initiateur ne reçoit toujours pas de message R2 après avoir retransmis des messages I2 I2_RETRIES_MAX fois, l'initiateur DEVRAIT revenir à la retransmission du message I1.

7.13 Réception des messages I2

Un hôte DOIT éliminer en silence tout message I2 reçu qui ne satisfait pas tous les essais de validité suivants en plus de ceux spécifiés au paragraphe 12.3 :

- o Le champ Longueur d'extension d'en-tête est au moins 2, c'est-à-dire, la longueur est au moins 24 octets.

À réception d'un message I2, l'hôte extrait la paire d'ULID et l'identifiant d'instance fourchée du message. Si il n'y a pas d'option Paire d'ULID, alor la paire d'ULID est prise des champs Source et Destination dans l'en-tête IPv6. Si il n'y a pas

d'option FII dans le message, alors la valeur de FII est prise comme étant zéro.

Ensuite, l'hôte vérifie que le nom occasionnel de répondeur est récent (les noms occasionnels qui ne sont pas plus vieux que VALIDATOR_MIN_LIFETIME DEVRAIENT être considérés comme récents) et que l'option Valideur de répondeur correspond au valideur que l'hôte a calculé pour l'ULID, les localisateurs, le nom occasionnel de répondeur, le nom occasionnel d'initiateur, et le FII.

Si une structure de données de paramètres de CGA (PDS) est incluse dans le message, l'hôte DOIT alors vérifier si la PDS actuelle contenue dans le message correspond à l'ULID(homologue).

Si une des vérifications ci-dessus échoue, alors l'hôte élimine en silence le message ; il a terminé le traitement I2.

Si toutes les vérifications ci-dessus sont réussies, l'hôte procède alors à une recherche d'état de contexte pour l'initiateur. L'hôte cherche un contexte avec la paire d'ULID et le FII extraits. Si il n'en existe pas, l'état du contexte (non existant) est vu comme étant IDLE ; donc, les actions dépendent de l'état comme suit :

- Si l'état est IDLE (c'est-à-dire, le contexte n'existe pas) l'hôte alloue une étiquette de contexte (CT(local)) crée l'état du contexte pour le contexte, et règle son état à ESTABLISHED. Il enregistre CT(homologue) et l'ensemble de localisateurs de l'homologue ainsi que son propre ensemble de localisateurs dans le contexte. Il DEVRAIT effectuer la vérification de HBA/CGA de l'ensemble de localisateurs de l'homologue à ce moment, comme spécifié au paragraphe 7.2. Puis l'hôte renvoie un message R2 comme spécifié ci-dessous.

- Si l'état est I1-SENT, l'hôte vérifie alors si le localisateur de source est inclus dans Ls(homologue) ou dans la liste de localisateurs contenue dans le message I2 et que la vérification de HBA/CGA pour cet ensemble spécifique de localisateurs est réussie.
  * Si ce n'est pas le cas, le message est alors éliminé en silence et l'état du contexte reste inchangé.
  * Si c'est le cas, alors l'hôte met à jour les informations de contexte (CT(homologue) Ls(homologue)) avec les données contenues dans le message I2, et l'hôte DOIT renvoyer un message R2 comme spécifié ci-dessous. Noter qu'avant de mettre à jour les informations de Ls(homologue) l'hôte DEVRAIT effectuer la validation de HBA/CGA de l'ensemble de localisateurs de l'homologue à ce moment, comme spécifié au paragraphe 7.2. L'hôte passe à l'état ESTABLISHED.

- Si l'état est ESTABLISHED, I2-SENT, ou I2BIS-SENT, alors l'hôte vérifie si le localisateur de source est inclus dans Ls(homologue) ou dans la liste de localisateurs contenue dans le message I2 et que la vérification de HBA/CGA pour ce localisateur spécifique est réussie.
  * Si ce n'est pas le cas, le message est alors éliminé en silence et l'état du contexte reste inchangé.
  * Si c'est le cas, alors l'hôte met à jour les informations de contexte (CT(homologue) Ls(homologue)) avec les données contenues dans le message I2, et l'hôte DOIT renvoyer un message R2 comme spécifié au paragraphe 7.14. Noter qu'avant de mettre à jour les informations de Ls(homologue) l'hôte DEVRAIT effectuer la validation de HBA/CGA de l'ensemble de localisateurs de l'homologue à ce moment, comme spécifié au paragraphe 7.2. L'état du contexte reste inchangé.

7.14 Envoi des messages R2

Avant que l'hôte envoie le message R2, il DOIT regarder si il y a une possible confusion de contexte, c'est-à-dire, si il pourrait finir avec plusieurs contextes utilisant le même CT(homologue) pour le même hôte homologue. Voir le paragraphe 7.15.

Quand l'hôte a besoin d'envoyer un message R2, il forme le message et son étiquette de contexte, et copie le nom occasionnel d'initiateur provenant du message déclencheur (I2, I2bis, ou I1). De plus, il peut inclure des localisateurs de remplacement et les options nécessaires afin que l'homologue puisse les vérifier. En particulier, le message R2 peut inclure la liste de localisateurs du répondeur et l'option PDS. Si CGA (et non HBA) est utilisé pour vérifier la liste des localisateurs, alors le répondeur signe aussi les parties clés du message et inclut une option Signature de CGA contenant la signature.

Les messages R2 ne sont jamais retransmis. Si le message R2 est perdu, l'initiateur va alors retransmettre un message I2/I2bis ou un message I1. L'une ou l'autre retransmission va faire que le répondeur trouve l'état du contexte et répond avec un message R2.

7.15 Confrontation pour confusion de contexte

Quand l'hôte reçoit un I2, I2bis, ou R2, il DOIT chercher une possible confusion de contexte, c'est-à-dire, où il finirait avec plusieurs contextes utilisant le même CT(homologue) pour le même hôte homologue. Cela peut arriver quand l'hôte a reçu les messages ci-dessus, car ils créent un nouveau contexte avec un nouveau CT(homologue). Le même problème s'applique quand CT(homologue) est mis à jour pour un contexte existant.

L'hôte prend CT(homologue) pour le contexte nouvellement créé ou mis à jour, et cherche un autre contexte qui :
- o soit dans l'état ESTABLISHED ou I2BIS-SENT
- o ait le même CT(homologue)
- o ait un Ls(homologue) qui ait au moins un localisateur en commun avec le nouveau contexte créé ou mis à jour.

Si un tel contexte est trouvé, l'hôte vérifie alors si la paire d'ULID ou l'identifiant d'instance fourchée sont différents de celui du contexte nouvellement créé ou mis à jour :
- o Si l'un ou l'autre ou les deux sont différents, alors l'homologue réutilise l'étiquette de contexte pour la création d'un contexte avec une paire d'ULID ou FII différent, ce qui est l'indication que l'homologue a perdu le contexte original. Dans ce cas, on est dans une situation de confusion de contexte, et l'hôte NE DOIT PAS utiliser le vieux contexte pour envoyer des paquets. Il PEUT juste éliminer le vieux contexte (après tout, l'homologue l'a éliminé) ou il PEUT tenter de rétablir le vieux contexte en envoyant un nouveau message I1 et en passant son état à I1-SENT. Dans tous les cas, une fois que cette situation est détectée, l'hôte NE DOIT PAS garder ces deux contexts avec des ensembles de localisateurs Ls(homologue) en chevauchement et les mêmes étiquette de contexte dans l'état ESTABLISHED, car il en résulterait des problèmes de démultiplexage chez l'homologue.
- o Si tous deux sont les mêmes, alors ce contexte est en fait le contexte qui est créé ou mis à jour ; donc, il n'y a pas de confusion.

7.16 Réception de messages R2

Un hôte DOIT éliminer en silence tout message R2 reçu qui ne satisfait pas tous les essais de validité suivants en plus de ceux spécifiés au paragraphe 12.3 :

- o Le champ Longueur d'extension d'en-tête fait au moins 1, c'est-à-dire, la longueur est au moins 16 octets.

À réception d'un message R2, l'hôte extrait du message le nom occasionnel d'initiateur et la paire de localisateurs (ce dernier des champs Source et Destination dans l'en-tête IPv6). Ensuite, l'hôte cherche un contexte existant qui corresponde au nom occasionnel d'initiateur et où les localisateurs sont Lp(homologue) et Lp(local) respectivement. Sur la base de l'état :

- o Si un tel contexte n'est pas trouvé, c'est-à-dire, l'état est IDLE, alors le message est éliminé en silence.

- o Si l'état est I1-SENT, I2-SENT, ou I2BIS-SENT, alors l'hôte effectue les actions suivantes. Si une structure de données de paramètres de CGA (PDS) est incluse dans le message, l'hôte DOIT alors vérifier que la PDS actuelle contenue dans le message correspond à l'ULID(homologue) comme spécifié au paragraphe 7.2. Si la vérification échoue, alors le message est éliminé en silence. Si la vérification réussit, l'hôte enregistre alors les informations provenant du message R2 dans l'état du contexte ; il enregistre l'ensemble de localisateurs de l'homologue et CT(homologue). L'hôte DEVRAIT effectuer la vérification de HBA/CGA de l'ensemble de localisateurs de l'homologue à ce moment, comme spécifié au paragraphe 7.2. L'hôte règle son état à ESTABLISHED.

- o Si l'état est ESTABLISHED, le message R2 est ignoré en silence, (car c'est probablement une réponse à un message I2 retransmis).

Avant que l'hôte achève le traitement R2, il DOIT chercher une possible confusion de contexte, c'est-à-dire, où il finirait avec plusieurs contextes utilisant le même CT(homologue) pour le même hôte homologue. Voir le paragraphe 7.15.

7.17 Envoi des messages R1bis

À réception d'un en-tête Extension de charge utile Shim6 où il n'y a pas de contexte Shim6 en cours chez le receveur, le receveur va répondre avec un message R1bis afin de permettre un rétablissement rapide du contexte Shim6 perdu.

Aussi, un hôte va répondre avec un R1bis à réception de tout message de contrôle qui a un type de message dans la gamme de 64 à 127 (c'est-à-dire, excluant les messages d'établissement de contexte tels que I1, R1, R1bis, I2, I2bis, R2, et leurs futures extensions) où le message de contrôle se réfère à un contexte non existant.

On suppose que tous les paquets entrants qui déclenchent la génération d'un message R1bis contiennent une paire de localisateurs (dans les champs d'adresse de l'en-tête IPv6) et une étiquette de contexte.

À réception de tout paquet décrit ci-dessus, l'hôte va répondre avec un R1bis incluant les informations suivantes :
- o Le nom occasionnel de répondeur est un nombre pris par le répondeur que l'initiateur va retourner dans le message I2bis.
- o L'étiquette de contexte de paquet est l'étiquette de contexte contenue dans le paquet reçu qui a déclenché la génération du message R1bis.
- o L'option Valideur de répondeur est incluse, avec un valideur qui est calculé comme suggéré au paragraphe suivant.

7.17.1 Génération du valideur R1bis

Une façon pour que le répondeur génère correctement les valideurs est de maintenir un seul secret (S) et un compteur C fonctionnant pour le nom occasionnel de répondeur, incrémenté à des intervalles de temps fixés (cela permet au répondeur de vérifier l'âge d'un nom occasionnel de répondeur, indépendamment du contexte dans lequel il est utilisé).

Quand le valideur est généré pour être inclus dans un message R1bis -- c'est-à-dire, envoyé en réponse à un paquet de contrôle spécifique ou à un paquet contenant le message En-tête Extension de charge utile Shim6 -- le répondeur peut effectuer la procédure suivante pour générer la valeur du valideur :

D'abord, le répondeur utilise la valeur du compteur C comme nom occasionnel de répondeur.

Ensuite, il utilise les informations suivantes (enchaînées) comme entrées à la fonction unidirectionnelle :
- o Le secret S
- o Ce nom occasionnel de répondeur
- o L'étiquette de contexte de receveur incluse dans le paquet reçu
- o Les localisateurs provenant du paquet reçu.

Enfin, il utilise le résultat de la fonction de hachage comme chaîne de valideur.

7.18 Réception des messages R1bis et envoi des messages I2bis

Un hôte DOIT éliminer en silence tout message R1bis reçu qui ne satisfait pas toutes les vérifications de validité suivantes en plus de celles spécifiées au paragraphe 12.3 :

- o Le champ Longueur d'extension d'en-tête est au moins 1, c'est-à-dire, la longueur est au moins 16 octets.

À réception d'un message R1bis, l'hôte extrait l'étiquette de contexte de paquet et la paire de localisateurs du message (cette dernière des champs Source et Destination dans l'en-tête IPv6). Ensuite, l'hôte cherche un contexte existant où l'étiquette de contexte de paquet correspond à CT(homologue) et où les localisateurs correspondent respectivement à Lp(homologue) et Lp(local).

- o Si un tel contexte n'est pas trouvé, c'est-à-dire, si l'état est IDLE, alors le message R1bis est éliminé en silence.

- o Si l'état est I1-SENT, I2-SENT, ou I2BIS-SENT, alors le message R1bis est éliminé en silence.

- o Si l'état est ESTABLISHED, alors on est dans le cas où l'homologue a perdu le contexte, et le but est d'essayer de le rétablir. Pour cela, l'hôte laisse CT(homologue) inchangé dans l'état du contexte, transite à l'état I2BIS-SENT, et envoie un message I2bis, incluant l'option calculée Valideur de répondeur, l'étiquette de contexte de paquet, et le nom occasionnel de répondeur qui ont été reçus dans le message R1bis. Ce message R1bis est envoyé en utilisant la paire de localisateurs incluse dans le message R1bis. Dans le cas où cette paire de localisateurs diffère de la paire d'ULID définie pour ce contexte, alors une option ULID DOIT être incluse dans le message I2bis. De plus, si l'identifiant d'instance fourchée pour ce contexte n'est pas zéro, alors une option Identifiant d'instance fourchée portant la valeur d'identifiant d'instance pour ce contexte DOIT être incluse dans le message I2bis. Le message I2bis peut aussi inclure une liste de localisateurs. Si c'est le cas, il doit alors aussi inclure la structure de données de paramètres de CGA. Si CGA (et non HBA) est utilisé pour vérifier un ou plusieurs des localisateurs inclus dans la liste des localisateurs, alors l'initiateur doit aussi inclure une option Signature de CGA contenant la signature.

7.19 Retransmission des messages I2bis

Si l'initiateur ne reçoit pas de message R2 après I2bis_TIMEOUT après l'envoi d'un message I2bis, il PEUT retransmettre le message I2bis, en utilisant un retard binaire exponentiel et des temporisateurs aléatoires. L'option Valideur de répondeur pourrait avoir une durée de vie limitée -- c'est-à-dire, l'homologue pourrait rejeter les options Valideur de répondeur qui sont plus vieilles que VALIDATOR_MIN_LIFETIME pour éviter des attaques en répétition. Dans le cas où l'initiateur décide de ne pas retransmettre les messages I2bis, ou dans le cas où l'initiateur ne reçoit toujours pas de message R2 après la retransmission des messages I2bis I2bis_RETRIES_MAX fois, l'initiateur DEVRAIT revenir à la retransmission du message I1.

7.20 Réception des messages I2bis et envoi des messages R2

Un hôte DOIT éliminer en silence tout message I2bis reçu qui ne satisfait pas toutes les vérifications de validité suivantes en plus de celles spécifiées au paragraphe 12.3 :

- Le champ Longueur d'extension d'en-tête fait au moins 3, c'est-à-dire, la longueur est au moins 32 octets.

À réception d'un message I2bis, l'hôte extrait la paire d'ULID et l'identifiant d'instance fourchée du message. Si il n'y a pas d'option Paire d'ULID, alors la paire d'ULID est prise des champs Source et Destination dans l'en-tête IPv6. Si il n'y a pas d'option FII dans le message, alors la valeur FII est prise comme zéro.

Ensuite, l'hôte vérifie que le nom occasionnel de répondeur est récent (les noms occasionnels qui ne sont pas plus anciens que VALIDATOR_MIN_LIFETIME DEVRAIENT être considérés comme récents) et que l'option Valideur de répondeur correspond au valideur que l'hôte aurait calculé pour les localisateurs, nom occasionnel de répondeur, et étiquette de contexte de receveur au titre de l'envoi d'un message R1bis.

Si une structure de données de paramètres de CGA (PDS) est incluse dans le message, alors l'hôte DOIT vérifier si la PDS actuelle contenue dans le message correspond à l'ULID(homologue).

Si une des vérifications ci-dessus échoue, alors l'hôte élimine en silence le message ; il a terminé le traitement du I2bis.

Si les deux vérifications sont réussies, alors l'hôte procède à la recherche d'un état de contexte pour l'initiateur. L'hôte cherche un contexte avec la paire d'ULID et les FII extraites. Si il n'en existe pas, alors l'état du contexte (non existant) est vu comme étant IDLE ; donc, les actions dépendent de l'état comme suit :

- Si l'état est IDLE (c'est-à-dire, si le contexte n'existe pas) l'hôte alloue une étiquette de contexte (CT(local)) crée l'état du contexte pour le contexte, et règle son état à ESTABLISHED. L'hôte NE DEVRAIT PAS utiliser l'étiquette de contexte de paquet dans le message I2bis pour CT(local) ; à la place, il devrait prendre une étiquette de contexte aléatoire nouvelle juste comme quand il traite un message I2. Il enregistre CT(homologue) et l'ensemble de localisateurs de l'homologue ainsi que son propre ensemble de localisateurs dans le contexte. Il DEVRAIT effectuer la vérification de HBA/CGA de l'ensemble de localisateurs de l'homologue à ce moment, comme spécifié au paragraphe 7.2. Ensuite l'hôte renvoie un message R2 comme spécifié au paragraphe 7.14.

- Si l'état est I1-SENT, alors l'hôte vérifie si le localisateur de source est inclus dans Ls(homologue) ou dans la liste de localisateurs contenue dans le message I2bis et si la vérification de HBA/CGA pour ce localisateur spécifique est réussi.
  * Si ce n'est pas le cas, le message est éliminé en silence. L'état du contexte reste inchangé.
  * Si c'est le cas, l'hôte met à jour les informations de contexte (CT(homologue) Ls(homologue)) avec les données contenues dans le message I2bis, et l'hôte DOIT renvoyer un message R2 comme spécifié ci-dessous. Noter qu'avant de mettre à jour les informations de Ls(homologue) l'hôte DEVRAIT effectuer la validation de HBA/CGA de l'ensemble de localisateurs de l'homologue à ce moment, comme spécifié au paragraphe 7.2. L'hôte passe à l'état ESTABLISHED.

- Si l'état est ESTABLISHED, I2-SENT, ou I2BIS-SENT, alors l'hôte détermine si au moins une des deux conditions tient : i) si le localisateur de source est inclus dans Ls(homologue) ou, ii) si le localisateur de source est inclus dans la liste de localisateurs contenue dans le message I2bis et si la vérification de HBA/CGA pour ce localisateur spécifique est réussie.
  * Si aucune des deux conditions susmentionnées ne tient, alors le message est éliminé en silence. L'état du contexte reste inchangé.
  * Si au moins une des deux conditions susmentionnées tient, alors l'hôte met à jour les informations de contexte (CT(homologue) Ls(homologue)) avec les données contenues dans le message I2bis, et l'hôte DOIT renvoyer un message R2, comme spécifié au paragraphe 7.14. Noter qu'avant de mettre à jour les informations de Ls(homologue)

l'hôte DEVRAIT effectuer la validation de HBA/CGA de l'ensemble de localisateurs de l'homologue à ce moment, comme spécifié au paragraphe 7.2. L'état du contexte reste inchangé.

8. Traitement des messages d'erreur ICMP

Les routeurs dans le chemin ainsi que la destination pourraient générer des messages d'erreur ICMP. Dans certains cas, le Shim6 peut prendre des actions et résoudre le problème qui a résulté en l'erreur. Dans d'autres cas, la couche Shim6 ne peut pas résoudre le problème, et il est critique que ces paquets reviennent aux ULP afin qu'ils puissent prendre l'action appropriée.

C'est une question de mise en œuvre dans le sens où le mécanisme est complètement local pour l'hôte lui-même. Mais la question de comment les erreurs ICMP sont correctement communiquées à l'ULP sur l'hôte est importante ; donc, cette section spécifie la question.

Tous les messages ICMP DOIVENT être livrés à l'ULP dans tous les cas, sauf quand Shim6 agit avec succès sur le message (par exemple, choisit un nouveau chemin). Il DEVRAIT y avoir une option de configuration pour livrer sans condition tous les messages ICMP (y compris ceux sur lesquels agit shim6) à l'ULP.

En accord avec cette recommandation, les messages d'erreur ICMP suivants devraient être traités par la couche Shim6 et non pas passés à l'ULP :

Erreur ICMP Destination injoignable, avec les codes :
- 0 (Pas de chemin pour la destination)
- 1 (Communication avec la destination administrativement interdite)
- 2 (Au delà de la portée de l'adresse de source)
- 3 (Adresse injoignable)
- 5 (L'adresse de source ne satisfait pas la politique d'entrée/sortie)
- 6 (Rejet du chemin pour la destination)

Erreur ICMP Temps excédé.

Erreur ICMP Problème de paramètre, avec le paramètre qui a causé l'erreur étant un paramètre Shim6.

Les messages d'erreur ICMP suivants rapportent des problèmes qui ne peuvent pas être traités par la couche Shim6 et qui devraient être passés à l'ULP (comme décrit ci-dessous) :

Erreur ICMP Paquet trop gros.

Erreur ICMP Destination injoignable avec code 4 (Accès injoignable).

Erreur ICMP Problème de paramètre (si le paramètre qui a causé le problème n'est pas un paramètre Shim6).

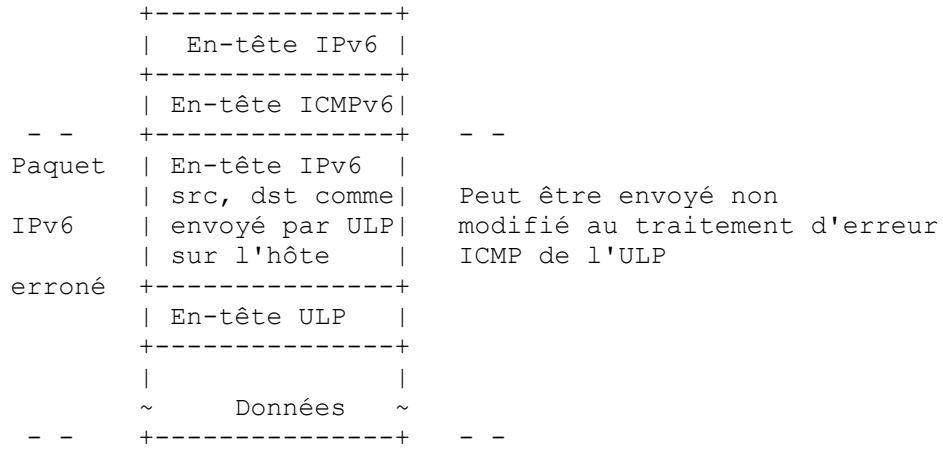


Figure 8 : Traitement d'erreur ICMP sans l'en-tête Extension de charge utile Shim6

Quand les paquets d'ULP sont envoyés sans l'en-tête Extension de charge utile Shim6 -- c'est-à-dire, alors que les localisateurs=ULID initiaux fonctionnent -- cela n'introduit pas de nouveau problème ; le mécanisme existant de mise en œuvre pour livrer ces erreurs à l'ULP va fonctionner. Voir la Figure 8.

Mais quand Shim sur le côté émetteur insère l'en-tête Extension de charge utile Shim6 et remplace les ULID dans le champ Adresse IP par d'autres localisateurs, alors une erreur ICMP revenant aura un "Paquet erroné", qui n'est pas un paquet que l'ULP a envoyé. Donc, la mise en œuvre va devoir appliquer une transposition inverse au "Paquet erroné" avant de passer l'erreur ICMP à l'ULP, incluant les extensions ICMP définies dans la [RFC4884]. Voir la Figure 9.

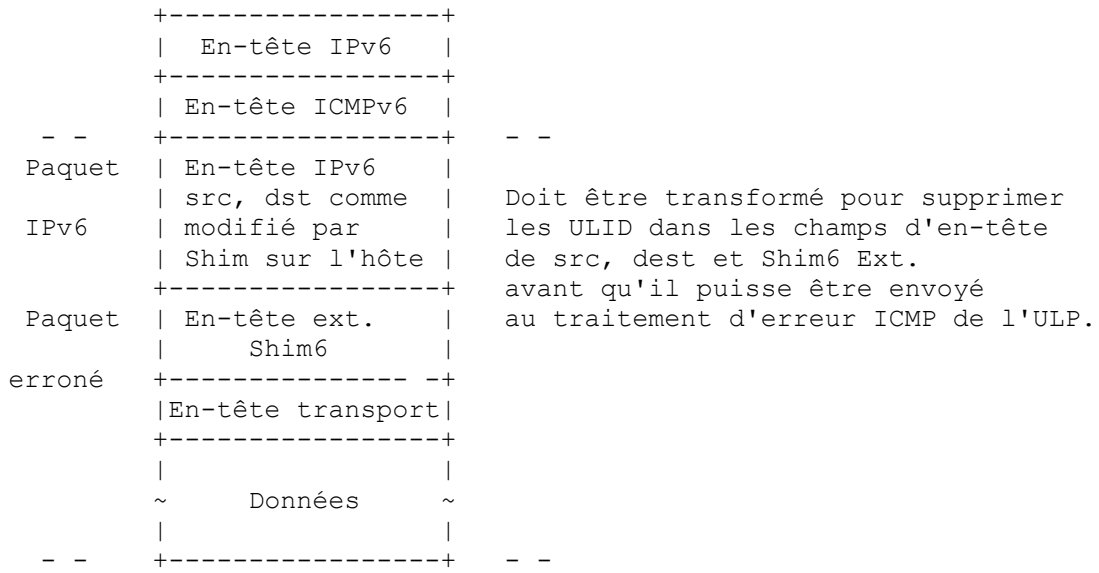


Figure 9 : Traitement d'erreur ICMP avec l'en-tête Extension de charge utile Shim6

Noter que cette transposition est différente de celle où on reçoit des paquets de l'homologue avec des en-têtes Extension de charge utile Shim6 parce que, dans ce cas, les paquets contiennent CT(local). Mais les erreurs ICMP ont un "paquet erroné" avec un en-tête Extension de charge utile Shim6 contenant CT(homologue). C'est parce que ils étaient destinés à être reçus par l'homologue. Dans tous les cas, comme les champs <Localisateur de source, Localisateur de destination, CT(homologue)> doivent être uniques quand ils sont reçus par l'homologue, l'hôte local devrait aussi seulement être capable de trouver un contexte qui corresponde à ce triplet.

Si l'erreur ICMP est un "paquet trop gros", la MTU rapportée doit être ajustée à 8 octets de moins, car le Shim va ajouter 8 octets quand il envoie des paquets.

Après que "Paquet erroné" a eu les ULID originaux insérés, alors cet en-tête Extension de charge utile Shim6 peut être supprimé. Le résultat est un "Paquet erroné" qui est passé à l'ULP comme si Shim n'existait pas.

9. Suppression du contexte Paire d'ULID

Chaque hôte peut décider unilatéralement quand supprimer un contexte de paire d'ULID. Il est RECOMMANDÉ que les hôtes ne suppriment pas le contexte quand ils savent qu'il y a un protocole de couche supérieure qui pourrait utiliser le contexte. Par exemple, une mise en œuvre pourrait savoir si il y a une prise ouverte qui est connectée au ULID(homologue). Cependant, il pourrait y avoir des cas où la connaissance n'est pas directement disponible à la couche Shim, par exemple, pour les applications UDP qui ne connectent pas leurs prises ou pour toute application qui conserve de l'état de niveau supérieur à travers les connexions (TCP) et les paquets UDP.

Donc, il est RECOMMANDÉ que les mises en œuvre minimisent la suppression prématurée en observant la quantité de trafic envoyé et reçu en utilisant le contexte, et ne suppriment l'état qu'après qu'il apparaît en repos. Une approche raisonnable serait de ne pas supprimer un contexte avant qu'au moins 5 minutes soit écoulées depuis le dernier message envoyé ou reçu en utilisant le contexte. (Noter que les paquets qui utilisent la paire d'ULID comme paire de localisateurs et n'exigent pas de réécriture d'adresse par la couche Shim6 sont aussi considérés comme des paquets qui utilisent le contexte

Shim6 associé.)

Comme il n'y a pas de suppression explicite coordonnée de l'état du contexte, il y a des problèmes potentiels autour de la réutilisation de l'étiquette de contexte. Une extrémité pourrait supprimer l'état et potentiellement réutiliser cette étiquette de contexte pour une autre communication, et l'homologue pourrait ensuite essayer d'utiliser le vieux contexte (qu'il n'a pas supprimé). Le protocole a des mécanismes pour récupérer de cela, qui fonctionnent que la suppression d'état ait été totale et accidentelle (par exemple, panne et réamorçage de l'hôte) ou juste une mauvaise collecte de l'état Shim qui ne semblait pas être utilisé. Cependant, l'hôte devrait essayer de minimiser la réutilisation des étiquettes de contexte en essayant de parcourir au hasard les 2^47 valeurs d'étiquettes de contexte. (Voir à l'Appendice C un résumé de comment fonctionne la récupération dans les différents cas.)

10. Mise à jour de l'homologue

La demande de mise à jour et l'accusé de réception sont tous deux utilisés pour la mise à jour de la liste des localisateurs (seulement possible quand CGA est utilisé pour vérifier le ou les localisateurs) et pour la mise à jour des préférences associées à chaque localisateur.

10.1 Envoi des messages de demande de mise à jour

Quand un hôte a un changement dans l'ensemble de localisateurs, il peut le communiquer à l'homologue en envoyant une demande de mise à jour. Quand un hôte a un changement des préférences pour son ensemble de localisateurs, il peut aussi le communiquer à l'homologue. Le message Demande de mise à jour peut inclure juste une option Liste de localisateurs (pour porter le nouvel ensemble de localisateurs) juste une option Préférences de localisateurs, ou à la fois une nouvelle Liste de localisateurs et de nouvelles Préférences de localisateurs.

Si l'hôte envoie une nouvelle liste de localisateurs, l'hôte prend un nouveau nombre aléatoire de génération locale, l'enregistre dans le contexte, et le met dans l'option Liste de localisateurs. Toute option Préférence de localisateurs, envoyée dans la même demande de mise à jour ou dans une future demande de mise à jour, va utiliser de nombre de génération pour s'assurer que les préférences sont appliquées à la version correcte de liste des localisateurs.

L'hôte prend un nom occasionnel de demande aléatoire pour chaque mise à jour et garde le même nom occasionnel pour chaque retransmission de la demande de mise à jour. Le nom occasionnel est utilisé pour confronter l'accusé de réception avec la demande.

Le message Demande de mise à jour peut aussi inclure une structure de données de paramètres de CGA (c'est nécessaire si la PDS de CGA n'avait pas été échangée précédemment). Si CGA (et non HBA) est utilisé pour vérifier un ou plusieurs des localisateurs inclus dans la liste des localisateurs, alors une option Signature de CGA contenant la signature doit aussi être incluse dans le message Demande de mise à jour.

10.2 Retransmission des messages de demande de mise à jour

Si l'hôte ne reçoit pas de message R2 d'accusé de réception de mise à jour en réponse au message Demande de mise à jour après UPDATE_TIMEOUT, il doit alors retransmettre le message Demande de mise à jour. Les retransmissions devraient utiliser un temporisateur de retransmission avec un retard binaire exponentiel pour éviter de créer des problèmes d'encombrement au réseau quand de grandes quantités d'hôtes effectuent des retransmissions de demande de mise à jour. Aussi, la valeur réelle du temporisateur devrait être aléatoire entre 0,5 et 1,5 de la valeur nominale pour éviter l'auto-synchronisation.

Si il n'y a pas de réponse, les retransmissions continuent indéfiniment. Le retard binaire exponentiel s'arrête à MAX_UPDATE_TIMEOUT. Mais la seule façon dont les retransmissions devraient s'arrêter quand il n'y a pas d'accusé de réception est quand Shim6, par le protocole REAP ou quelque autre mécanisme, décide d'éliminer l'état du contexte du fait du manque d'usage de l'ULP en combinaison avec l'absence de réponses au protocole REAP.

10.3 Informations plus récentes à la retransmission

Il ne peut y avoir au plus qu'un message Demande de mise à jour en instance à tout moment. Donc jusqu'à ce que, par exemple, une mise à jour avec une nouvelle liste de localisateurs ait été acquittée, aucune nouvelle liste de localisateurs ou nouvelles préférences de localisateurs ne peut être envoyée. Cependant, quand il y a de nouvelles informations et que les

anciennes informations n'ont pas encore été acquittées, l'hôte peut, au lieu d'attendre un accusé de réception, abandonner la précédente mise à jour et construire une nouvelle demande de mise à jour (avec un nouveau nom occasionnel de demande) qui inclut les nouvelles informations aussi bien que celles qui n'ont pas encore été acquittées.

Par exemple, si la liste originale de localisateurs était juste (A1, A2) et si une demande de mise à jour avec la liste de localisateurs (A1, A3) est en instance, et si l'hôte détermine qu'il devrait ajouter A4 à la liste des localisateurs et marquer A1 comme BROKEN, il va alors devoir :
- o Prendre un nouveau nom occasionnel de demande aléatoire pour la nouvelle demande de mise à jour.
- o Prendre un nouveau nombre de génération aléatoire pour la nouvelle liste de localisateurs.
- o Former la nouvelle liste de localisateurs : (A1, A3, A4).
- o Former une option Préférence de localisateurs qui utilise le nouveau nombre de génération et a le fanion BROKEN pour le premier localisateur.
- o Envoyer la demande de mise à jour et lancer un temporisateur de retransmission.

Tout accusé de réception de mise à jour qui ne correspond pas au nom occasionnel de demande actuel (par exemple, un accusé de réception pour la demande de mise à jour abandonnée) va être ignoré en silence.

10.4 Réception des messages de demande de mise à jour

Un hôte DOIT éliminer en silence tout message de demande de mise à jour reçu qui ne satisfait pas toutes les vérifications de validité suivantes en plus de celles spécifiées au paragraphe 12.3:

- o Le champ Longueur d'extension d'en-tête est au moins 1, c'est-à-dire, la longueur est au moins 16 octets.

À réception d'un message Demande de mise à jour, l'hôte extrait l'étiquette de contexte du message. Il cherche alors un contexte qui ait un CT(local) correspondant à l'étiquette de contexte. Si un tel contexte n'est pas trouvé, il envoie un message R1bis comme spécifié au paragraphe 7.17.

Comme les étiquettes de contexte peuvent être réutilisées, l'hôte DOIT vérifier que le champ Adresse IPv6 de source fait partie de Ls(homologue) et que le champ Adresse IPv6 de destination fait partie de Ls(local). Si ce n'est pas le cas, l'envoyeur de la demande de mise à jour a un contexte périmé qui se trouve correspondre au CT(local) pour ce contexte. Dans ce cas, l'hôte DOIT envoyer un message R1bis et autrement ignorer le message Demande de mise à jour.

Si une structure de données de paramètres de CGA (PDS) est incluse dans le message, alors l'hôte DOIT vérifier si la PDS actuelle contenue dans le paquet correspond à l'ULID(homologue). Si cette vérification échoue, le message est éliminé en silence.

Ensuite, selon l'état du contexte :
- o Si c'est ESTABLISHED, il procède au traitement du message.
- o Si c'est I1-SENT, il élimine le message et reste dans l'état I1-SENT.
- o Si c'est I2-SENT, il envoie I2 et procède au traitement du message.
- o Si c'est I2BIS-SENT, il envoie I2bis et procède au traitement du message.

Les questions de vérification pour la portée des localisateurs dans le message Demande de mise à jour sont spécifiées au paragraphe 7.2. Si la liste des localisateurs ne peut pas être vérifiée, cette procédure devrait envoyer un message d'erreur Shim6 avec Code d'erreur=2. Dans tous les cas, si elle ne peut pas être vérifiée, il n'y a pas d'autre traitement de la demande de mise à jour.

Une fois que toute option Liste de localisateurs dans la demande de mise à jour a été vérifiée, le nombre de génération de l'homologue dans le contexte est mis à jour pour être celui de l'option Liste de localisateurs.

Si le message Demande de mise à jour contient une option Préférence de localisateurs, alors le nombre de génération dans l'option de préférence est comparé au nombre de génération de l'homologue dans le contexte. Si ils ne correspondent pas, l'hôte génère alors un message d'erreur Shim6 avec Code d'erreur=3 et avec le champ Pointeur se référant au premier octet de la génération de liste de localisateurs : nombre dans l'option Préférence de localisateurs. De plus, si le nombre d'éléments dans l'option Préférence de localisateurs ne correspond pas au nombre de localisateurs dans Ls(homologue) alors un message d'erreur Shim6 avec Code d'erreur=4 est envoyé avec le champ Pointeur se référant au premier octet du champ Longueur dans l'option Préférence de localisateurs. Dans les deux cas d'échec, aucun autre traitement n'est effectué sur le message Demande de mise à jour.

Si le nombre de générations correspond, les préférences de localisateurs sont enregistrées dans le contexte.

Une fois que l'option Liste de localisateurs (si elle est présente) a été vérifiée et que toute nouvelle liste de localisateurs ou préférences de localisateurs a été enregistrée, l'hôte envoie un message d'accusé de réception de mise à jour, en copiant le nom occasionnel provenant de la demande et en utilisant le CT(homologue) comme étiquette de contexte de receveur.

Tout nouveau localisateur (ou, plus probablement, nouvelles préférences de localisateur) pourrait résulter en ce que l'hôte veuille choisir une paire différente de localisateurs pour le contexte -- par exemple, si l'option Préférences de localisateurs fait la liste du Lp(homologue) courant comme BROKEN. L'hôte utilise la procédure d'exploration d'accessibilité décrite dans la [RFC5534] pour vérifier que le nouveau localisateur est accessible avant de changer le Lp(homologue).

10.5 Réception des messages Accusé de réception de mise à jour

Un hôte DOIT éliminer en silence tout message reçu d'accusé de réception de mise à jour qui ne satisfait pas toutes les vérifications de validité suivantes en plus de celles spécifiées au paragraphe 12.3:

- Le champ Longueur d'extension d'en-tête est au moins 1, c'est-à-dire, la longueur est au moins 16 octets.

À réception d'un message d'accusé de réception de mise à jour, l'hôte extrait l'étiquette de contexte et le nom occasionnel de demande du message. Il cherche ensuite un contexte qui a un CT(local) correspondant à l'étiquette de contexte. Si aucun contexte de ce type n'est trouvé, il envoie un message R1bis comme spécifié au paragraphe 7.17.

Comme les étiquettes de contexte peuvent être réutilisées, l'hôte DOIT vérifier que le champ Adresse IPv6 de source fait partie de Ls(homologue) et que le champ Adresse IPv6 de destination fait partie de Ls(local). Si ce n'est pas le cas, l'envoyeur de l'accusé de réception de mise à jour a un contexte périmé qui se trouve correspondre à CT(local) pour ce contexte. Dans ce cas, l'hôte DOIT envoyer un message R1bis et autrement ignorer le message d'accusé de réception de mise à jour.

Ensuite, selon l'état du contexte :
- Si c'est ESTABLISHED, il procède au traitement du message.
- Si c'est I1-SENT, il élimine le message et reste dans l'état I1-SENT.
- Si c'est I2-SENT, il envoie R2 et procède au traitement du message.
- Si c'est I2BIS-SENT, il envoie R2 et procède au traitement du message.

Si le nom occasionnel de demande ne correspond pas au nom occasionnel pour la dernière demande de mise à jour envoyée pour le contexte, alors l'accusé de réception de mise à jour est ignoré en silence. Si le nom occasionnel correspond, alors la mise à jour a été achevée et le temporisateur de retransmission de mise à jour peut être relancé.

11. Envoi des charges utiles d'ULP

Quand il n'y a pas d'état de contexte pour la paire d'ULID chez l'envoyeur, cela n'a pas d'effet sur la façon dont les paquets d'ULP sont envoyés. Si l'hôte utilise une heuristique pour déterminer quand effectuer un établissement de contexte différé, alors l'hôte pourrait avoir besoin de faire un compte (compte du nombre de paquets envoyés et reçus) même avant qu'il y ait un contexte de paire d'ULID.

Si le contexte n'est pas dans l'état ESTABLISHED ou I2BIS-SENT, cela n'a pas non plus d'effet sur la façon dont les paquets d'ULP sont envoyés. C'est seulement dans les états ESTABLISHED et I2BIS-SENT que l'hôte a CT(homologue) et Ls(homologue) établis.

Si il y a un contexte de paire d'ULID pour la paire d'ULID, alors l'envoyeur doit vérifier si le contexte utilise les ULID comme localisateurs -- c'est-à-dire si Lp(homologue) == ULID(homologue) et Lp(local) == ULID(local).

Si c'est le cas, alors les paquets peuvent être envoyés sans modification par le Shim. Si ce n'est pas le cas, alors la logique du paragraphe 11.1 va devoir être utilisée.

Il y aura aussi des activités de maintenance relatives à la détection d'(in)accessibilité, si les paquets sont ou non envoyés avec les localisateurs originaux. Les détails de cela sortent du domaine d'application de ce document et sont spécifiés dans la [RFC5534].

11.1 Envoi de charge utile d'ULP après une commutation

Quand il envoie des paquets, si il y a un contexte de paire d'ULID pour la paire d'ULID, et si la paire d'ULID n'est plus utilisée comme paire de localisateurs, alors l'envoyeur doit transformer le paquet. À part remplacer les champs IPv6 de source et de destination par une paire de localisateurs, un en-tête de 8 octets est ajouté afin que le receveur puisse trouver le contexte et inverser la transformation.

Si il y a eu une défaillance causant une commutation, et si plus tard le contexte revient à l'envoi de choses en utilisant la paire d'ULID comme paire de localisateurs, il n'est alors plus besoin de faire de transformation de paquet par l'envoyeur ; donc, il n'est pas nécessaire d'inclure l'en-tête Extension de 8 octets.

D'abord, les champs d'adresse IP sont remplacés. Le champ Adresse IPv6 de source est réglé à Lp(local) et le champ Adresse de destination est réglé à Lp(homologue). Noter que ceci NE DOIT PAS causer de nouveau calcul des sommes de contrôle ULP, car les sommes de contrôle ULP sont portées de bout en bout et le pseudo en-tête ULP contient les ULID qui sont préservés de bout en bout.

L'envoyeur saute tous les "en-têtes d'extension de sous couche d'acheminement" que l'ULP pourrait avoir inclus ; donc, il saute tout en-tête Extension bond par bond, tout en-tête d'acheminement, et tout en-tête d'options de destination qui sont suivis par un en-tête d'acheminement. Après tout en-tête de ce type, l'en-tête Extension Shim6 va être ajouté. Cela pourrait être avant un en-tête de fragment, un en-tête Options de destination, un en-tête ESP ou AH, ou un en-tête d'ULP.

L'en-tête Extension de charge utile Shim6 inséré inclut l'étiquette de contexte de l'homologue. Il prend la valeur de prochain en-tête de l'en-tête Extension précédent, car cet en-tête Extension va avoir une valeur de prochain en-tête de Shim6.

12. Réception des paquets

Le côté receveur de la communication peut recevoir des paquets associés à un contexte Shim6, avec ou sans l'en-tête Extension Shim6. Dans le cas où la paire d'ULID est utilisée comme paire de localisateurs, les paquets reçus n'auront pas l'en-tête Extension Shim6 et vont être traités pas la couche Shim6 comme décrit ci-dessous. Si le paquet reçu ne porte pas l'en-tête Extension Shim6, comme dans le traitement normal de paquet par le côté IPv6 receveur, le receveur analyse les en-têtes (d'extension) dans l'ordre. Si il trouve l'en-tête Extension Shim6, il va chercher le champ "P" dans cet en-tête. Si ce bit est à zéro, alors le paquet doit être passé au traitement de charge utile Shim6 pour être réécrit. Autrement, le paquet est passé au traitement de contrôle Shim6.

12.1 Réception de charge utile sans en-tête d'extension

Le receveur extrait les champs IPv6 de source et de destination et les utilise pour trouver un contexte de paire d'ULID, tel que les champs d'adresse IPv6 correspondent à l'ULID(local) et l'ULID(homologue). Si un tel contexte est trouvé, le contexte ne paraît pas être au repos ; cela devrait être mémorisé afin d'éviter de supprimer le contexte et pour la détection d'accessibilité, comme décrit dans la [RFC5534]. L'hôte continue avec le traitement normal du paquet IP.

12.2 Réception d'en-têtes d'extension de charge utile Shim6

Le receveur extrait l'étiquette de contexte de l'en-tête Extension de charge utile Shim6 et l'utilise pour trouver un contexte de paire d'ULID. Si aucun contexte n'est trouvé, le receveur DEVRAIT générer un message R1bis (paragraphe 7.17).

Ensuite, selon l'état du contexte :
- Si c'est ESTABLISHED, il procède au traitement du message.
- Si c'est I1-SENT, il élimine le message et reste à l'état I1-SENT.
- Si c'est I2-SENT, il envoie I2 et procède au traitement du message.
- Si c'est I2BIS-SENT, il envoie I2bis et procède au traitement du message.

Avec ce contexte, le receveur peut maintenant remplacer les champs d'adresse IP par les ULID conservés dans le contexte. Finalement, l'en-tête Extension de charge utile Shim6 est supprimé du paquet (afin que l'ULP ne soit pas dans la confusion à cause de lui) et la valeur de prochain en-tête dans l'en-tête précédant est réglée au numéro de protocole actuel pour la charge utile. Ensuite, le paquet peut être passé au protocole identifié par la valeur de prochain en-tête (qui pourrait être une fonction associée à la sous couche IP du point d'extrémité ou à un ULP).

Si l'hôte utilise une heuristique pour déterminer quand effectuer un établissement de contexte différé, alors l'hôte pourrait

avoir besoin d'une certaine comptabilité (compte du nombre de paquets envoyés et reçus) pour les paquets qui n'ont pas d'en-tête Extension Shim6 et pour lesquels il n'y a pas de contexte. Mais ce besoin dépend de l'heuristique choisie par la mise en œuvre.

12.3 Réception de messages de contrôle Shim

Un message de contrôle Shim a son champ Somme de contrôle vérifié. Le champ Longueur d'en-tête Shim est aussi vérifié par rapport à la longueur du paquet IPv6 pour s'assurer que le message Shim ne demande pas de finir au delà de l'extrémité du paquet IPv6. Finalement, il vérifie que ni le champ Destination IPv6 ni le champ Source IPv6 n'est une adresse de diffusion groupée ou une adresse non spécifiée. Si une de ces vérifications échoue, le paquet est éliminé en silence.

Le message est alors distribué sur la base du type du message Shim. Chaque type de message est alors traité comme décrit ailleurs dans ce document. Si le paquet contient un type de message Shim qui est inconnu du receveur, alors un message d'erreur Shim6 avec le code d'erreur 0 est généré et renvoyé. Le champ Pointeur est réglé à pointer sur le premier octet du type de message Shim.

Tous les messages de contrôle peuvent contenir toute option avec C=0. Si il y a une option dans le message avec C=1 qui n'est pas connue de l'hôte, celui-ci DOIT envoyer un message d'erreur Shim6 avec le code d'erreur 1 avec le champ Pointeur se référant au premier octet du type d'option.

12.4 Recherche de contexte

On suppose que chaque contexte Shim a son propre automate à états. On suppose qu'un répartiteur livre les paquets entrants à l'automate à états auquel ils appartiennent. Ici, on décrit les règles utilisée pour que le répartiteur livre les paquets à l'automate à états du contexte Shim correct.

Il y a un automate à états par contexte identifié qui est conceptuellement identifié par la paire d'ULID et l'identifiant d'instance fourchée (qui est zéro par défaut) ou identifié par CT(local). Cependant, les règles de recherche détaillées sont plus complexes, en particulier durant l'établissement de contexte.

En clair, si le contexte demandé n'est pas établi, il va être dans l'état IDLE.

Durant l'établissement de contexte, le contexte est identifié comme suit :

- Paquets I1 : livré au contexte associé avec la paire d'ULID et l'identifiant d'instance fourchée.
- Paquets I2 : livré au contexte associé avec la paire d'ULID et l'identifiant d'instance fourchée.
- Paquets R1 : livré au contexte avec la paire de localisateurs incluse dans le paquet et le nom occasionnel d'initiateur inclus dans le paquet (R1 ne contient pas de paire d'ULID ni de CT(local)). Si aucun contexte n'existe avec cette paire de localisateurs et nom occasionnel d'initiateur, éliminer en silence.
- Paquets R2 : livré au contexte avec la paire de localisateurs incluse dans le paquet et le nom occasionnel d'initiateur inclus dans le paquet (R2 ne contient pas de paire d'ULID ni de CT(local)). Si aucun contexte n'existe avec cette paire de localisateurs et ce nom occasionnel d'initiateur, éliminer en silence.
- Paquets R1bis : livré au contexte qui a la paire de localisateurs et CT(homologue) égaux à l'étiquette de contexte de paquet incluse dans le packet R1bis.
- Paquets I2bis : livré au contexte associé avec la paire d'ULID et l'identifiant d'instance fourchée.
- En-têtes Extension de charge utile Shim6 : livré au contexte avec CT(local) égal à l'étiquette de contexte de receveur incluse dans le paquet.
- Autres messages de contrôle (Update, Keepalive, Probe) : livré au contexte avec CT(local) égal à l'étiquette de contexte de receveur incluse dans le paquet. Vérifier que le champ Adresse IPv6 de source fait partie de Ls(homologue) et que le champ Adresse IPv6 de destination fait partie de Ls(local). Sinon, envoyer un message R1bis.
- Messages d'erreur Shim6 et erreurs ICMP qui contiennent un en-tête Extension de charge utile Shim6 ou autre paquet de contrôle Shim dans le "paquet erroné" : utilise le "paquet erroné" pour répartir comme suit : livrer au contexte avec CT(homologue) égal à l'étiquette de contexte du receveur -- Lp(local) étant l'adresse IPv6 de source et Lp(homologue) étant l'adresse IPv6 de destination.

De plus, le Shim sur le côté envoyeur doit être capable de trouver l'état du contexte quand un paquet d'ULP est passé depuis l'ULP. Dans ce cas, la clé de recherche est la paire d'ULID et FII=0. Si on a une API d'ULP qui permet à l'ULP de faire le fourchement de contexte, l'ULP va probablement passer l'identifiant d'instance fourchée.

13. Contact initial

Le contact initial est une communication non Shim entre deux ULID, comme décrit à la Section 2. À ce moment, il n'y a pas d'activité de Shim.

Que le Shim finisse ou non par être utilisé (par exemple, l'homologue pourrait ne pas prendre en charge Shim6) il est très souhaitable que le contact initial puisse être établi même si il y a une défaillance pour une ou plusieurs adresses IP.

L'approche retenue est de s'appuyer sur les applications et protocoles de transport pour réessayer avec des adresses différentes de source et de destination, cohérentes avec ce qui est déjà spécifié dans la [RFC3484] "Choix d'adresse par défaut pour IPv6" ainsi qu'avec des correctifs à cette spécification [9], pour lui faire essayer différentes adresses de source et pas seulement différentes adresses de destination.

La mise en œuvre d'une telle approche peut éventuellement résulter en de longues temporisations. Par exemple, si on considère une mise en œuvre simple à l'API de prise qui utilise getaddrinfo() pour restituer toutes les adresses de destination et ensuite essaye bind() et connect() pour essayer toutes les combinaisons d'adresse de source et destination et attend que TCP arrive en fin de temporisation pour chaque combinaison avant d'essayer la suivante.

Cependant, si les mises en œuvre encapsulent cela dans une nouvelle API connect-by-name() et utilisent des appels de connexion non bloquants, il est possible de passer toutes les combinaisons disponibles de manière plus rapide jusqu'à trouver une paire de source et destination qui fonctionne. Donc, les problèmes dans ce domaine sont des problèmes de mises en œuvre et de l'API de prise courante, et non des problèmes de spécification de protocole. Honnêtement, fournir une API connect-by-name() d'application facile pour TCP et autres transports en mode connexion est aisé, fournir une capacité similaire à l'API pour UDP est difficile parce que le protocole lui-même ne fournit aucun retour de "succès". Donc, même le problème de UDP est celui des API et de la mise en œuvre.

14. Constantes du protocole

Le protocole utilise les constantes suivantes :
I1_RETRIES_MAX = 4
I1_TIMEOUT = 4 s
NO_R1_HOLDDOWN_TIME = 1 min
ICMP_HOLDDOWN_TIME = 10 min
I2_TIMEOUT = 4 s
I2_RETRIES_MAX = 2
I2bis_TIMEOUT = 4 s
I2bis_RETRIES_MAX = 2
VALIDATOR_MIN_LIFETIME = 30 s
UPDATE_TIMEOUT = 4 s
MAX_UPDATE_TIMEOUT = 120 s

Les temporisateurs de retransmission (I1_TIMEOUT, I2_TIMEOUT, UPDATE_TIMEOUT) sont soumis à un retard binaire exponentiel ainsi qu'à un aléa dans la gamme de 0,5 et 1,5 fois la valeur de retard nominal. Cela supprime tout risque de synchronisation entre des quantités d'hôtes effectuant des opérations Shim indépendantes en même temps.

L'aléa est appliqué après le retard binaire exponentiel. Donc, la première retransmission va se produire sur la base d'un nombre aléatoire uniformément distribué dans la gamme de [0,5*4, 1,5*4] secondes ; la seconde retransmission, [0,5*8, 1,5*8] secondes après la première, etc.

15. Autres implications

15.1 Considérations de contrôle de l'encombrement

Quand la paire de localisateurs actuellement utilisée pour échanger des paquets dans un contexte Shim6 devient inaccessible, la couche Shim6 va détourner la communication à travers une paire de localisateurs de remplacement, qui dans la plupart des cas va résulter en la redirection du flux de paquets à travers un chemin de réseau de remplacement. Dans ce cas, il est recommandé que le Shim6 suive les recommandations définies dans [21] et informe les couches supérieures sur le changement de chemin, afin de permettre aux mécanismes de contrôle d'encombrment des couches supérieures de

réagir en conséquence.

15.2 Considérations de boîtiers de médiation

Les paquets de données qui appartiennent à un contexte Shim6 portant l'en-tête de charge utile Shim6 contiennent des localisateurs de remplacement autres que les ULID dans les champs d'adresse de source et de destination de l'en-tête IPv6. Par ailleurs, les couches supérieures des homologues impliqués dans la communication opèrent sur la paire d'ULID qui leur est présentée par la couche Shim6, plutôt que sur la paire de localisateurs contenue dans l'en-tête IPv6 des paquets actuels. On devrait noter que la couche Shim6 ne modifie pas les paquets de données mais, parce que une paire d'ULID constante est présentée aux couches supérieures sans égard aux changements de la paire de localisateurs, la relation entre l'en-tête de couche supérieure (telle que TCP, UDP, ICMP, ESP, etc) et l'en-tête IPv6 est modifiée. En particulier, quand l'en-tête Extension Shim6 est présent dans le paquet, si ces paquets de données sont des paquets TCP, UDP, ou ICMP, le pseudo en-tête utilisé pour le calcul de somme de contrôle va contenir la paire d'ULID, plutôt que la paire de localisateurs contenue dans le paquet de données.

Il est possible que des pare-feu ou autre boîtiers de médiation essayent de vérifier la validité des vérifications de bonne santé de couche supérieure du paquet sur le chemin. Si ils font cela sur la base des adresses actuelles de source et de destination contenues dans l'en-tête IPv6 sans considérer les informations de contexte Shim6 (en particulier, sans remplacer la paire de localisateurs par la paire d'ULID utilisée par le contexte Shim6) de telles vérifications peuvent échouer. Ces boîtiers de médiation doivent être mis à jour afin d'être capables d'analyser l'en-tête Shim6 de charge utile et de trouver le prochain en-tête. Il est recommandé que des pare-feu et autre boîtiers de médiation n'éliminent pas les paquets qui portent l'en-tête Shim6 de charge utile avec des vérifications de validité de couche supérieure apparemment incorrectes qui impliquent les adresses dans l'en-tête IPv6 pour leurs calculs, sauf si ils sont capables de déterminer la paire d'ULID du contexte Shim6 associé au paquet de données et d'utiliser la paire d'ULID pour la vérification de l'essai de validité.

Dans le cas particulier des sommes de contrôle TCP, UDP, et ICMP, il est recommandé que les pare-feu et autre boîtiers de médiation n'éliminent pas les paquets TCP, UDP, et ICMP qui portent d'en-tête Shim6 de charge utile avec des sommes de contrôle apparemment incorrectes quand elles utilisent les adresses dans l'en-tête IPv6 pour le calcul de pseudo en-tête, sauf si ils sont capables de déterminer la paire d'ULID du contexte Shim6 associé au paquet de données et d'utiliser la paire d'ULID pour déterminer la somme de contrôle qui doit être présente dans un paquet avec des adresses réécrites par Shim6.

De plus, des pare-feu qui aujourd'hui passent un trafic limité, par exemple, de connexions TCP sortantes, vont probablement bloquer le protocole Shim6. Cela signifie que même quand des hôtes à capacité Shim6 sont en communication, les messages I1 vont être éliminés ; donc, les hôtes ne vont pas découvrir que leur homologue est à capacité Shim6. C'est, en fait, un avantage car, si les hôtes ont projeté d'établir un contexte de paire d'ULID, le pare-feu va probablement éliminer les paquets "différents" qui sont envoyés après une défaillance (ceux qui utilisent l'en-tête Extension de charge utile Shim6 avec un paquet TCP à l'intérieur). Donc, les pare-feu à états pleins qui sont modifiés pour passer les messages Shim6 devraient aussi être modifiés pour passer l'en-tête Extension de charge utile Shim6 afin que le Shim puisse utiliser les localisateurs de remplacement pour récupérer des défaillances. Cela implique probablement que le pare-feu a besoin de tracer l'ensemble de localisateurs utilisé en cherchant les échanges de contrôle Shim6. De tels pare-feu pourraient même vouloir vérifier les localisateurs en utilisant la vérification de HBA/CGA elle-même, qu'ils peuvent faire sans modifier de paquet Shim6 à travers lequel ils passent.

15.3 Considérations de fonctionnement et de gestion

Ce paragraphe examine les aspects relatifs au fonctionnement et à la gestion du protocole Shim6.

Déploiement du protocole Shim6 : le protocole Shim6 est une solution fondée sur l'hôte. Donc, afin d'être déployé, la pile des hôtes qui utilisent le protocole Shim6 doit être mise à jour pour le prendre en charge. Cela permet un déploiement incrémentaire du protocole, car il n'exige pas un signal pour le déploiement -- juste une seule mise à jour de l'hôte. Si la solution Shim6 va être déployée sur un site, l'hôte peut être graduellement mis à jour pour prendre en charge la solution. De plus, pour prendre en charge le protocole Shim6, seuls les hôtes d'extrémité ont besoin d'être mis à jour et aucun changement de routeur n'est nécessaire. Cependant, on devrait noter que, afin de bénéficier du protocole Shim6, les deux extrémités d'une communication devraient prendre en charge le protocole, ce qui signifie que les deux hôtes doivent être mis à jour pour être capables d'utiliser le protocole Shim6. Néanmoins, le protocole Shim6 utilise une capacité différée d'établissement de contexte qui permet aux hôtes d'extrémité d'établir des communications IPv6 normales et, plus tard, si les deux points d'extrémité sont à capacité Shim6, d'établir le contexte Shim6 en utilisant le protocole Shim6. Ceci a un important avantage de déploiement, car les nœuds à capacité Shim6 peuvent parfaitement parler à des nœuds sans capacité Shim6 sans introduire de problème dans la communication.

Configuration de nœuds à capacité Shim6 : le protocole Shim6 lui-même n'exige aucune configuration spécifique pour fournir ses caractéristiques de base. Le protocole Shim6 est conçu pour fournir un service par défaut aux couches supérieures qui devrait satisfaire les applications générales. La couche Shim6 va automatiquement tenter de protéger les communications de longue durée en déclenchant l'établissement du contexte Shim6 en utilisant une heuristique prédéfinie. Bien sûr, si un réglage spécial est requis par certaines applications, cela peut exiger de la configuration supplémentaire. Des considérations similaires s'appliquent à un site qui tente d'effectuer des formes d'ingénierie du trafic en utilisant différentes préférences pour différents localisateurs.

Configuration d'adresse et de préfixe : le protocole Shim6 suppose que dans un site multi rattachements, plusieurs préfixes vont être disponibles. Une telle configuration peut augmenter le travail de fonctionnement d'un réseau. Cependant, on devrait noter que la capacité d'avoir plusieurs préfixes dans un site et plusieurs adresses allouées à une interface est une capacité IPv6 qui va au delà du cas de Shim6, et on s'attend à ce qu'elle soit largement utilisée. Donc, même si c'est le cas pour Shim6, on considère que les implications d'une telle configuration vont au delà du cas particulier de Shim6 et doivent être traitées pour le cas IPv6 générique. Néanmoins, Shim6 suppose aussi l'usage des adresses CGA/HBA par les hôtes Shim6. Cela implique que les hôtes à capacité Shim6 devraient configurer les adresses en utilisant les mécanismes de génération de HBA/CGA. Des considérations supplémentaires sur cette question se trouvent dans la [RFC6629].

15.4 Autres considérations

L'approche général de Shim6 ainsi que les spécificités de la solution proposée ont des implications autres, incluant :

- Les applications qui effectuent des références ou des rappels en utilisant des adresses IP comme des "identifiants" peuvent encore fonctionner de façon limitée, comme décrit dans [18]. Mais, afin que de telles applications soient capables de tirer parti de plusieurs localisateurs pour la redondance, les applications doivent être modifiées pour utiliser des noms de domaine pleinement qualifiés (FQDN) comme des "identifiants" ou elles doivent passer tous les localisateurs comme "identifiants", c'est-à-dire, un "identifiant" du point de vue de l'application devient un ensemble d'adresses IP au lieu d'une seule adresse IP.

- Les protocoles de signalisation pour la qualité de service ou pour autre chose qui implique d'avoir des appareils dans le chemin du réseau qui cherchent des adresses IP et des numéros d'accès (ou des adresses IP et des étiquettes de flux) doivent être invoqués sur les hôtes quand la paire de localisateurs change à cause d'une défaillance. À ce moment, ces protocoles doivent informer les appareils qu'une nouvelle paire d'adresses IP va être utilisée pour le flux. Noter que c'est le cas même dans ce protocole, à la différence de certaines propositions antérieures, qui ne surcharge pas l'étiquette de flux comme étiquette de contexte ; les appareils dans le chemin doivent savoir l'utilisation des nouveaux localisateurs même quand l'étiquette de flux reste la même.

- Implications de MTU. En calculant un minimum sur les MTU de chemin récemment observées, les mécanismes de MTU de chemin qu'on utilise sont robustes à l'égard de différents paquets prenant des chemins différents à travers l'Internet. Quand Shim6 échoue en utilisant une paire de localisateurs pour une autre, cela signifie que des paquets pourraient voyager sur un chemin différent à travers l'Internet ; donc, la MTU de chemin pourrait être assez différente. Afin de traiter ce changement de MTU, l'usage de la découverte de MTU de chemin de couche de mise en paquet, comme définie dans la [RFC4281] est recommandé.

Le fait que Shim va ajouter un en-tête Extension de charge utile Shim6 de 8 octets aux paquets ULP après une commutation de localisateur peut aussi affecter la MTU de chemin utilisable pour les ULP. Dans ce cas, le changement de MTU est local chez l'hôte envoyeur ; donc, porter le changement aux ULP est une affaire de mise en œuvre. En portant les informations à la couche de transport, il peut adapter et réduire la taille de segment maximum (MSS, *Maximum Segment Size*) en conséquence.

16. Considérations sur la sécurité

Le présent document satisfait les exigences spécifiées dans la [RFC4218] comme suit :

- Les techniques de HBA [RFC5535] et CGA [RFC3972] pour vérifier les localisateurs pour empêcher un attaquant de rediriger le flux de paquets sur une autre destination, préviennent les menaces décrites aux paragraphes 4.1.1, 4.1.2, 4.1.3, et 4.2 de la [RFC4218]. Ces deux techniques donnent un niveau de protection similaire mais aussi fournissent des fonctions différentes avec des coûts de calcul différents.

  Le mécanisme HBA s'appuie sur la capacité de générer toutes les adresses d'un hôte multi rattachement comme un

ensemble inaltérable d'adresses IPv6 liées intrinsèquement, appelée un ensemble de HBA. Dans cette approche, les adresses incorporent un hachage cryptographique unidirectionnel de l'ensemble de préfixes disponible dans la partie identifiant d'interface. Le résultat est que le lien entre toutes les adresses disponibles est codé dans les adresses elles-mêmes, fournissant une protection contre la capture. Tout homologue qui utilise le nœud de protocole Shim peut vérifier efficacement que les adresses de remplacement proposées pour continuer la communication sont liées à l'adresse initiale par un simple calcul de hachage.

Dans une approche fondée sur la CGA, l'adresse utilisée comme ULID est une CGA qui contient un hachage d'une clé publique dans son identifiant d'interface. Le résultat est un lien sûr entre l'ULID et la paire de clés associée. Cela permet à chaque homologue d'utiliser la clé privée correspondante pour signer les messages Shim qui portent des informations d'ensemble de localisateurs. La chaîne de confiance dans ce cas est la suivante : l'ULID utilisé pour la communication est lié de façon sûre à la paire de clés parce que il contient le hachage de la clé publique, et l'ensemble de localisateurs est lié à la clé publique par la signature.

Ces deux mécanismes, HBA et CGA, fournissent une protection contre l'attaque différée (décrite au paragraphe 4.1.2 de la [RFC4218]) car l'ULID est lié de façon sûre à un ensemble de localisateurs qui peut seulement être défini par le possesseur de l'ULID. La longueur minimum acceptable de clé pour les clés RSA utilisées dans la génération des CGA DOIT être d'au moins 1024 bits. Toute mise en œuvre devrait suivre une pratique cryptographique prudente pour déterminer les longueurs de clé appropriées.

- Les attaques d'inondation par des tiers, décrites au paragraphe 4.3 de la [RFC4218], sont empêchées en exigeant d'un homologue Shim6 qu'il effectue un échange de sondage + réponse réussi d'accessibilité avant d'accepter l'utilisation d'un nouveau localisateur comme destination de paquet.

- Le premier message ne crée aucun état sur le répondeur. Essentiellement, un échange en trois phases est exigé avant que le répondeur crée un état. Cela signifie que une attaque de déni de service fondée sur l'état (essayant d'utiliser toute la mémoire chez le répondeur) exige au moins que l'attaquant crée un état, consommant ses propres ressources ; cela donne aussi une adresse IPv6 que l'attaquant a utilisée.

- Les messages d'établissement de contexte utilisent les noms occasionnels pour empêcher les attaques en répétition, qui sont décrites au paragraphe 4.1.4 de la [RFC4218], et empêcher des attaquants hors chemin d'interférer avec l'établissement.

- Chaque message de contrôle du protocole Shim6, après l'établissement de contexte, porte l'étiquette de contexte allouée à un contexte particulier. Cela implique qu'un attaquant a besoin de découvrir cette étiquette de contexte avant d'être capable de simuler un message de contrôle Shim6 comme décrit au paragraphe 4.4 de la [RFC4218]. Une telle découverte exige probablement qu'un attaquant soit sur le chemin afin de renifler la valeur de l'étiquette de contexte. Le résultat est que, par cette technique, le protocole Shim6 est protégé contre les attaquants hors chemin.

16.1 Interaction avec IPsec

Shim6 a deux modes de traitement des paquets de données. Si la paire d'ULID est aussi la paire de localisateurs utilisée, alors le paquet de données n'est pas modifié par Shim6. Dans ce cas, l'interaction avec IPsec est exactement la même que si la couche Shim6 n'était pas présente chez l'hôte.

Si la paire d'ULID diffère de la paire de localisateurs courante pour ce contexte Shim6, alors Shim6 va prendre le paquet de données, remplacer les ULID contenus dans les champs d'adresse IP de source et de destination par la paire de localisateurs courante, et ajouter l'extension Shim6 avec l'étiquette de contexte correspondante. Dans ce cas, comme mentionné au paragraphe 1.6, Shim6 fonctionne comme un mécanisme de tunnel, où l'en-tête interne contient l'ULID et l'en-tête externe contient les localisateurs. La principale différence est que l'en-tête interne est "compressé" et une étiquette de compression, à savoir l'étiquette de contexte, est ajoutée pour décompresser l'en-tête interne chez l'extrémité receveuse.

Dans ce cas, l'interaction entre IPsec et Shim6 est alors similaire à l'interaction entre IPsec et un mécanisme de tunnel. Quand le paquet est généré par le protocole de couche supérieure, il est passé à la couche IP contenant les ULID dans le champ Source et Destination IP. IPsec est alors appliqué à ce paquet. Puis le paquet est passé à la sous couche Shim6, qui "encapsule" le paquet reçu et inclut un nouvel en-tête IP contenant la paire de localisateurs dans le champ Source et Destination IP. Ce nouveau paquet IP est à son tour passé à IPsec pour être traité, tout comme dans le cas d'un tunnel. Cela peut être vu comme si IPsec était situé à la fois au dessus et en-dessous de la sous couche Shim6 et comme si les politiques IPsec s'appliquaient à la fois aux ULID et aux localisateurs.

Quand IPsec a traité le paquet après la sous couche Shim6 (c'est-à-dire, le paquet portant les localisateurs dans le champ Source et Destination IP) la sous couche Shim6 peut avoir ajouté l'en-tête Extension Shim6. Dans ce cas, IPsec doit sauter l'en-tête Extension Shim6 pour trouver les sélecteurs pour les protocoles de la couche suivante (par exemple, TCP, UDP, SCTP).

Quand un paquet est reçu à l'autre extrémité, il est traité sur la base de l'ordre des en-têtes d'extension. Donc, si un en-tête ESP ou AH précède un en-tête Shim6, cela détermine l'ordre. Shim6 introduit le besoin de faire des vérifications de politique, analogues à celles qui sont faites pour les tunnels, quand Shim6 reçoit un paquet et que la paire d'ULID pour ce paquet n'est pas identique à la paire de localisateurs dans le paquet.

16.2 Menaces résiduelles

Certaines des menaces résiduelles de cette proposition sont :

- Un attaquant qui arrive tard sur le chemin (après l'établissement du contexte) peut utiliser le message R1bis pour causer la re-création du contexte par un homologue et, à ce moment, il peut observer tout l'échange. Mais cela ne semble pas ouvrir de nouvelles portes à l'attaquant dans la mesure où il peut observer les étiquettes de contexte qui sont utilisées et, une fois connues, il peut les utiliser pour envoyer des messages bogués.

- Un attaquant présent sur le chemin afin de trouver les étiquettes de contexte peut générer un message R1bis après avoir quitté le chemin. Pour que ce paquet soit efficace, il doit avoir un localisateur de source qui appartient au contexte ; donc, il ne peut pas y avoir "trop" de filtrage d'entrée entre la nouvelle localisation de l'attaquant et les homologues communicants. Mais cela ne semble pas être si sévère parce que, une fois que le R1bis cause le rétablissement du contexte, une nouvelle paire d'étiquettes de contexte va être utilisée, qui ne va pas être connue de l'attaquant. Si cela pose toujours un problème, on pourrait exiger une prise de contact en deux phases, "avez vous réellement perdu l'état ?", en réponse au message d'erreur.

- Il serait possible à un attaquant d'essayer des étiquettes de contexte aléatoires de 47 bits et de voir si elles peuvent causer la perturbation de la communication entre deux hôtes. En particulier, dans le cas de paquets de charge utile, les effets d'une telle attaque seraient similaires à ceux d'un attaquant qui envoie des paquets avec une adresse de source usurpée. Dans le cas de paquets de contrôle, il n'est pas suffisant de trouver l'étiquette de contexte correcte -- des informations supplémentaires sont requises (par exemple, les noms occasionnels, les adresses de source appropriées ; voir le point précédent pour le cas de R1bis). Si une étiquette de 47 bits, qui est la plus grande qui tient dans un en-tête Extension de 8 octets, n'est pas suffisante, on pourrait utiliser une étiquette encore plus grande dans les messages de contrôle Shim6 et utiliser les 47 bits de moindre poids dans l'en-tête Extension de charge utile Shim6.

- Quand l'en-tête Extension de charge utile Shim6 est utilisé, un attaquant qui peut deviner l'étiquette de contexte aléatoire de 47 bits peut injecter des paquets dans le contexte avec tout localisateur de source. Donc, si il y a un filtrage d'entrée entre l'attaquant et sa cible, cela pourrait éventuellement permettre à l'attaquant d'outrepasser le filtrage d'entrée. Cependant, en plus de deviner l'étiquette de contexte de 47 bits, l'attaquant doit aussi trouver un contexte où, après le remplacement par le receveur des localisateurs par les ULID, la somme de contrôle d'ULP est correcte. Mais même cela ne serait pas suffisant avec des ULP comme TCP, car les numéros d'accès et les numéros de séquence TCP doivent correspondre à une connexion existante. Donc, même si les problèmes d'attaquant hors du chemin qui injectent des paquets sont différents du filtrage d'entré d'aujourd'hui, il est quand même très difficile à un attaquant hors du chemin de les deviner. Si IPsec est appliqué, le problème serait alors complètement éliminé.

- Le valideur inclus dans les paquets R1 et R1bis est généré comme un hachage de plusieurs paramètres d'entrée. Alors que la plupart des entrées sont en fait déterminées par l'envoyeur, et que seule la valeur secrète S est inconnue de l'envoyeur, la protection résultante est réputée être suffisante car il serait plus facile pour l'attaquant d'obtenir juste un nouveau valideur en envoyant un paquet I1 que d'effectuer tous les calculs requis pour déterminer le secret S. Néanmoins, il est recommandé que l'hôte change périodiquement le secret S.

17. Considérations relatives à l'IANA

L'IANA a alloué une nouvelle valeur de numéro de protocole IP (140) pour le protocole Shim6.

L'IANA a enregistré un type de message CGA pour le protocole Shim6 dans le registre des Types d'extension de CGA avec la valeur 0x4A30 5662 4858 574B 3655 416F 506A 6D48.

L'IANA a établi un registre Paramètres Shim6 avec quatre composants : enregistrements de type Shim6, enregistrements d'options Shim6, enregistrements de code d'erreur Shim6, et enregistrements de méthode de vérification Shim6.

Le contenu initial du registre de type Shim6 est le suivant :

Valeur de type	Message
0	Réservé
1	I1 (premier message d'établissement de l'initiateur)
2	R1 (premier message d'établissement du répondeur)
3	I2 (second message d'établissement de l'initiateur)
4	R2 (second message d'établissement du répondeur)
5	R1bis (réponse pour référence à un contexte non existant)
6	I2bis (réponse à un message R1bis)
7-59	Alloué en utilisant une action de normalisation
60-63	Pour utilisation expérimentale
64	Demande de mise à jour
65	Accusé de réception de mise à jour
66	Maintien en vie *(Keepalive)*
67	Message de sonde
68	Message d'erreur
69-123	Alloué en utilisant une action de normalisation
124-127	Pour utilisation expérimentale

Le contenu initial du registre des options Shim6 est le suivant :

Type	Nom d'option
0	Réservé
1	Valideur de répondeur
2	Liste de localisateurs
3	Préférences de localisateurs
4	Structure de données de paramètres de CGA
5	Signature de CGA
6	Paire d'ULID
7	Identifiant d'instance fourchée
8-9	Alloué en utilisant une action de normalisation
10	Option Fin de temporisation de maintien en vie
11-16383	Alloué en utilisant une action de normalisation
16384-32767	Pour utilisation expérimentale

Le contenu initial du registre des codes d'erreur Shim6 est le suivant :

Valeur de code	Description
0	Type de message Shim6 inconnu
1	Option critique non reconnue
2	Échec de la méthode de vérification de localisateur
3	Génération de liste de localisateurs : nombre hors synchronisation
4	Erreur du nombre de localisateurs
5-19	Alloué en utilisant une action de normalisation
120-127	Réservé aux fins de débogage

Le contenu initial du registre des méthodes de vérification Shim6 est le suivant :

Valeur	Méthode de vérification
0	Réservé
1	CGA
2	HBA
3-200	Alloué en utilisant une action de normalisation
201-254	Pour utilisation expérimentale
255	Réservé

18. Remerciements

Au fil des ans, de nombreuses peronnes actives dans les groupes de travail multi6 et shim6 ont contribué par des idées et des suggestions qui sont reflétées dans cette spécification. Des remerciements particuliers pour les commentaires précis de Sam Hartman, Cullen Jennings, Magnus Nystrom, Stephen Kent, Geoff Huston, Shinta Sugimoto, Pekka Savola, Dave Meyer, Deguang Le, Jari Arkko, Iljitsch van Beijnum, Jim Bound, Brian Carpenter, Sebastien Barre, Matthijs Mekking, Dave Thaler, Bob Braden, Wesley Eddy, Pasi Eronen, et Tom Henderson sur les versions antérieures de ce document.

19. Références

19.1 Références normatives

[RFC2119] S. Bradner, "Mots clés à utiliser dans les RFC pour indiquer les niveaux d'exigence", BCP 14, mars 1997. *(MàJ par* RFC8174)

[RFC3972] T. Aura, "Adresses générées cryptographiquement (CGA)", mars 2005. (*MàJ par* RFC4581, RFC4982) *(P.S.)*

[RFC5534] J. Arkko, I. van Beijnum, "Protocole d'exploration de paire de détection et localisateur de défaillance pour multi-rattachement IPv6" juin 2009. *(P. S.)*

[RFC5535] M. Bagnulo, "Adresses fondées sur le hachage (HBA)", juin 2009. *(P. S.)*

19.2 Références pour information

[RFC2782] A. Gulbrandsen, P. Vixie et L. Esibov, "Enregistrement de ressource DNS pour la spécification de la localisation des services (DNS SRV)", février 2000, DOI 10.17487/RFC2782.

[RFC2827] P. Ferguson, D. Senie, "Filtrage d'entrée de réseau : Combattre les attaques de déni de service qui utilisent l'usurpation d'adresse de source IP", mai 2000. (*MàJ par* RFC3704) (BCP0038)

[RFC3484] R. Draves, "Choix d'adresse par défaut pour le protocole Internet version 6 (IPv6)", février 2003. (*Remplacée par la* RFC6724) *(P.S.)*

[RFC3550] H. Schulzrinne, S. Casner, R. Frederick et V. Jacobson, "RTP : un protocole de transport pour les applications en temps réel", STD 64, DOI 10.17487/RFC3550, juillet 2003. *(MàJ par* RFC7164, RFC7160, RFC8083, RFC8108, RFC8860)

[RFC3582] J. Abley, B. Black, V. Gill, "Objectifs des architectures IPv6 multi-sites/multi-rattachements", août 2003. *(Information)*

[RFC3697] J. Rajahalme et autres, "Spécification d'étiquette de flux IPv6", mars 2004. (*Obsolète, voir* RFC6437) *(P.S.)*

[RFC4086] D. Eastlake 3rd, J. Schiller, S. Crocker, "Exigences d'aléa pour la sécurité", juin 2005, DOI 10.17487/RFC4086, (*Remplace* RFC1750) (BCP0106)

[RFC4193] R. Hinden, B. Haberman, "Adresses IPv6 en envoi individuel uniques localement", octobre 2005. *(P.S.)*

[RFC4218] E. Nordmark, T. Li, "Menaces qui pèsent sur les solutions de rattachements multiples IPv6", octobre 2005. *(Information)*

[RFC4821] M. Mathis, J. Heffner, "Découverte de la MTU de chemin de couche de mise en paquet", DOI 10.17487/RFC4821, mars 2007. *(P.S.)*

[RFC4884] R. Bonica et autres, "ICMP étendu pour la prise en charge de messages multiparties", avril 2007. (*MàJ* RFC0792, RFC4443) *(P.S. ; MàJ par* RFC8335)

[RFC5201] R. Moskowitz et autres, "Protocole d'identité d'hôte (HIP)", avril 2008. *(Expérimentale)*

[RFC5386] N. Williams, M. Richardson, "La sécurité mieux que rien : un mode non authentifié de IPsec", novembre 2008. *(P.S.)*

[RFC6316] M. Komu, M. Bagnulo, K. Slavov, S. Sugimoto, "Interface de programme d'application (API) de prise pour Shim multi rattachements", juillet 2011. *(Information)*

[RFC6629] J. Abley, M. Bagnulo, A. Garcia-Martinez, "Considérations sur l'application du protocole Shim de multi rattachements de niveau 3 pour IPv6 (Shim6)", juin 2012. *(Information)*

[8] Nordmark, E., "Multihoming without IP Identifiers", Travail en cours, juillet 2004.

[9] Bagnulo, M., "Updating RFC 3484 for multihoming support", Travail en cours, novembre 2007.

[16] Huitema, C., "Ingress filtering compatibility for IPv6 multihomed sites", Travail en cours, septembre 2005.

[17] Bagnulo, M. et E. Nordmark, "SHIM - MIPv6 Interaction", Travail en cours, juillet 2005.

[18] Nordmark, E., "Shim6-Application Referral Issues", Travail en cours, juillet 2005.

[21] Schuetz, S., Koutsianas, N., Eggert, L., Eddy, W., Swami, Y., et K. Le, "TCP Response to Lower-Layer Connectivity-Change Indications", Travail en cours, février 2008.

Appendice A. Extensions de protocole possibles

Durant le développement de ce protocole, plusieurs problèmes importants à traiter ont été soulevés mais ils n'ont pas besoin de figurer dans le protocole de base ; ils peuvent être des extensions au protocole. Les principaux sont :

- Comme déclaré dans les hypothèses à la Section 3, afin que le protocole Shim6 soit capable de récupérer d'une large gamme de défaillances (par exemple, quand un des hôtes communicants est à un seul rattachement) et pour traiter le cas des FAI d'un site qui font le filtrage d'entrée sur la base de l'adresse IPv6 de source, il est nécessaire que l'hôte soit capable d'influencer le choix de sortie de son site. Ce problème est discuté dans [16].

- Faut il garder secrète la liste des localisateurs entre les deux points d'extrémité communicants ? On peut éventuellement réaliser cela en utilisant CGA (pas quand on utilise HBA) mais seulement au prix des opérations de chiffrement et déchiffrement de clé publique au titre de l'établissement de contexte. La suggestion est de laisser cela pour une future extension du protocole.

- Définir une forme de mécanisme de "compression" de bout en bout qui supprime le besoin d'inclure l'en-tête Extension de charge utile Shim6 quand la paire de localisateurs n'est pas la paire d'ULID.

- Prendre en charge l'établissement dynamique des préférences de localisateur au niveau d'un site et utiliser l'option Préférence de localisateurs dans le protocole Shim6 pour porter ces préférences aux hôtes communicants distants. Cela pourrait refléter la notion de priorité et de pondération des enregistrements SRV du DNS.

- Spécifier des API afin que les ULP soient informés des localisateurs que Shim utilise et soient capables d'influencer le choix des localisateurs (contrôler les préférences et déclencher la commutation d'une paire de localisateurs). Cela inclut de fournir des API que les ULP peuvent utiliser pour fourcher un contexte Shim.

- Déterminer si il est faisable de relâcher les suggestions pour quand l'état de contexte est supprimé afin qu'on puisse finir avec une distribution asymétrique de l'état du contexte et avoir quand même (la plupart) des avantages de Shim. Par exemple, le serveur occupé va passer à travers l'établissement de contexte mais va rapidement supprimer l'état du contexte après cela (afin de garder de la mémoire) ; cependant, le client pas trop encombré va conserver l'état du contexte. Le mécanisme de récupération de contexte présenté au paragraphe 7.5 va alors re-créer l'état si le client devait envoyer un message de contrôle Shim (par exemple, un message de sonde parce que il voit un problème) ou un paquet d'ULP dans un en-tête Extension de charge utile Shim6 (parce que il a précédemment échoué sur une paire de localisateurs de remplacement mais a été silencieux pendant un moment). Cela semble donner l'avantage au Shim tant que le client peut détecter la défaillance. Si le client n'envoie rien et si c'est le serveur qui essaye d'envoyer, alors il ne va pas être capable de récupérer parce que le Shim sur le serveur n'a pas d'état de contexte et donc ne sait rien sur la

paire de localisateur de remplacement.

- Étudier si il se pourrait que le protocole de contrôle Shim6 ne s'appuie pas du tout sur un localisateur de source stable dans les paquets. Cela peut probablement être réalisé en ayant les messages de contrôle Shim qui incluent l'option de paire d'ULID.

- Si chaque hôte pourrait avoir un grand nombre de localisateurs, alors l'exigence de les inclure essentiellement tous dans les messages I2 et R2 pourrait être contraignante. Si c'est le cas, on peut chercher à utiliser la structure de données de paramètres de CGA pour la comparaison, au lieu des ensembles de préfixes, pour être capable de détecter une confusion de contexte. Cela ferait peser par exemple des contraintes sur l'utilisation seulement d'une clé publique (logique) de CGA ; cela exigerait aussi des règles soigneusement adaptées sur la façon dont deux PDS sont comparés comme "étant le même hôte". Mais si on n'attend pas plus d'une poignée de localisateurs par hôte, on n'a pas besoin d'ajouter cette complexité.

- Les temporisateurs spécifiés par l'ULP pour le mécanisme de détection d'accessibilité (qui peut être particulièrement utile quand il y a des contextes fourchés).

- Pré vérifier des paires de localisateurs de "secours", afin de raccourcir le délai de reprise.

- Étudier comment Shim6 et IPv6 Mobile pourraient interagir [17].

Appendice B. Automate à états simplifié7

Les états sont définis au paragraphe 6.2. L'intention est que la description stylisée ci-dessous soit cohérente avec la description textuelle de la spécification ; cependant, en cas de conflit, la description textuelle est normative.

Le tableau suivant décrit les actions possibles dans l'état IDLE et leurs déclencheurs respectifs :

Déclencheur	**Action**
Reçoit I1	Envoie R1 et reste à IDLE
L'heuristique déclenche un nouvel établissement de contexte	Envoie I1 et passe à I1-SENT
Reçoit I2, vérifie le valideur et le nom occ. de RÉP	Si réussi, envoie R2 et passe à ESTABLISHED Si échec, reste à IDLE
Reçoit I2bis, vérifie le valideur et le nom occ. de RÉP	Si réussi, envoie R2 et passe à ESTABLISHED Si échec, reste à IDLE
R1, R1bis, R2	N/A (ce contexte n'a pas les info requises pour que le répartiteur les livre)
Reçoit l'en-tête Extension de charge utile ou autre paquet de contrôle	Envoie R1bis et reste à IDLE

Le tableau suivant décrit les actions possibles dans l'état I1-SENT et leurs déclencheurs respectifs :

Déclencheur	**Action**
Reçoit R1, vérifie nom occasionnel INIT	Si réussi, envoie I2 et passe à I2-SENT Si échec, élimine et reste à I1-SENT
Reçoit I1	Envoie R2 et reste à I1-SENT
Reçoit R2, vérifie nom occasionnel INIT	Si réussi, passe à ESTABLISHED Si échec, élimine et reste à I1-SENT
Reçoit I2, vérifie valideur et nom occasionnel RESP	Si réussi, envoie R2 et passe à ESTABLISHED Si échec, élimine et reste à I1-SENT
Reçoit 2bis, vérifie valideur et nom occasionnel RESP	Si réussi, envoie R2 et passe à ESTABLISHED Si échec, élimine et reste à I1-SENT
Fin de temporisation, incrémente le compteur de tempo.	Si compteur =< I1_RETRIES_MAX, envoie I1 et reste à I1-SENT Si compteur > I1_RETRIES_MAX, passe à E-FAILED
Reçoit erreur charge utile ICMP inconnue	Passe à E-FAILED
R1bis	N/A (le répartiteur ne livre pas car CT(homologue) n'est pas établi
Reçoit en-tête Extension de charge utile ou autre paquet de contrôle	Élimine et reste à I1-SENT

Le tableau suivant décrit les actions possibles dans l'état I2-SENT et leurs déclencheurs respectifs :

Déclencheur	**Action**
Reçoit R2, vérifie nom occasionnel INIT	Si réussi, passe à ESTABLISHED Si échec, reste à I2-SENT
Reçoit I1	Envoie R2 et reste à I2-SENT
Reçoit I2, vérifie valideur et nom occasionnel RESP	Envoie R2 et reste à I2-SENT
Reçoit I2bis, vérifie valideur et nom occasionnel RESP	Envoie R2 et reste à I2-SENT
Reçoit R1	Élimine et reste à I2-SENT
Fin de temporisation, incrémente le compteur de tempo.	Si compteur =< I2_RETRIES_MAX, envoie I2 et reste à I2-SENT Si compteur > I2_RETRIES_MAX, envoie I1 et passe à I1-SENT
R1bis	N/A (le répartiteur ne livre pas car CT(homologue) n'est pas établi
Reçoit en-tête Extension de charge utile ou autre paquet de contrôle	Accepte et envoie I2 (R2 a probablement été envoyé par l'homologue et perdu)

Le tableau suivant décrit les actions possibles dans l'état I2BIS-SENT et leurs déclencheurs respectifs :

Déclencheur	**Action**
Reçoit R2, vérifie nom occasionnel INIT	Si réussi, passe à ESTABLISHED Si échec, reste à I2BIS-SENT
Reçoit I1	Envoie R2 et reste à I2BIS-SENT
Reçoit I2, vérifie valideur et nom occasionnel RESP	Envoie R2 et reste à I2BIS-SENT
Reçoit I2bis, vérifie valideur et nom occasionnel RESP	Envoie R2 et reste à I2BIS-SENT
Reçoit R1	Élimine et reste à I2BIS-SENT
Fin de temporisation, incrémente le compteur de tempo.	Si compteur =< I2_RETRIES_MAX, envoie I2bis et reste à I2BIS-SENT Si compteur > I2_RETRIES_MAX, envoie I1 et passe à I1-SENT
R1bis	N/A (le répartiteur ne livre pas car CT(homologue) n'est pas établi
Reçoit en-tête Extension de charge utile ou autre paquet de contrôle	Accepte et envoie I2bis (R2 a probablement été envoyé par l'homologue et perdu)

Le tableau suivant décrit les actions possibles dans l'état ESTABLISHED et leurs déclencheurs respectifs :

Déclencheur	**Action**
Reçoit I1, compare CT(homologue) avec CT reçu	Si pas de correspondance, envoie R1 et reste à ESTABLISHED Si correspondance, envoie R2 et reste à ESTABLISHED
Reçoit I2, vérifie valideur et nom occasionnel RESP	Si réussi, envoie R2 et reste à ESTABLISHED Autrement, élimine et reste à ESTABLISHED
Reçoit I2bis, vérifie valideur et nom occasionnel RESP	Si réussi, envoie R2 et reste à ESTABLISHED Autrement, élimine et reste à ESTABLISHED
Reçoit R2	Élimine et reste à ESTABLISHED
Reçoit R1	Élimine et reste à ESTABLISHED
Reçoit R1bis	Envoie I2bis et passe à I2BIS-SENT
Reçoit en-tête Extension de charge utile ou autre paquet de contrôle	Traite et reste à ESTABLISHED
Heuristique spécifique de la mise en œuvre (par exemple, pas de prise ULP ouverte et au repos pour un temps)	Élimine l'état et passe à IDLE

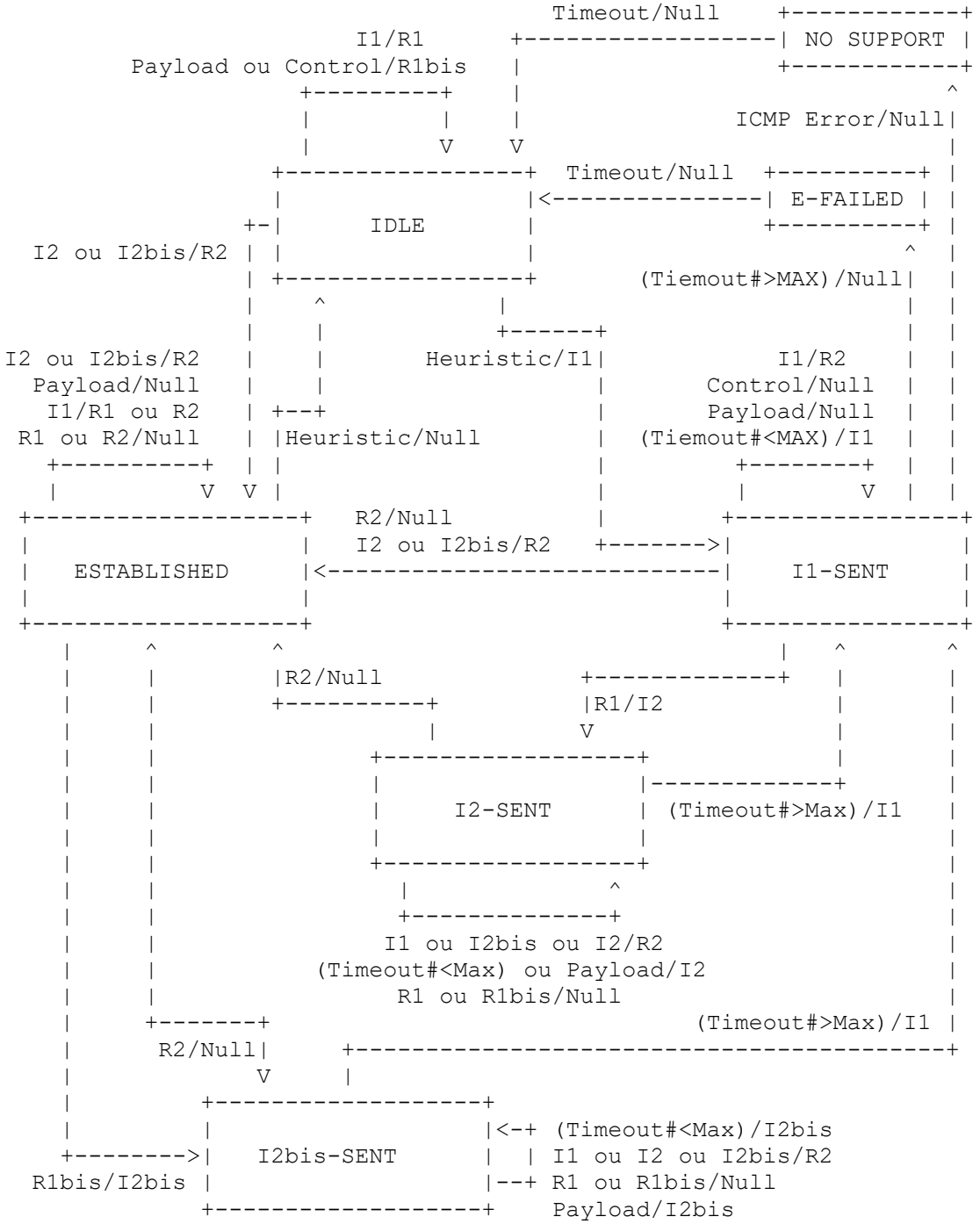
Le tableau suivant décrit les actions possibles dans l'état E-FAILED et leurs déclencheurs respectifs :

Déclencheur	**Action**
Attend pendant NO_R1_HOLDDOWN_TIME	Passe à IDLE
Tout paquet	Traité comme dans IDLE

Le tableau suivant décrit les actions possibles dans l'état NO-SUPPORT et leurs déclencheurs respectifs :

Déclencheur	**Action**
Attend pendant ICMP_HOLDDOWN_TIME	Passe à IDLE
Tout paquet	Traité comme dans IDLE

B.1 Diagramme d'automate à états simplifié



Appendice C. Réutilisation d'étiquette de contexte

Le protocole Shim6 n'a pas de mécanisme pour coordonner la suppression d'état entre les homologues parce que une telle suppression d'état ne semble pas utile, étant donné qu'un hôte peut être défaillant et réamorcer à tout moment. Il en résulte que le protocole a besoin d'être robuste contre la réutilisation d'une étiquette de contexte pour un autre contexte. Cette section résume les différents cas dans lesquels une étiquette peut être réutilisée, et comment fonctionne la récupération.

Un exemple des différents cas est donné ci-après. Supposons que les hôtes A et B communiquent en utilisant un contexte avec la paire d'ULID <A1, B2>, et que B ait alloué l'étiquette de contexte X à ce contexte. On suppose que B utilise seulement l'étiquette de contexte pour démultiplexer les en-têtes Extension de charge utile Shim6 reçus, car c'est le cas le plus général. De plus, on suppose que B supprime cet état de contexte, alors que A le conserve. B pourrait alors

ultérieurement allouer CT(local)=X à un autre contexte, et à ce moment on a plusieurs cas possibles :

- L'étiquette de contexte est réallouée à un contexte pour la même paire d'ULID <A1, B2>. On appelle cela "récupération de contexte" dans ce document.

- L'étiquette de contexte est réallouée à un contexte pour une paire d'ULID différente entre les deux mêmes hôtes, par exemple, <A3, B3>. On appelle cela "confusion de contexte" dans ce document.

- L'étiquette de contexte est réallouée à un contexte entre B et un autre hôte C, par exemple, pour la paire d'ULID <C3, B2>. C'est-à-dire une forme de confusion de contexte à trois.

C.1 Récupération de contexte

Ce cas est relativement simple et est discuté au paragraphe 7.5. L'observation est que comme la paire d'ULID est la même, quand A ou B essaye d'établir le nouveau contexte, A peut garder le vieux contexte alors que B recrée le contexte avec la même étiquette de contexte CT(B) = X.

C.2 Confusion de contexte

Ce cas est un peu plus complexe et est discuté au paragraphe 7.6. Quand le nouveau contexte est créé, si A ou B l'initie, l'hôte A peut détecter quand il reçoit l'ensemble de localisateurs de B (dans le message I2 ou R2) en ce qu'il finit avec deux contextes pour le même hôte homologue (ensembles de localisateurs Ls(homologue) en chevauchement) qui ont la même étiquette de contexte: CT(homologue) = X. À ce moment, l'hôte A peut supprimer toute possibilité de causer de la confusion en n'utilisant pas le vieux contexte pour envoyer d'autres paquets. Il élimine juste le vieux contexte (il ne pourrait pas être utilisé par du trafic d'ULP, car B l'a éliminé) ou il re-crée un contexte différent pour la vieille paire d'ULID (<A1, B2>) pour laquelle B va allouer un unique CT(B) au titre du mécanisme normal d'établissement de contexte.

C.3 Confusion de contexte à trois

Le troisième cas n'a pas sa place lorsque l'ancien état sur A peut être vérifié car le nouveau contexte est établi entre B et C. Donc, quand B reçoit les en-têtes Extension de charge utile Shim6 avec X comme étiquette de contexte, il va trouver le contexte pour <C3, B2> et, donc, va réécrire les paquets pour avoir C3 dans le champ Adresse de source et B2 dans le champ Adresse de destination avant de les passer à l'ULP. Cette réécriture est correcte quand les paquets sont en fait envoyés par l'hôte C, mais si l'hôte A se trouve envoyer un paquet en utilisant le vieux contexte, alors l'ULP sur A envoie un paquet avec les ULID <A1, B2> et le paquet arrive à l'ULP sur B avec les ULID <C3, B2>.

Ceci est clairement une erreur, et le paquet va très probablement être rejeté par l'ULP sur B du fait d'une mauvaise somme de contrôle de pseudo en-tête. Même si la somme de contrôle est juste (probabilité de 2^-16) l'ULP ne va probablement pas avoir de connexion pour ces ULID et numéros d'accès. Et si l'ULP est sans connexion, le traitement du paquet est très probablement sans danger : un tel ULP doit être capable de prendre en charge des paquets aléatoires envoyés par des homologues aléatoires dans tous les cas.

Cet état rompu, où les paquets sont envoyés de A à B en utilisant le vieux contexte sur l'hôte A, pourrait persister pendant quelques temps, mais ne va pas durer très longtemps. La détection de l'inaccessibilité sur l'hôte A va se faire parce que il ne voit aucun trafic de retour (messages de charge utile ou de maintien en vie) pour le contexte. Il va en résulter que l'hôte A envoie des messages de sonde à l'hôte B pour trouver une paire de localisateurs fonctionnelle. Cela a pour effet que l'hôte B va remarquer qu'il n'a pas de contexte pour la paire d'ULID <A1, B2> et CT(B) = X, ce qui fait que l'hôte B envoie un paquet R1bis pour rétablir ce contexte. Le contexte rétabli, tout comme dans le paragraphe précédent, va obtenir un CT(B) unique alloué par l'hôte B ; donc, il n'y aura plus de confusion.

C.4 Résumé

En résumé, il y a des cas où une étiquette de contexte pourrait être réutilisée alors qu'un homologue conserve l'état, mais le protocole peut s'en relever. La probabilité ce cet événement est faible, étant données la taille de 47 bits de l'étiquette de contexte. Cependant, il est important que ces mécanismes de récupération soient vérifiés. Donc, durant le développement et l'essai, il est recommandé que les mises en œuvre n'utilisent pas l'espace de 47 bits complet mais gardent plutôt, par exemple, le 40 premiers bits à zéro, laissant seulement l'hôte avec 128 étiquettes de contexte uniques. Cela aidera l'essai des mécanismes de récupération.

Appendice D. Solutions de remplacement de conception

Le présent document a effectué un certain ensemble de choix de conception afin de d'essayer de creuser un certain nombre de détails et de stimuler la discussion. Mais, comme cela a été discuté sur la liste de diffusion, d'autres choix auraient pu avoir du sens. Cet appendice énumère certaines solutions de remplacement.

D.1 Granularité du contexte

Au fil des ans, diverses suggestions ont été faite sur ce dont Shim devrait connaître, même si il opère à la couche IP, sur les connexions et sessions d'ULP et, par suite, être capable de faire des contextes Shim séparés pour des connexions et sessions d'ULP. Quelques options différentes ont été discutées :

- o Chaque connexion d'ULP se transpose en son propre contexte Shim.
- o Shim ignore la notion ULP de connexions et opère juste à une granularité d'hôte à hôte (adresse IP).
- o Les hybrides dans lesquels Shim est conscient de certains ULP (comme TCP) et traite les autres ULP hôte par hôte.

Avoir l'état Shim pour chaque connexion d'ULP signifie potentiellement de plus grands frais généraux car les frais généraux d'établissement d'état pourraient devenir significatifs ; il est utile d'être capable d'amortir cela sur plusieurs connexions.

Mais ignorer complètement les connexions d'ULP pourrait handicaper les ULP qui veulent que des communications différentes utilisent des paires de localisateurs différentes, par exemple, pour des raisons de qualité ou de coût.

Le protocole a un ajustement qui opère avec une granularité au niveau de l'hôte (strictement parlant, une granularité de paire d'ULID) pour être capable d'amortir l'établissement de contexte sur plusieurs connexions d'ULP. Ceci est combiné avec la capacité des ULP à capacité Shim6 de demander un fourchement de contexte afin que du trafic d'ULP différents puisse utiliser des paires de localisateurs différentes.

D.2 Démultiplexage des paquets de données dans les communications Shim6

Une fois qu'un contexte de paire d'ULID est établi entre deux hôtes, les paquets peuvent porter des localisateurs qui diffèrent des ULID présentés aux ULP en utilisant le contexte établi. Une des fonctions principales de la couche Shim6 est d'effectuer la transposition entre les localisateurs utilisés pour transmettre les paquets à travers le réseau et les ULID présentés à l'ULP. Afin d'effectuer cette traduction pour les paquets entrants, la couche Shim6 a besoin d'identifier d'abord quels paquets entrants doivent être traduits et ensuite d'effectuer la transposition entre localisateurs et ULID en utilisant le contexte associé. Cette opération est appelée "démultiplexage". On devrait noter que, parce que toute adresse peut être utilisée comme localisateur et comme ULID, des informations supplémentaires, autres que les adresses portées dans les paquets, doivent être prises en compte pour cette opération.

Par exemple, si un hôte a les adresses A1 et A2 et commence à communiquer avec un homologue qui a les adresses B1 et B2, alors une communication (connexion) pourrait utiliser la paire <A1, B1> as ULID et d'autres pourraient utiliser, par exemple, <A2, B2>. Initialement, il n'y a pas de défaillance, donc ces paires d'adresses sont utilisées comme localisateurs, c'est-à-dire, dans le champ Adresse IP des paquets sur le réseau. Mais quand il y a une défaillance, la couche Shim6 sur A pourrait décider d'envoyer les paquets qui utilisent <A1, B1> comme ULID en utilisant <A2, B2> comme localisateurs. Dans ce cas, B doit être capable de réécrire le champ Adresse IP pour certains paquets et pas pour d'autres, mais les paquets ont tous la même paire de localisateurs.

Afin de réaliser avec succès l'opération de démultiplexage, les paquets de données portent l'étiquette de contexte qui permet au receveur du paquet de déterminer le contexte de Shim à utiliser pour effectuer l'opération.

Deux mécanismes pour porter les informations d'étiquette de contexte ont été considérées en profondeur durant la conception du protocole Shim : celles portant l'étiquette de contexte dans le champ Étiquette de flux de l'en-tête IPv6 et celles utilisant un nouvel en-tête Extension pour porter l'étiquette de contexte. Dans cet appendice, on décrit les avantages et inconvénients de chaque mécanisme et on justifie l'option choisie.

D.2.1 Étiquette de flux

Une approche possible est de porter l'étiquette de contexte dans le champ Étiquette de flux de l'en-tête IPv6. Cela signifie que quand un contexte Shim6 est établi, une valeur d'étiquette de flux est associée à ce contexte (et peut-être une étiquette de flux séparée pour chaque direction).

La façon la plus simple de faire cela est d'avoir le triplet <étiquette de flux, localisateur de source, localisateur de destination> qui identifie le contexte chez le receveur.

Le problème avec cette approche est que, parce que les ensembles de localisateurs sont dynamiques, il n'est pas possible à un moment donné d'être sûr que deux contextes pour lesquels la même étiquette de contexte est allouée vont avoir des ensembles disjoints de localisateurs durant la vie des contextes.

Supposons que le Nœud A ait les adresses IPA1, IPA2, IPA3, et IPA4 et que l'Hôte B ait les adresses IPB1 et IPB2.
Supposons que deux différents contextes soient établis entre Hôte A et Hôte B.
Le contexte n° 1 utilise IPA1 et IPB1 comme ULID. L'ensemble de localisateurs associé à IPA1 est IPA1 et IPA2, tandis que l'ensemble de localisateurs associé à IPB1 est juste IPB1.
Le contexte n° 2 utilise IPA3 et IPB2 comme ULID. L'ensemble de localisateurs associé à IPA3 est IPA3 et IPA4, tandis que l'ensemble de localisateurs associé à IPB2 est juste IPB2.

Parce que les ensembles de localisateurs du contexte n° 1 et du contexte n° 2 sont disjoints, les hôtes pourraient penser que la même valeur d'étiquette de contexte peut être allouée aux deux. Le problème survient quand, plus tard, IPA3 est ajouté comme localisateur valide pour IPA1 dans le contexte n° 2 et que IPB2 est ajouté comme localisateur valide pour IPB1 dans le contexte n° 1. Dans ce cas, le triplet <étiquette de flux, localisateur de source, localisateur de destination> n'identifierait plus un contexte unique, et un démultiplexage correct ne serait plus possible.

Une approche possible pour surmonter cette limitation est simplement de ne pas répéter les valeurs d'étiquette de flux pour une communication établie dans un hôte. Cela signifie fondamentalement que chaque fois qu'une nouvelle communication est établie en utilisant des ULID différents, une nouvelle valeur d'étiquette de flux lui est allouée. De cette façon, chaque communication qui utilise des ULID différents peut être différenciée parce que chacune a une valeur différente d'étiquette de flux.

Le problème d'une telle approche est qu'elle exige que le receveur de la communication alloue la valeur d'étiquette de flux utilisée pour les paquets entrants, afin de les allouer de façon univoque. Pour cela, une négociation d'ajustement de la valeur de l'étiquette de flux à utiliser dans la communication est nécessaire avant d'échanger les paquets de données. Cela pose des problèmes avec les hôtes sans capacité Shim6, car ils ne vont pas être capables de négocier une valeur acceptable pour l'étiquette de flux. Cette limitation peut être évitée en démarquant les paquets qui appartiennent aux sessions Shim de ceux qui ne le font pas. Ces marquages exigeraient au moins un bit dans l'en-tête IPv6, qui n'est actuellement pas disponible, de sorte que des options plus créatives seraient exigées, par exemple, en utilisant de nouvelles valeurs de Prochain en-tête pour indiquer que le paquet appartient à une communication à capacité Shim6 et que l'étiquette de flux porte les informations de contexte, comme proposé dans [8]. Cependant, même si de nouvelles valeurs de Prochain en-tête sont utilisées de cette façon, cette approche est incompatible avec la capacité d'établissement différé du protocole Shim, qui est une fonction préférée car elle supprime les délais dus à l'établissement du contexte Shim avant l'initiation de la communication. Une telle capacité permet aussi aux nœuds de définir à quelle étape de la communication ils décident, sur la base de leur propre politique, qu'une certaine communication exige la protection par Shim.

Afin de tenir compte des limitations identifiées, un autre approche qui ne contraigne pas les valeurs d'étiquette de flux utilisées par les communications utilisant des ULID égaux aux localisateurs (c'est-à-dire, sans traduction Shim) est d'exiger seulement que différentes valeurs d'étiquette de flux soient allouées aux différents contextes Shim. Dans une telle approche, les communications commencent avec un usage non modifié de l'étiquette de flux (qui pourrait être zéro ou comme suggéré dans la [RFC3697]). Les paquets envoyés après une défaillance quand une paire de localisateurs différente est utilisée auraient une étiquette de flux complètement différente, et cette étiquette de flux pourrait être allouée par le receveur au titre de l'établissement du contexte Shim. Comme elle est allouée durant l'établissement de contexte, le receveur des paquets "défaillants" peut prendre une étiquette de flux de son choix (qui est unique dans le sens où aucun autre contexte ne l'utilise comme étiquette de contexte) sans aucun impact de performance, respectant que, pour chaque paire de localisateurs, la valeur de l'étiquette de flux utilisée pour une paire de localisateurs donnée ne change pas du fait du fonctionnement de l'ajustement multi rattachements.

Dans cette approche, la contrainte est que les valeurs d'étiquette de flux utilisées comme identifiants de contexte ne peuvent pas être utilisées par d'autres communications qui utilisent des ensembles de localisateurs non disjoints. Cela signifie que une fois d'une valeur d'étiquette de flux a été allouée à un contexte Shim qui a certains ensembles de localisateurs associés, la même valeur ne peut pas être utilisée pour d'autres communications qui utilisent une paire d'adresses contenue dans les ensembles de localisateurs du contexte. C'est une contrainte dans les stratégies potentielles d'allocation d'étiquette de flux .

Une façon possible de contourner cette contrainte est de marquer les paquets Shim qui exigent une traduction, afin de les différencier des paquets IPv6 réguliers, en utilisant les valeurs artificielles de Prochain en-tête décrites ci-dessus. Dans ce

cas, les valeurs d'étiquette de flux contraintes sont seulement celles des paquets qui sont traduits pas le Shim. Cette dernière approche serait la préférée si l'étiquette de contexte doit être portée dans le champ Étiquette de flux. C'est le cas non seulement parce que cela impose les contraintes minimum aux stratégies d'allocation d'étiquette de flux limitant les restrictions seulement aux paquets qui ont besoin d'être traduits par le Shim, mais aussi parce que les mécanismes de détection de perte de contexte bénéficient largement du fait que les paquets de données de Shim sont identifiés comme tels, permettant à l'extrémité receveuse d'identifier si un contexte Shim associé à un paquet reçu est supposé exister, comme va être discuté dans l'appendice sur la détection de perte de contexte ci-dessous.

D.2.2 En-tête d'extension

Une autre approche, qui est celle choisie par ce protocole, est de porter l'étiquette de contexte dans un nouvel en-tête Extension. Ces étiquettes de contexte sont allouées par l'extrémité receveuse durant la négociation initiale du protocole Shim6, ce qui implique que chaque contexte va avoir deux étiquettes de contexte, une pour chaque direction. Les paquets de données vont être démultiplexés en utilisant l'étiquette de contexte portée dans l'en-tête Extension. Cela semble une approche correcte car elle ne surcharge pas les champs existants. Cependant, elle introduit des frais généraux supplémentaires dans le paquet du fait de l'en-tête additionnel. Les frais généraux supplémentaires introduits sont de 8 octets. Cependant, on devrait noter que l'étiquette de contexte est seulement requise quand un localisateur autre que celui utilisé comme ULID est contenu dans le paquet. Les paquets où les champs Adresse de source et Adresse de destination contiennent les ULID n'exigent pas d'étiquette de contexte, car aucune réécriture n'est nécessaire chez le receveur. Cette approche réduirait les frais généraux parce que l'en-tête supplémentaire n'est exigé qu'après une défaillance. Par ailleurs, cette approche causerait des changements dans la MTU disponible pour certains paquets, car les paquets qui incluent l'en-tête Extension vont avoir une MTU plus courte de 8 octets. Cependant, les changements de chemin à travers le réseau peuvent résulter en une MTU différente dans tous les cas ; donc, avoir un changement de localisateur, qui implique un changement de chemin, et affecte la MTU, n'introduit pas de nouveaux problèmes.

D.3 Détection de perte de contexte

Dans cet appendice, on présente les différentes approches considérées pour détecter la perte de contexte et de potentielles stratégies de récupération de contexte. Le scénario considéré est le suivant : le Nœud A et le Nœud B communiquent en utilisant IPA1 et IPB1. Un peu plus tard, un contexte Shim est établi entre eux, avec, respectivement, IPA1 et IPB1 comme ULID et avec IPA1,...,IPAn et IPB1,...,IPBm comme ensembles de localisateurs.

Il peut arriver, plus tard, qu'un des hôtes (par exemple, l'Hôte A) perde le contexte Shim. La raison peut en être que l'Hôte A a une politique de collecte plus agressive que l'Hôte B ou qu'une erreur s'est produite dans la couche Shim chez l'Hôte A et a résulté en la perte de l'état du contexte.

Les mécanismes considérés dans cet appendice visent à traiter ce problème. Il y a essentiellement deux tâches qui doivent être effectuées afin de traiter ce problème : d'abord, la perte de contexte doit être détectée et, ensuite, le contexte doit être récupéré/rétabli.

Les mécanismes pour détecter la perte de contexte.

Ces mécanismes consistent en ce que chaque extrémité du contexte envoie périodiquement un paquet contenant des informations spécifiques du contexte à l'autre extrémité. À réception de ces paquets, le receveur vérifie que le contexte requis existe. Si le contexte n'existe pas, il envoie un paquet notifiant le problème à l'envoyeur.

Une solution de remplacement évidente serait de créer un échange de maintien en vie spécifique du contexte, consistant en l'envoi périodique de paquets dans ce but. Cette option a été considérée et éliminée parce que il semblait exagéré de définir un nouvel échange de paquets pour traiter ce problème.

Une autre solution de remplacement est de porter la fonction de détection de perte de contexte dans un autre échange de paquets existant. En particulier, les deux paquets de contrôle et de données Shim peuvent être utilisés pour cela.

Les paquets de contrôle Shim peuvent être utilisés sans problème pour cela parce que ils portent des informations spécifiques du contexte. De cette façon, quand un nœud reçoit un tel paquet, il vérifie si le contexte existe. Cependant, la fréquence du contrôle Shim peut n'être pas adéquate pour la détection de perte de contexte car les échanges de paquets de contrôle peuvent être très limités pour une session dans certains scénarios.

Les paquets de données sont par ailleurs supposés être échangés à une fréquence plus élevée mais ne portent pas nécessairement des informations spécifiques du contexte. En particulier, les paquets qui s'écoulent avant un changement de

localisateur (c'est-à-dire, un paquet portant les ULID dans les champs d'adresses) n'ont pas besoin d'informations de contexte car il n'ont pas besoin de traitement Shim. Les paquets qui portent des localisateurs différents des ULID portent des informations de contexte.

Cependant, on a besoin ici de faire une distinction entre les différentes approches considérées pour porter l'étiquette de contexte -- en particulier, entre les approches où les paquets sont explicitement marqués comme des paquets Shim et celles où les paquets ne sont pas marqués comme tels. Par exemple, dans le cas où l'étiquette de contexte est portée dans l'étiquette de flux et où les paquets ne sont pas marqués comme des paquets Shim (c'est-à-dire, aucune nouvelle valeur de prochain en-tête n'est définie pour Shim) un receveur qui a perdu le contexte associé n'est pas capable de détecter que le paquet est associé à un contexte manquant. Il en résulte que le paquet va être passé inchangé au protocole de couche supérieure, qui à son tour va probablement l'éliminer en silence du fait d'une erreur de somme de contrôle. Le comportement résultant est que la perte de contexte n'est pas détectée. C'est une raison supplémentaire pour éliminer une approche qui porte l'étiquette de contexte dans le champ d'étiquette de flux et ne marque pas explicitement les paquets Shim comme tels. Par ailleurs, les approches qui marquent les paquets de données Shim (comme ceux qui utilisent l'en-tête Extension ou l'étiquette de flux avec de nouvelles valeurs de Prochain en-tête) permettent au receveur de détecter si le contexte associé au paquet reçu manque. Dans ce cas, les paquets de données effectuent aussi la fonction d'un échange de détection de perte de contexte. Cependant, on doit noter que seuls les paquets qui portent un localisateur différent de l'ULID sont marqués. Cela signifie que la perte de contexte va être détectée après qu'une panne s'est produite, c'est-à-dire, que des localisateurs de remplacement sont utilisés.

En résumé, les mécanismes de détection de perte de contexte proposés utilisent les paquets de contrôle Shim et les en-têtes Extension de charge utile Shim6 pour détecter la perte de contexte. Les paquets de contrôle Shim détectent la perte de contexte durant toute la durée de vie du contexte, mais la fréquence attendue dans certains cas est très faible. Par ailleurs, les en-têtes Extension de charge utile Shim6 ont une fréquence attendue en général supérieure, mais elles détectent seulement la perte de contexte après une panne. Ce comportement implique qu'il va être courant que la perte de contexte soit détectée après une défaillance, c'est-à-dire, une fois qu'elle est réellement nécessaire. À cause de cela, un mécanisme pour récupérer de la perte de contexte est exigé si cette approche est utilisée.

Globalement, le mécanisme de détection d'un contexte perdu va fonctionner comme suit : l'extrémité qui a encore le contexte disponible envoie un message se référant au contexte. À réception de ce message, l'extrémité qui a perdu le contexte identifie la situation et notifie à l'autre extrémité l'événement de perte de contexte en envoyant un paquet contenant les informations de contexte perdu extraites du paquet reçu.

Une option est de simplement envoyer un message d'erreur contenant les paquets reçus (ou au moins autant du paquet reçu que le permet la MTU). Un des buts de cette notification est de permettre à l'autre extrémité qui conserve encore l'état de contexte de rétablir le contexte perdu. Le mécanisme pour rétablir le contexte perdu consiste à effectuer la prise de contact initiale en quatre phases. C'est un échange qui prend du temps et, à ce moment, le temps peut être critique car on rétablit un contexte actuellement nécessaire (parce que la détection de perte de contexte peut se produire après une défaillance). Une autre option, qui est celle utilisée dans ce protocole, est de remplacer le message d'erreur par un message R1 modifié afin que le temps nécessaire pour effectuer l'échange d'établissement de contexte puisse être réduit. À réception de ce message R1 modifié, l'extrémité qui a toujours l'état du contexte peut finir l'échange d'établissement de contexte et restaurer le contexte perdu.

D.4 Sécurisation des ensembles de localisateurs

L'adoption d'un protocole comme SHIM, qui permet le lien d'un ULID avec un ensemble de localisateurs, ouvre la porte à différents types d'attaques de redirection comme décrit dans la [RFC4218]. Le but, en termes de sécurité, pour la conception du protocole Shim est de ne pas introduire de nouvelle vulnérabilité dans l'architecture de l'Internet. Le but n'est pas de fournir une protection supplémentaire autre que celle actuellement disponible dans l'Internet IPv6 à rattachement simple.

Plusieurs mécanismes de sécurité ont été envisagés pour protéger le protocole Shim. Dans cet appendice, on présente certains d'entre eux.

La plus simple option de protection du protocole Shim est d'utiliser des mouchards *(cookies)* c'est-à-dire, une chaîne de bits générée aléatoirement qui est négociée durant la phase d'établissement de contexte et est ensuite incluse dans les messages de signalisation suivants. Ainsi, il serait possible de vérifier que la partie qui a été impliquée dans la prise de contact initiale est la même que celle qui introduit les nouveaux localisateurs. De plus, avant d'utiliser un nouveau localisateur, un échange est effectué à travers le nouveau localisateur, vérifiant que la partie située au nouveau localisateur connaît le mouchard, c'est-à-dire, que c'est la même partie qui a effectué la prise de contact initiale.

Bien que ce mécanisme de sécurité fournisse bien une protection considérable, il laisse la porte ouverte à ce qu'on appelle des attaques en temps glissant. Dans ces attaques, un attaquant sur le chemin découvre le mouchard en envoyant un message de signalisation quelconque. Après cela, l'attaquant peut laisser le chemin et effectuer une attaque en redirection qui, comme il est en possession du mouchard, peut introduire un nouveau localisateur dans l'ensemble de localisateurs et peut aussi réussir à effectuer l'échange d'accessibilité si il est capable de recevoir des paquets au nouveau localisateur. La différence avec la situation actuelle de IPv6 à rattachement unique est que dans la situation actuelle, l'attaquant a besoin d'être sur le chemin durant toute la durée de l'attaque, tandis que dans cette nouvelle situation (où seulement la protection du mouchard est fournie) un attaquant qui a été sur le chemin peut effectuer des attaques après qu'il a quitté cette situation en chemin.

De plus, parce que le mouchard est inclus dans les messages de signalisation, l'attaquant peut découvrir le mouchard en reniflant l'un d'eux, rendant le protocole vulnérable durant toute la vie du contexte Shim. Une approche possible pour augmenter la sécurité est d'utiliser un secret partagé, c'est-à-dire, une chaîne binaire qui est négociée durant la prise de contact initiale mais est utilisée comme clé pour protéger les messages suivants. Avec cette technique, l'attaquant doit être présent sur le chemin et renifler les paquets durant la prise de contact initiale, car c'est le seul moment où le secret partagé est échangé. Bien que cela impose que l'attaquant soit sur le chemin à ce moment très spécifique (la phase d'établissement) et bien qu'elle améliore la sécurité, cette approche est quand même vulnérable aux attaques en temps glissant. On devrait noter que, selon les détails du protocole, un attaquant peut être capable de forcer la re-création de la prise de contact initiale (par exemple, en bloquant les messages et en faisant croire aux parties que le contexte a été perdu) ; donc, la situation résultante peut ne pas différer beaucoup de l'approche fondée sur le mouchard.

Une autre option qui a été discutée durant la conception de ce protocole était la possibilité d'utiliser IPsec pour protéger le protocole Shim. Maintenant, le problème considéré dans ce scénario est comment lier de façon sûre une adresse qui est utilisé comme ULID avec un ensemble de localisateurs qui puisse être utilisé pour échanger des paquets. Le mécanisme fourni par IPsec pour lier de façon sûre l'adresse utilisée avec des clés cryptographiques est l'usage de certificats numériques. Cela implique qu'une solution fondée sur IPsec exigerait qu'un tiers commun et mutuellement de confiance génère des certificats numériques qui lient la clé et l'ULID. Considérant que la portée d'application du protocole Shim est mondiale, cela impliquerait une infrastructure de clé publique (PKI, *Public Key Infrastructure*) mondiale. Les problèmes majeurs de cette approche sont les difficultés de déploiement associées à une PKI mondiale. L'autre possibilité serait d'utiliser une forme IPsec opportuniste, comme la sécurité mieux que rien (BTNS, *Better-Than-Nothing-Security*) [RFC5386]. Cependant, cela présenterait encore des problèmes. En particulier, cette approche exige un acte de foi afin de lier une adresse donnée à la clé publique utilisée, ce qui empêcherait en fait les plus critiques caractéristiques de sécurité que la solution de sécurité Shim6 a besoin de réaliser d'être fournies : prouver la propriété de l'identifiant. En plus de cela, l'utilisation de IPsec exigerait d'activer AH/ESP par paquet juste pour que le multi rattachements se produise.

En général, SHIM6 était supposé fonctionner entre des paires d'hôtes qui n'ont pas d'accord préalable, une association de sécurité, ou un tiers de confiance commun. Il était aussi vu comme indésirable de devoir activer AH/ESP par paquet juste pour que le multi rattachements se produise. Cependant, Shim6 devrait fonctionner et avoir un niveau de sécurité supplémentaire lorsque deux hôtes choisissent d'utiliser IPsec.

Une autre solution de remplacement aurait été d'employer une forme de IPsec opportuniste ou de sécurité mieux que rien (BTNS) pour effectuer ces tâches avec IPsec à la place. Essentiellement, HIP en mode opportuniste est très similaire à SHIM6, sauf que HIP utilise IPsec, emploie ESP par paquet, et introduit un autre ensemble d'identifiants.

Finalement, deux technologies différentes ont été choisies pour protéger le protocole Shim : HBA [RFC5535] et CGA [RFC3972]. Ces deux techniques donnent un niveau similaire de protection mais fournissent aussi une fonctionnalité différente avec des coûts de calcul différents.

Le mécanisme HBA s'appuie sur la capacité de générer toutes les adresses d'un hôte multi rattachements comme un ensemble inaltérable d'adresses IPv6 intrinsèquement liées, appelé un ensemble de HBA. Dans cette approche, les adresses incorporent un hachage cryptographique unidirectionnel de l'ensemble de préfixes disponible dans la partie Identifiant d'interface. Il en résulte que le lien entre toutes les adresses disponibles est codé dans les adresses elles-mêmes, fournissant la protection contre la capture. Tout homologue qui utilise le nœud de protocole Shim peut efficacement vérifier que les adresses de remplacement proposées pour continuer la communication sont liées à l'adresse initiale par un simple calcul de hachage. Une limitation de la technique HBA est que, une fois générée, l'ensemble d'adresses est fixé et ne peut pas être changé sans changer aussi toutes les adresses de l'ensemble de HBA. En d'autres termes, la technique HBA ne prend pas en charge l'ajout dynamique d'adresse à un ensemble de HBA précédemment généré. Un avantage de cette approche est qu'elle exige seulement des opérations de hachage pour vérifier un ensemble de localisateurs, imposant au protocole un coût de calcul très faible.

Dans une approche fondée sur CGA, l'adresse utilisée comme ULID est une CGA qui contient un hachage d'une clé publique dans son identifiant d'interface. Le résultat est un lien sûr entre l'ULID et la paire de clés associée. Cela permet à chaque homologue d'utiliser la clé privée correspondante pour signer les messages Shim qui portent les informations d'ensemble de localisateurs. La chaîne de confiance dans ce cas est la suivante : l'ULID utilisé pour la communication est liée de façon sûre à la paire de clés parce que elle contient le hachage de la clé publique, et l'ensemble de localisateurs est lié à la clé publique par la signature. L'approche de CGA prend alors en charge l'ajout dynamique de nouveaux localisateurs dans l'ensemble de localisateurs, parce que afin de faire cela le nœud a seulement besoin de signer le nouveau localisateur avec la clé privée associée à la CGA utilisée comme ULID. Une limitation de cette approche est qu'elle impose l'usage systématique de la cryptographie de clé publique avec son coût de calcul associé.

Chacun de ces deux mécanismes, HBA et CGA, fournit une protection contre l'attaque en glissement de temps, car l'ULID est lié de façon sûre à un ensemble de localisateurs qui peut seulement être défini par le possesseur de l'ULID. Donc la décision de conception adoptée a été la prise en charge de ces deux mécanismes, HBA et CGA. De cette façon, quand seulement des ensembles stables d'adresses sont requis, les nœuds peuvent bénéficier du faible coût de calcul offert par HBA, tandis que quand des ensembles de localisateurs dynamiques sont requis, cela peut être réalisé par des CGA avec un coût supplémentaire. De plus, parce que les HBA sont définies comme une extension de CGA, les adresses disponibles dans un nœud peuvent simultanément être des CGA et des HBA, permettant l'usage de la fonction de HBA et de CGA quand nécessaire, sans exiger de changement des adresses utilisées.

D.5 Échange d'établissement de contexte de paire d'ULID

Deux options ont été considérées pour l'échange de paire d'ULID d'établissement de contexte : une prise de contact en deux phases et une prise de contact en quatre phases.

Un objectif clé de la conception de cet échange est la protection contre les attaques de DoS. L'attaque considérée était fondamentalement une situation où un attaquant lance une grande quantité de paquets de demande d'établissement de paires d'ULID, épuisant les ressources de la victime, comme dans les attaques d'inondation de SYN TCP.

Un échange de prise de contact en quatre phases protège contre ces attaques parce que le receveur ne crée aucun état associé à un contexte donné avant la réception du second paquet, qui contient la preuve avant contact sous la forme d'un jeton. À ce point, le receveur peut vérifier qu'au moins l'adresse utilisée par l'initiateur est valide dans une certaine mesure, car l'initiateur est capable de recevoir des paquets à cette adresse. Dans le pire des cas, le répondeur peut tracer l'attaquant en utilisant cette adresse. L'inconvénient de cette approche est qu'elle impose quatre échanges de paquets pour tout établissement de contexte. Cela poserait un gros problème si le contexte Shim nécessaire devait être établi d'abord, avant que la communication puisse s'établir. Cependant, grâce à la capacité d'établissement de contexte différée du protocole Shim, cette limitation a un impact réduit sur les performances du protocole. (Cependant, cela peut avoir un plus grand impact dans la situation de récupération de contexte, comme discuté précédemment. Néanmoins, dans ce cas, il est possible d'effectuer des optimisations pour réduire le nombre de paquets, comme décrit ci-dessus.)

L'autre option considérée était une prise de contact en deux phases avec la possibilité de revenir à une prise de contact en quatre phases en cas d'attaque. Dans cette approche, l'échange d'établissement de paire d'ULID consiste normalement en un double échange de paquets et ne vérifie pas que l'initiateur a effectué un contact préalable avant de créer l'état de contexte. En cas de détection d'une attaque de DoS, le répondeur revient à une prise de contact en quatre phases similaire à celle décrite précédemment, afin d'empêcher l'attaque détectée de se poursuivre. La principale difficulté avec cette attaque est comment détecter qu'un répondeur est actuellement attaqué. On devrait noter que, parce que c'est un échange en deux phases, il n'est pas possible d'utiliser de nombre de sessions semi ouvertes (comme dans TCP) pour détecter une attaque en cours ; différentes heuristiques doivent être considérées.

La décision de conception prise était que, considérant l'impact actuel des attaques de DoS et le faible impact de l'échange en quatre phases dans le protocole Shim (grâce à la capacité d'établissement de contexte différé) un échange en quatre phases serait adopté pour le protocole de base.

D.6 Mise à jour des ensembles de localisateurs

Il y a deux approches possibles à l'ajout et la suppression de localisateurs : l'approche atomique et l'approche différentielle. L'approche atomique envoie essentiellement l'ensemble complet de localisateurs chaque fois qu'une variation se produit dans l'ensemble de localisateurs. L'approche différentielle envoie les différences entre l'ensemble de localisateurs existant et le nouveau. L'approche atomique impose des frais généraux supplémentaires car tout l'ensemble de localisateurs doit être échangé à chaque fois, tandis que l'approche différentielle exige la re-synchronisation des deux extrémités à travers les changements (c'est-à-dire, exige que les deux extrémités aient la même idée de ce qu'est l'ensemble actuel de localisateurs).

À cause des difficultés imposées par l'exigence de synchronisation, l'approche atomique a été choisie.

D.7 Nettoyage d'état

Il y a essentiellement deux approches pour éliminer un état existant des localisateurs, clés, et identifiants d'un nœud correspondant : une approche coordonnée et une approche unilatérale.

Dans l'approche unilatérale, chaque nœud élimine les informations sur l'autre nœud sans coordination avec l'autre nœud, sur la base de temporisateurs et heuristiques locaux. Aucun échange de paquets n'est exigé pour cela. Dans ce cas, il serait possible qu'un des nœuds ait éliminé l'état alors que l'autre nœud l'a conservé. Dans ce cas, un message d'erreur No Context peut être nécessaire pour informer l'autre nœud de la situation ; éventuellement un mécanisme de récupération est aussi nécessaire.

Une approche coordonnée utiliserait un mécanisme CLOSE explicite, comme celui spécifié dans HIP [RFC5201]. Si une prise de contact explicite CLOSE et le temporisateur associé est utilisé, alors il ne sera plus nécessaire d'avoir le message d'erreur No Context du fait qu'un homologue a une mauvaise collecte à son extrémité du contexte. Cependant, il y a éventuellement toujours besoin d'un message d'erreur No Context dans le cas d'une perte d'état complète de l'homologue (aussi appelé une panne suivie d'un réamorçage). C'est seulement si on suppose que le réamorçage prend au moins le temps du temporisateur CLOSE, ou si il y a accord pour ne pas fournir de service complet jusqu'à CLOSE-timer minutes après la panne, qu'on peut complètement abandonner le message d'erreur No Context.

De plus, un autre aspect pertinent pour ce choix de conception est le problème de la confusion de contexte. En particulier, utiliser l'approche unilatérale pour éliminer l'état de contexte ouvre clairement la possibilité de confusion de contexte, lorsque une des extrémités élimine unilatéralement l'état du contexte, alors que l'autre ne le fait pas. Dans ce cas, l'extrémité qui a éliminé l'état peut réutiliser la valeur d'étiquette de contexte utilisée pour l'état éliminé pour un autre contexte, créant une potentielle confusion de contexte. Afin d'illustrer les cas où des problèmes vont se poser, on considère le scénario suivant :

- Les hôtes A et B établissent le contexte 1 en utilisant CTA et CTB comme étiquettes de contexte.
- Plus tard, A élimine le contexte 1 et la valeur d'étiquette de contexte CTA devient disponible pour réutilisation.
- Cependant, B conserve le contexte 1.

Cela va créer une confusion de contexte dans les deux cas suivants :

- Un nouveau contexte 2 est établi entre A et B avec une paire d'ULID différente (ou identifiant d'instance fourchée) et A utilise CTA comme étiquette de contexte. Si les ensembles de localisateurs utilisés pour les deux contextes ne sont pas disjoints, on a une confusion de contexte.
- Un nouveau contexte est établi entre A et C, et A utilise CTA comme valeur d'étiquette de contexte pour ce nouveau contexte. Plus tard, B envoie des messages d'en-tête Extension de charge utile et/ou de contrôle contenant CTA, ce qui pourrait être interprété par A comme appartenant au contexte 2 (si on n'y fait pas attention). Là encore on a une confusion de contexte.

On pourrait penser qu'en utilisant une approche coordonnée on éliminerait cette confusion de contexte, rendant le protocole plus simple. Cependant, ce n'est pas le cas, parce que même dans le cas d'une approche coordonnée utilisant un échange CLOSE/CLOSE ACK, il y a toujours la possibilité qu'un hôte réamorce sans avoir le temps d'effectuer l'échange CLOSE. Donc, il est vrai que l'approche coordonnée élimine la possibilité de confusion de contexte du fait d'une mauvaise collecte prématurée, mais cela n'empêche pas les mêmes situations du fait d'une panne et réamorçage d'un des hôtes impliqués. Le résultat est que, même si on a une approche coordonnée, on va quand même devoir traiter la confusion de contexte et fournir le moyen de détecter et récupérer de ces situations.

Adresse des auteurs

Erik Nordmark
Sun Microsystems
17 Network Circle
Menlo Park, CA 94025
USA
téléphone : +1 650 786 2921
mél : erik.nordmark@sun.com

Marcelo Bagnulo
Universidad Carlos III de Madrid
Av. Universidad 30
Leganes, Madrid 28911
ESPANA
téléphone : +34 91 6248814
mél : marcelo@it.uc3m.es
URI : http://www.it.uc3m.es