Groupe de travail Réseau
Request for Comments : 5357
Catégorie : Sur la voie de la normalisation

K. Hedayat, Brix Networks
R. Krzanowski, Verizon
A. Morton, AT&T Labs
K. Yum, Juniper Networks
J. Babiarz, Nortel Networks

Traduction Claude Brière de L'Isle
octobre 2008

Protocole de mesures actives bidirectionnelles (TWAMP)

Résumé
Le protocole de mesures actives unidirectionnelles (OWAMP, *One-way Active Measurement Protocol*) spécifié dans la RFC 4656, fournit un protocole commun pour mesurer des métriques unidirectionnelles entre des appareils du réseau. OWAMP peut être utilisé en bidirectionnel pour mesurer des métriques unidirectionnelles dans les deux directions entre deux éléments de réseau. Cependant, il ne s'accommode pas des mesures d'aller-retour ou bidirectionnelles. Le présent mémoire spécifie un protocole de mesures actives bidirectionnelles (TWAMP, *Two-Way Active Measurement Protocol*) fondé sur OWAMP, qui ajoute des capacités de mesure bidirectionnelles ou d'aller-retour. L'architecture de mesures TWAMP est généralement constituée entre deux hôtes qui ont des rôles spécifiques, et cela permet des simplifications du protocole, en faisant une solution de remplacement intéressante dans certaines circonstances.

Table des matières

1. Introduction

L'équipe d'ingénierie de l'Internet (IETF, *Internet Engineering Task Force*) a réalisé une proposition de norme sur la métrique du délai d'aller-retour [RFC2681]. L'IETF a aussi produit un protocole pour le contrôle et la collecte de mesures unidirectionnelles, le protocole de mesures actives unidirectionnelles (OWAMP, *One-way Active Measurement Protocol*) [RFC4656]. Cependant, OWAMP ne traite pas les mesures d'aller-retour ou bidirectionnelles.

Les mesures bidirectionnelles sont courantes dans les réseaux IP, principalement parce que la synchronisation entre les horloges, locale et distante, n'est pas nécessaire pour le délai d'aller-retour, et la prise en charge des mesures à l'extrémité distante peut être limitée à une simple fonction d'écho. Cependant, la facilité la plus courante pour les mesures de délai d'aller-retour est la demande/réponse d'écho ICMP (utilisée par l'outil ping) et les problèmes de cette méthode sont documentés au paragraphe 2.6 de la [RFC2681]. Le présent mémoire spécifie le protocole de mesures actives bidirectionnelles (TWAMP, *Two-Way Active Measurement Protocol*). TWAMP utilise la méthodologie et l'architecture de OWAMP [RFC4656] pour définir un protocole ouvert pour les mesures de métriques bidirectionnelles ou d'aller-retour (à partir d'ici dans ce document le terme bidirectionnel signifie aussi aller-retour) en plus des métriques unidirectionnelles de OWAMP. TWAMP emploie des horodatages appliqués à la destination d'écho (réflecteur) pour permettre une plus grande précision (il peut tenir compte des délais). L'architecture de mesures de TWAMP est généralement constituée seulement de deux hôtes avec des rôles spécifiques, et cela permet des simplifications du protocole, qui en font une solution de remplacement intéressante de OWAMP dans certaines circonstances.

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].

1.1 Relations entre les protocoles d'essais et de contrôle

Comme OWAMP [RFC4656], TWAMP consiste en deux protocoles en inter relations : TWAMP-Control et TWAMP-Test. Les relations de ces protocoles sont celles définies au paragraphe 1.1 de OWAMP [RFC4656]. TWAMP-Control est utilisé pour initier, démarrer, et arrêter les sessions d'essais, tandis que TWAMP-Test est utilisé pour échanger des paquets d'essai entre deux entités TWAMP.

1.2 Modèle logique

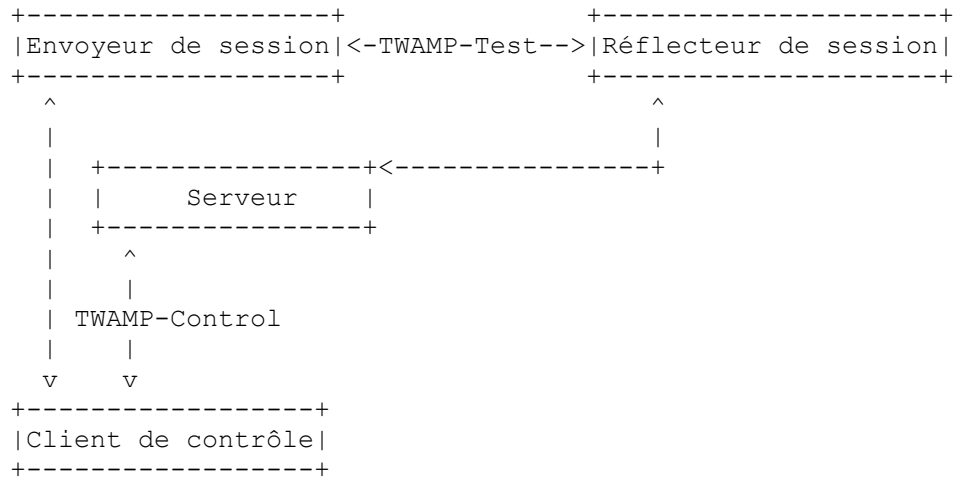
Le rôle et la définition des entités logiques sont comme défini au paragraphe 1.2 de OWAMP [RFC4656] avec les exceptions suivantes:

- Le receveur de session est appelé le réflecteur de session dans l'architecture TWAMP. Le réflecteur de session a la capacité de créer et envoyer un paquet de mesure quand il reçoit un paquet de mesure. À la différence du receveur de session, le réflecteur de session ne collecte aucune information de paquet.

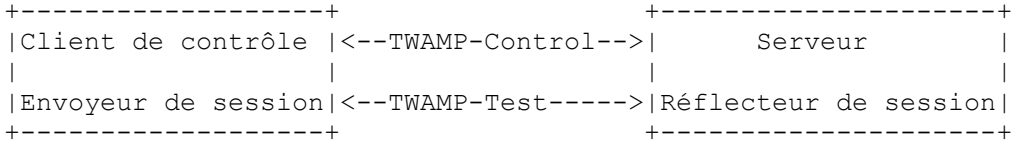
- Le serveur est un système d'extrémité qui gère une ou plusieurs sessions TWAMP, et est capable de configurer l'état par session dans les points d'extrémité. Cependant, un serveur associé à un réflecteur de session n'aurait pas la capacité de retourner les résultats d'une session d'essais, et c'est une différence d'avec OWAMP.

- L'entité Client de collecte n'existe pas dans l'architecture TWAMP, parce que le réflecteur de session ne collecte aucune information de paquet à aller chercher. Par conséquent, il n'y a pas besoin d'un client de collecte.

Un exemple des scénarios de relations possible entre ces rôles est présenté ci-dessous. Dans cet exemple, différents rôles logiques sont joués sur différents hôtes. Les liaisons non étiquetées dans la figure ne sont pas spécifiées dans ce document et peuvent être des protocoles propriétaires.



Comme dans OWAMP [RFC4656], des rôles logiques différents peuvent être joués par le même hôte. Par exemple, dans la figure ci-dessus, il pourrait en fait y avoir deux hôtes : un jouant les rôles de client de contrôle et d'envoyeur de session, et l'autre jouant les rôles de serveur et de réflecteur de session. Cet exemple est montré ci-dessous.



1.3 Guide de prononciation

L'acronyme OWAMP est généralement prononcé en deux syllabes, oh wamp.

L'acronyme TWAMP est aussi prononcé en deux syllabes, té wamp.

2. Vue d'ensemble du protocole

Le protocole de mesures actives bidirectionnel est un protocole ouvert pour la mesure de métriques bidirectionnelles. Il est fondé sur OWAMP [RFC4656] et adhère à l'architecture et la conception globale de OWAMP. Les protocoles TWAMP-Control et TWAMP-Test réalisent leurs tâches d'essais comme mentionné ci-dessous :

Le client de contrôle initie une connexion TCP sur l'accès bien connu de TWAMP, et le serveur (son rôle étant maintenant établi) répond avec son message d'accueil, indiquant le mode de sécurité/intégrité qu'il veut prendre en charge.

- Le client de contrôle répond avec le mode de communication choisi et les informations de prise en charge de la protection d'intégrité et de chiffrement, si le mode les exige. Le serveur répond accepter le mode et donne son heure de début. Cela achève l'établissement de la connexion de contrôle.

- Le client de contrôle demande (et décrit) une session d'essais avec un unique message TWAMP-Control. Le serveur répond avec son acceptation et les informations de prise en charge. Plus d'une session d'essais peut être demandée avec des messages supplémentaires.

- Le client de contrôle initie tous les essais demandés avec un message Start-Sessions, et le serveur en accuse réception.

- L'envoyeur de session et le réflecteur de session échangent des paquets d'essai en accord avec le protocole TWAMP-Test pour chaque session active.

- Quand c'est approprié, le client de contrôle envoie un message pour arrêter toutes les sessions d'essai.

Il y a deux mécanismes d'extension reconnus dans le protocole TWAMP.

1) Le champ Modes est utilisé pour établir les options de la communication durant l'établissement de la connexion à TWAMP-Control.

2) Le numéro de commande de TWAMP-Control est un autre mécanisme d'extension prévu, qui permet que des commandes supplémentaires soient définie à l'avenir.

Le protocole TWAMP-Control assure différents niveaux de capacité entre le client de contrôle et le serveur.

Toutes les quantités multi-octets définies dans ce document sont représentées comme des entiers non signés dans l'ordre des octets du réseau, sauf mention contraire.

Dans ce mémoire, les bits marqués MBZ (Must Be Zero) DOIVENT être réglés à zéro par les envoyeurs et DOIVENT être ignorés des receveurs.

3. TWAMP-Control

TWAMP-Control est dérivé de OWAMP-Control pour les mesures bidirectionnelles. Tous les messages TWAMP-Control sont de format similaire et suivent des lignes directrices similaires à celles définies à la Section 3 de OWAMP [RFC4656] avec les exceptions mentionnées dans les paragraphes qui suivent. Une de ces exceptions est la commande Fetch-Session, qui n'est pas utilisée dans TWAMP.

3.1 Établissement de connexion

L'établissement de connexion de TWAMP suit la même procédure que définie au paragraphe 3.1 de OWAMP [RFC4656]. Le champ Modes est un mécanisme d'extension reconnu dans TWAMP, et les valeurs de mode courantes sont identiques à celles utilisées dans OWAMP. La seule exception est le numéro d'accès bien connu pour TWAMP-Control. Un client ouvre une connexion TCP au serveur sur l'accès bien connu 862. L'hôte qui initie la connexion TCP prend les rôles de client de contrôle et (dans la mise en œuvre à deux hôtes) d'envoyeur de session. L'hôte qui accuse réception de la connexion TCP accepte les rôles de serveur et (dans la mise en œuvre à deux hôtes) de réflecteur de session.

Le client de contrôle PEUT régler un codet désiré dans le champ Codet Diffserv (DSCP, *Diffserv Code Point*) dans l'en-tête IP pour TOUS les paquets d'une connexion de contrôle spécifique. Le serveur DEVRAIT utiliser le DSCP TCP SYN du client de contrôle dans TOUS les paquets suivants sur cette connexion (évitant toute ambiguïté en cas de nouveau marquage).

Il existe une possibilité que le client de contrôle ait une défaillance après l'établissement de la connexion TWAMP-Control, ou d'une défaillance du chemin entre le client de contrôle et le serveur alors qu'une connexion est en cours. Le serveur PEUT arrêter toute connexion de contrôle établie quand aucun paquet associé à cette connexion n'a été reçu dans les SERVWAIT secondes. Le serveur DEVRA suspendre l'activité de surveillance de la connexion de contrôle après la réception d'une commande Start-Sessions, et DEVRA reprendre après la réception d'une commande Stop-Sessions (SI l'option SERVWAIT est prise en charge). Noter que le temporisateur REFWAIT (décrit ci-dessous) couvre les défaillances durant les sessions d'essai, et si REFWAIT expire sur TOUTES les sessions d'essai initiées par une connexion TWAMP-Control, alors la surveillance de SERVWAIT DEVRA reprendre (bien qu'une commande Stop-Sessions ait été reçue). Une mise en œuvre qui prend en charge le temporisateur SERVWAIT DEVRAIT aussi mettre en œuvre le temporisateur REFWAIT. La valeur par défaut de SERVWAIT DEVRA être 900 secondes, et ce temps d'attente PEUT être configurable. Cette temporisation permet au serveur de libérer les ressources en cas de défaillance.

Le serveur et le client utilisent tous deux les mêmes transpositions des KeyID en des secrets partagés. Le serveur, étant prêt à conduire des sessions avec plus d'un client, utilise les KeyID pour choisir la clé secrète appropriée ; un client va normalement avoir différentes clés secrètes pour des serveurs différents. Le secret partagé est une phrase de passe. Pour maximiser l'interopérabilité des phrases de passe, le jeux de caractères de la phrase de passe DOIT être codé en utilisant l'Appendice B de la [RFC5198] (définition ASCII du terminal virtuel de réseau). Il NE DOIT PAS contenir de saut à la ligne (toute combinaison de caractères retour-chariot (CR, *Carriage-Return*) et/ou saut à la ligne (LF, *Line-Feed*)) et les caractères de contrôle DEVRAIT être évitée.

3.2 Protection de l'intégrité

La protection de l'intégrité de TWAMP suit la même procédure que définie au paragraphe 3.2 de OWAMP [RFC4656]. Comme dans OWAMP, chaque code d'authentification de message haché (HMAC, *Hashed Message Authentication Code*) envoyé couvre tout ce qui est envoyé dans une direction entre les HMAC précédents (mais ceux-ci non inclus) et le début des nouveaux HMAC. De cette façon, une fois le chiffrement établi, chaque bit de la connexion TWAMP-Control est authentifié exactement une fois par un HMAC.

Noter que le message Server-Start (envoyé par un serveur durant les échanges initiaux de connexion de contrôle) ne se termine pas avec un champ HMAC. Donc, le HMAC dans le premier message Accept-Session couvre aussi le message Server-Start et inclut le champ Start-Time dans le calcul de HMAC.

Aussi, dans les modes authentifié et chiffré, le HMAC dans les paquets TWAMP-Control est chiffré.

3.3 Valeurs du champ Accept

Les valeurs de Accept utilisées dans TWAMP sont les mêmes que celles définies au paragraphe 3.3 de OWAMP [RFC4656].

3.4 Commandes TWAMP-Control

Les commandes TWAMP-Control se conforment aux règles définies au paragraphe 3.4 de OWAMP [RFC4656].

Les commandes suivantes sont disponibles pour le client de contrôle : Request-TW-Session, Start-Sessions, et Stop-Sessions. Le serveur peut envoyer des messages spécifiques en réponse aux commandes qu'il reçoit (comme décrit dans les paragraphes qui suivent).

Noter que la commande OWAMP Request-Session est remplacée par la commande TWAMP Request-TW-Session, et que la commande Fetch-Session n'apparaît pas dans TWAMP.

3.5 Création de sessions d'essai

La création de session d'essai suit la même procédure que définie au paragraphe 3.5 de OWAMP [RFC4656]. La commande Request-TW-Session se fonde sur la commande OWAMP Request-Session, et utilise le format de message décrit au paragraphe 3.5 de OWAMP, mais sans les champs Descriptions de créneau de programmation et utilise seulement un HMAC. Voici la description du format de Request-TW-Session.

Dans TWAMP, le premier octet est appelé le numéro de commande, et le numéro de commande est un mécanisme d'extension reconnu. Les lecteurs sont invités à consulter le registre des numéros de commande TWAMP-Control pour déterminer si des valeurs supplémentaires ont été allouées.

La valeur de numéro de commande 5 indique une commande Request-TW-Session, et le serveur DOIT interpréter cette commande comme une demande de session d'essais bidirectionnelle utilisant le protocole TWAMP-Test.

Si un serveur TWAMP reçoit un numéro de commande inattendu, il DOIT répondre avec le champ Accept réglé à 3 (ce qui signifie "Un aspect de la demande n'est pas accepté") dans le message Accept-Session. Les numéros de commande qui sont interdits (et éventuellement les numéros qui sont réservés) sont inattendus.

Dans OWAMP, le champ Conf-Sender est réglé à 1 quand le message Request-Session décrit une tâche où le serveur va configurer un envoyeur de paquet d'essai unidirectionnel. De même, le champ Conf-Receiver est réglé à 1 quand le message décrit la configuration pour un receveur de session. Dans TWAMP, les deux points d'extrémité envoient et reçoivent des paquets d'essai, avec l'envoyeur de session qui envoie en premier et ensuite reçoit les paquets d'essai, complété par le réflecteur de session qui reçoit d'abord et ensuite envoie.

Les deux champs Conf-Sender et Conf-Receiver DOIVENT être réglés à 0 car le réflecteur de session va recevoir et envoyer des paquets, et les rôles sont établis selon l'hôte qui initie la connexion TCP pour le contrôle. Le serveur DOIT interpréter toute valeur non zéro comme une commande mal formatée, et DOIT répondre avec le champ Accept réglé à 3 (ce qui signifie "Un aspect de la demande n'est pas accepté") dans le message Accept-Session.

Le réflecteur de session dans TWAMP ne traite pas les paquets d'essai entrants pour les métriques de performances et par conséquent n'a pas besoin de savoir le nombre de paquets entrants et leur rythme de programmation. Par conséquent le nombre de créneaux programmés et le nombre de paquets DOIVENT être réglés à 0.

L'accès d'envoyeur est l'accès UDP à partir duquel les paquets TWAMP-Test vont être envoyés et l'accès auquel les paquets TWAMP-Test vont être envoyés par le réflecteur de session (l'envoyeur de session va utiliser le même accès UDP pour envoyer et recevoir les paquets). L'accès de receveur est l'accès UDP désiré auquel les paquets TWAMP-Test vont être envoyés par l'envoyeur de session (l'accès où il est demandé au réflecteur de session de recevoir les paquets d'essai). L'accès de receveur est aussi l'accès UDP à partir duquel les paquets TWAMP-Test vont être envoyés par le réflecteur de session (le réflecteur de session va utiliser le même accès UDP pour envoyer et recevoir les paquets).

Les champs Adresse d'envoyeur et Adresse de receveur contiennent, respectivement, les adresses d'envoyeur et de receveur des points d'extrémité du chemin Internet sur lequel une session TWAMP-Test est demandée. Elles PEUVENT être réglées à 0, et dans ce cas, les adresses IP utilisées pour l'échange de messages TWAMP-Control du client de contrôle au serveur DOIVENT être utilisées dans les paquets d'essai.

L'identifiant de session (SID, *Session Identifier*) est comme défini dans OWAMP [RFC4656]. Comme le SID est toujours généré par le côté receveur, le serveur détermine le SID, et le SID dans le message Request-TW-Session DOIT être réglé à 0.

L'heure de début *(Start Time)* est comme défini dans OWAMP [RFC4656].

La temporisation est interprétée différemment de la définition de OWAMP [RFC4656]. Dans TWAMP, Temporisation est l'intervalle pendant lequel le réflecteur de session DOIT attendre après la réception d'un message Stop-Sessions. Dans le cas où des paquets d'essai sont encore en transit, le réflecteur de session DOIT le refléter si ils arrivent dans l'intervalle Temporisation qui suit la réception du message Stop-Sessions. Le réflecteur de session NE DOIT PAS refléter les paquets qui sont reçus après la temporisation.

Le descripteur de type P est comme défini dans OWAMP [RFC4656]. La seule capacité de ce champ est de régler le codet de services différenciés (DSCP) comme défini dans la [RFC2474]. La même valeur de DSCP DOIT être utilisée dans les paquets d'essai reflétés par le réflecteur de session.

Comme il n'y a pas de descriptions de créneau de programmation, le message Request-TW-Session est terminé par les champs MBZ (Must Be Zero) et HMAC. Cela achève un message logique, appelé la commande Request-TW-Session.

Le réflecteur de session DOIT répondre à chaque commande Request-TW-Session par un message Accept-Session comme défini dans OWAMP [RFC4656]. Quand le champ Accept = 0, le champ Port confirme (répète) l'accès auquel les paquets TWAMP-Test sont envoyés par l'envoyeur de session vers le réflecteur de session. En d'autres termes, le champ Port indique le numéro d'accès où le réflecteur de session s'attend à recevoir les paquets de l'envoyeur de session.

Quand l'accès de receveur demandé n'est pas disponible (par exemple, l'accès est utilisé) le serveur au réflecteur de session PEUT suggérer un autre accès disponible pour cette session dans le champ Port. L'envoyeur de session accepte l'accès de remplacement, ou compose un nouveau message Session-Request avec des paramètres convenables. Autrement, le serveur au client de contrôle utilise le champ Accept pour porter d'autres formes de rejet de session ou d'échec et NE DOIT PAS suggérer un accès de remplacement ; dans ce cas, le champ Port DOIT être réglé à zéro.

3.6 Programmations d'envoi

Le programme d'envoi des paquets d'essai défini au paragraphe 3.6 de OWAMP [RFC4656] n'est pas utilisé dans TWAMP. Le client de contrôle et l'envoyeur de session PEUVENT décider chacun de son côté du programme d'envoi. Le réflecteur de session DEVRAIT retourner chaque paquet d'essai à l'envoyeur de session aussi vite que possible.

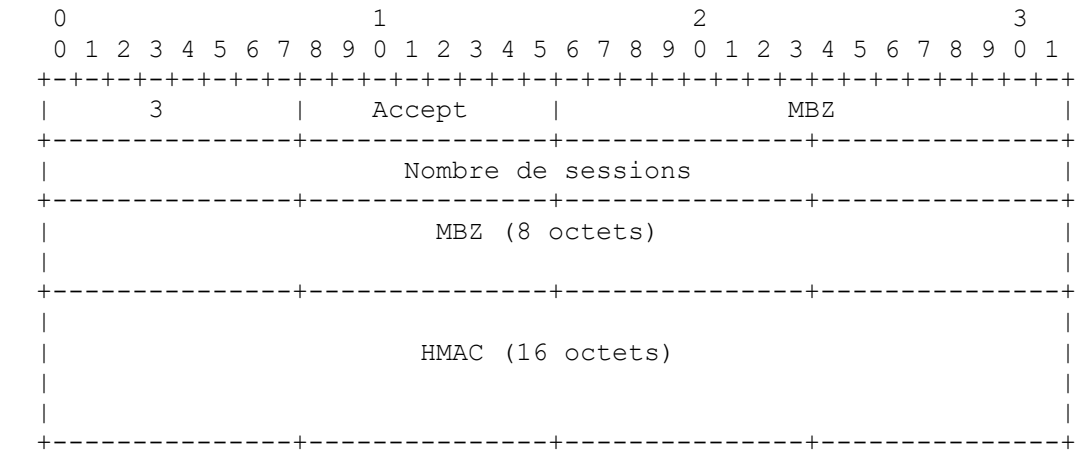
3.7 Début des sessions d'essai

La procédure et les lignes directrices pour débuter les sessions d'essai sont les mêmes que défini au paragraphe 3.7 de OWAMP [RFC4656].

3.8 Stop-Sessions

La procédure et les lignes directrices pour arrêter les sessions d'essai sont similaires à celles définies au paragraphe 3.8 de OWAMP [RFC4656]. La commande Stop-Sessions peut seulement être produite par le client de contrôle. Le message NE DOIT PAS contenir d'enregistrement de description de session ou de gammes de sauts. Le message est terminé par un seul bloc HMAC pour achever la commande Stop-Sessions. Comme la commande TWAMP Stop-Sessions ne porte pas de SID, elle s'applique à toutes les sessions demandées précédemment et débutées avec une commande Start-Sessions.

Donc, la commande TWAMP Stop-Sessions est construite comme suit :



Ci-dessus, le numéro de commande dans le premier octet (3) indique que c'est la commande Stop-Sessions.

La valeur de Accept non zéro indique une défaillance quelconque. Une valeur de zéro indique un achèvement normal (mais peut-être prématuré). La liste complète des valeurs Accept disponibles est décrite au paragraphe 3.3 de la [RFC4656], "Valeurs du champ Accept".

Si Accept a une valeur non zéro, le résultat de toutes les sessions TWAMP-Test collectées par cette session TWAMP-Control DEVRAIT être considéré comme invalide. Si le message Accept-Session n'a pas éte transmis du tout (pour une raison quelconque, y compris la défaillance de la connexion TCP utilisée pour TWAMP-Control) le résultat de toutes les sessions TWAMP-Test collectées par cette session TWAMP-Control PEUT être considéré comme invalide.

Nombre de sessions indique le nombre de sessions que le client de contrôle a l'intention d'arrêter.

Nombre de sessions DOIT contenir le nombre de sessions d'envoi commencées par le client de contrôle qui n'ont pas été terminées précédemment par une commande Stop-Sessions (c'est-à-dire, le client de contrôle DOIT tenir compte de chaque demande de session acceptée). Si le message Stop-Sessions ne tient pas compte d'exactement le nombre de sessions en cours, il est alors être considéré comme invalide, la connexion TWAMP-Control DEVRAIT être fermée, et tous les résultats obtenus considérés comme invalides.

À réception d'une commande Stop-Sessions de TWAMP-Control, le réflecteur de session DOIT éliminer tous les paquets TWAMP-Test qui arrivent à ce moment plus la temporisation (dans la commande Request-TW-Session).

3.9 Fetch-Session

Un objet de TWAMP est la mesure de métriques bidirectionnelles. Les méthodes de mesure bidirectionnelles n'exigent pas que des données soient collectées au niveau du paquet par le réflecteur de session (comme un numéro de séquence, horodatage, et durée de vie (TTL)) parce que ces données sont communiquées dans les paquets d'essai "reflétés". À ce titre, le protocole n'exige pas la restitution de données au niveau du paquet de la part du serveur et la commande OWAMP Fetch-Session n'est pas utilisée dans TWAMP.

4. TWAMP-Test

Le protocole TWAMP-Test est similaire au protocole OWAMP-test [RFC4656] à l'exception que le réflecteur de session

transmet les paquets d'essai à l'envoyeur de session en réponse à chaque paquet d'essai qu'il reçoit. TWAMP définit deux formats différents de paquet d'essai, un pour les paquets transmis par l'envoyeur de session et un pour les paquets transmis par le réflecteur de session. Comme avec le protocole OWAMP-test [RFC4656], il y a trois modes : non authentifié, authentifié, et chiffré.

4.1 Comportement de l'envoyeur

Le comportement d'envoyeur est déterminé par la configuration de l'envoyeur de session et n'est pas défini dans cette norme. De plus, le réflecteur de session n'a pas besoin de connaître le comportement de l'envoyeur de session au même degré de détail que nécessaire dans OWAMP [RFC4656]. De plus, l'envoyeur de session collecte et enregistre les informations nécessaires fournies des paquets transmis par le réflecteur de session pour la mesure des métriques bidirectionnelles. L'enregistrement des informations sur la base des paquets reçus par l'envoyeur de session est dépendant de la mise en œuvre.

4.1.1 Programmation des paquets

Comme le programme d'envoi n'est pas communiqué au réflecteur de session, il n'y a pas besoin d'un calcul normalisé du rythme des paquets.

Sans considération d'un délai de programmation, chaque paquet réellement envoyé DOIT avoir la meilleure approximation possible de son heure réelle de départ pour son horodatage (dans le paquet).

4.1.2 Format et contenu de paquet

Le format et le contenu du paquet de l'envoyeur de session suit la même procédure et lignes directrices que défini au paragraphe 4.1.2 de OWAMP [RFC4656] (à l'exception de la référence à la programmation d'envoi).

Noter que les formats de paquet d'essai du réflecteur sont plus grands que les formats d'envoyeur. L'envoyeur de session PEUT ajouter un bourrage de paquet suffisant pour permettre que les mêmes longueurs de charge utile de paquet IP soient utilisées dans chaque direction de transmission (c'est généralement désirable). Pour compenser le plus grand format de paquet d'essai du réflecteur, l'envoyeur ajoute au moins 27 octets de bourrage en mode non authentifié, et au moins 56 octets dans les modes authentifié et chiffré.

4.2 Comportement du réflecteur

TWAMP exige que le réflecteur de session transmette un paquet à l'envoyeur de session en réponse à chaque paquet qu'il reçoit.

Lorsque des paquets sont reçus, le réflecteur de session va faire ce qui suit :

- Horodater le paquet reçu. Chaque paquet réellement reçu DOIT avoir la meilleure approximation possible de son heure réelle d'arrivée entrée dans son horodatage de réception (dans le paquet).

- En mode authentifié ou chiffré, déchiffrer les sections appropriées du corps du paquet (premier bloc (16 octets) pour le mode authentifié, 96 octets pour le mode chiffré) et ensuite vérifier l'intégrité des sections couvertes par le HMAC.

- Copier le numéro de séquence du paquet dans le paquet reflété correspondant à l'envoyeur de session.

- Extraire la valeur du TTL d'envoyeur de la valeur du champ TTL/Limite de bonds des paquets reçus. Les mises en œuvre de réflecteur de session DEVRAIENT aller chercher la valeur de TTL/Limite de bonds dans l'en-tête IP du paquet, en remplaçant la valeur de 255 réglée par l'envoyeur de session. Si une mise en œuvre ne va pas chercher la valeur réelle de TTL (la seule bonne raison de ne pas le faire est l'incapacité d'accéder au champ TTL des paquets qui arrivent) elle DOIT régler la valeur du TTL d'envoyeur à 255.

- Dans les modes authentifié et chiffré, le HMAC DOIT être calculé d'abord, puis la portion appropriée du corps du paquet est chiffrée.

- Transmettre un paquet d'essai à l'envoyeur de session en réponse à chaque paquet reçu. La réponse DOIT être générée

aussitôt que possible. Le format et le contenu du paquet d'essai est défini au paragraphe 4.2.1. Avant la transmission du paquet d'essai, le réflecteur de session DOIT entrer la meilleure approximation possible de son heure d'envoi réelle comme son horodatage (dans le paquet). Cela permet la détermination du temps écoulé entre la réception du paquet et sa transmission.

- Les paquets non reçus dans le délai de temporisation (suivant la commande Stop-Sessions) DOIVENT être ignorés par le réflecteur. Le réflecteur de session NE DOIT PAS générer de paquet d'essai à l'envoyeur de session pour les paquets qui sont ignorés.

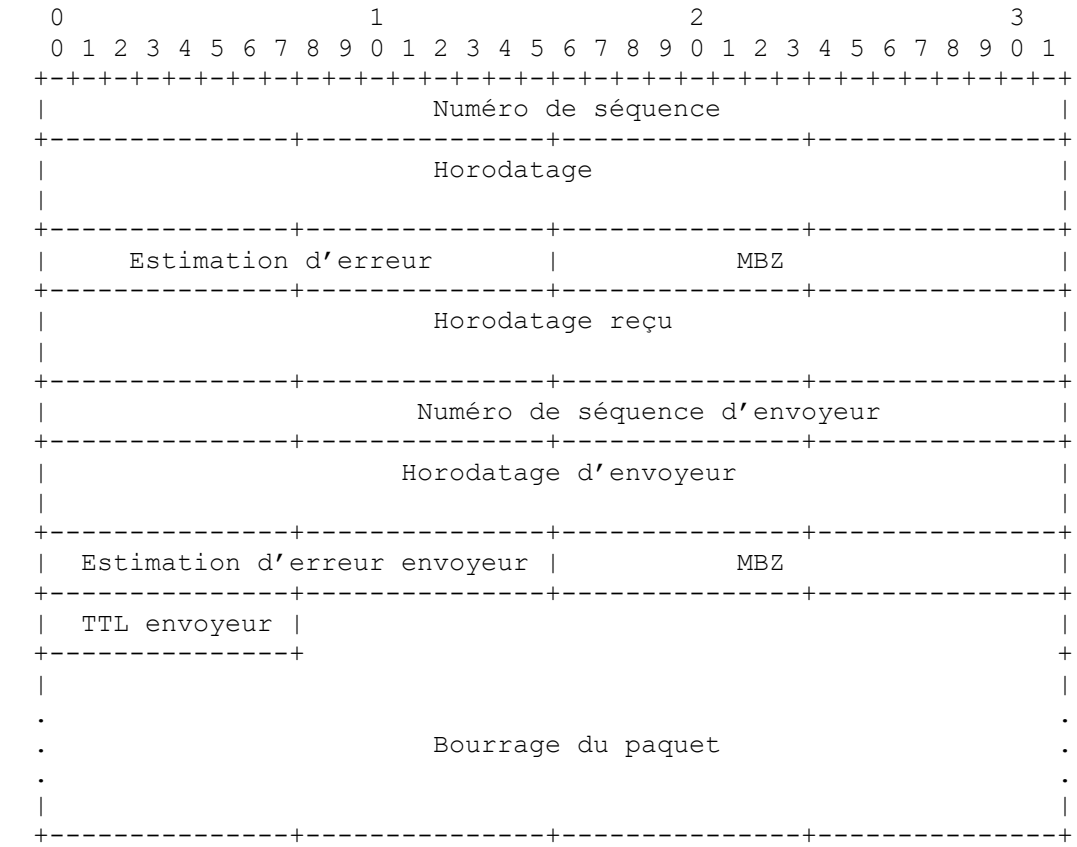
Il existe la possibilité d'une défaillance de l'envoyeur de session durant une session, ou le chemin entre l'envoyeur de session et le réflecteur de session peut avoir une défaillance pendant qu'une session d'essais est en cours. Le réflecteur de session PEUT arrêter toute session qui a été commencée quand aucun paquet associé à cette session n'a été reçu pendant REFWAIT secondes. La valeur par défaut de REFWAIT DEVRA être 900 secondes, et ce temps d'attente PEUT être configurable. Cette temporisation permet au réflecteur de session de libérer les ressources en cas de défaillance.

4.2.1 Format et contenu du paquet TWAMP-Test

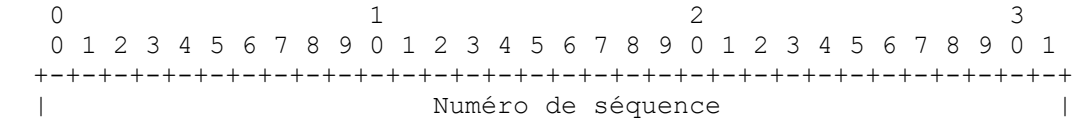
Le réflecteur de session DOIT transmettre un paquet à l'envoyeur de session en réponse à chaque paquet reçu. Le réflecteur de session DEVRAIT transmettre les paquets ausstôt que possible. Le réflecteur de session DEVRAIT régler le TTL dans IPv4 (ou la limite de bonds dans IPv6) dans le paquet UDP à 255.

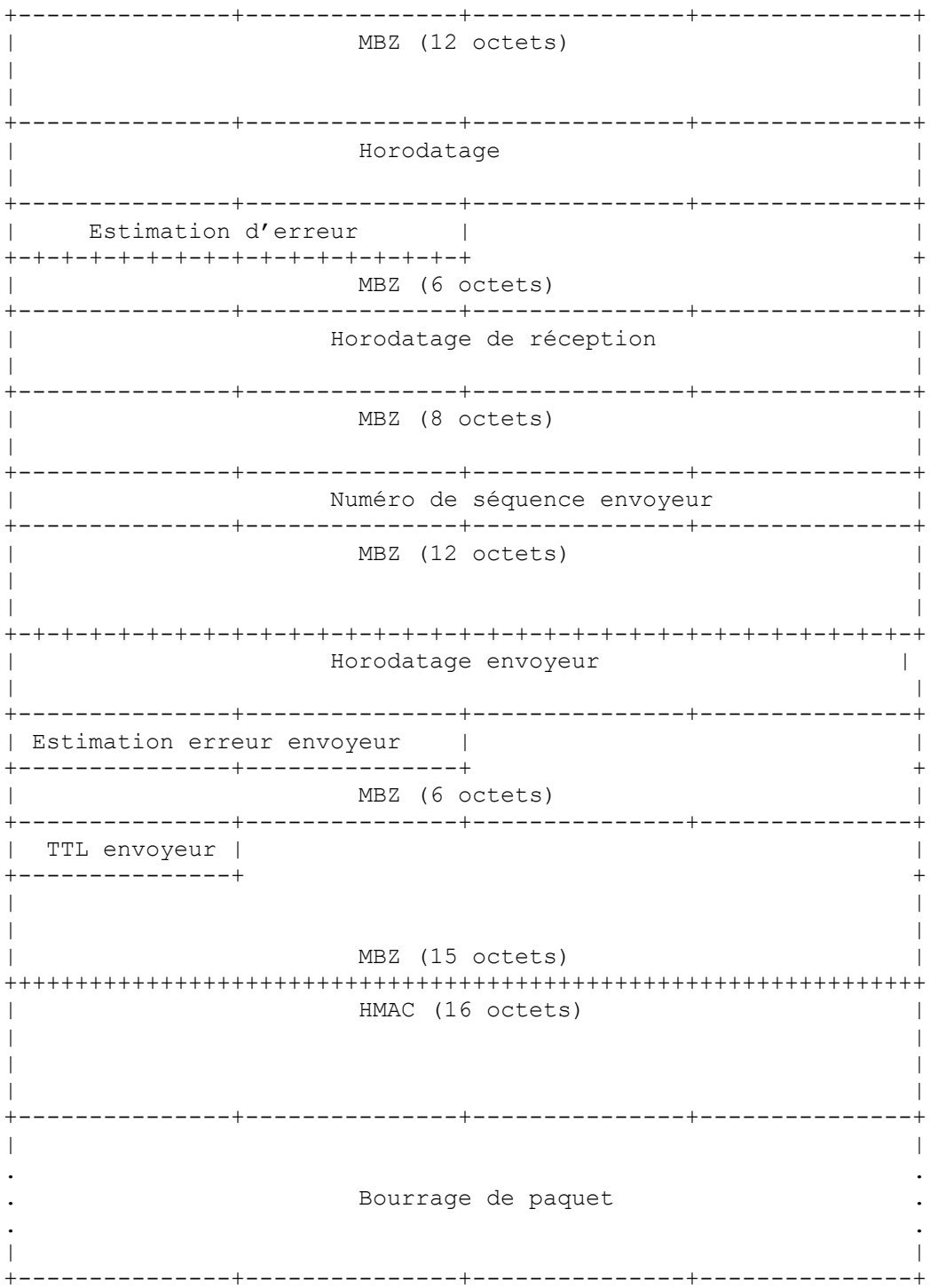
Le paquet d'essai va avoir les informations nécessaires pour le calcul des métriques bidirectionnelles par l'envoyeur de session. Le format du paquet d'essai dépend du mode utilisé. Les deux formats sont présentés ci-dessous.

Pour le mode non authentifié :

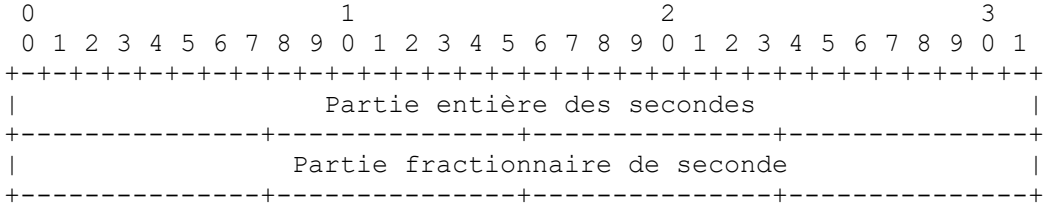


Pour les modes authentifié et chiffré :





Noter que tous les horodatages ont le même format que dans OWAMP [RFC4656] comme suit :



Numéro de séquence est le numéro de séquence du paquet d'essai conformément à son ordre de transmission. Il commence à zéro et est incrémenté de un pour chaque paquet suivant. Le numéro de séquence généré par le réflecteur de session est indépendant du numéro de séquence des paquets arrivants.

Horodatage et Estimation d'erreur sont respectivement l'horodatage de transmission et l'estimation d'erreur du réflecteur de session pour le paquet d'essai reflété. Le format de tous les champs d'horodatage et d'estimation d'erreur suit la définition et les formats définis pour OWAMP, au paragraphe 4.1.2 de la [RFC4656].

Horodatage d'envoyeur et Estimation d'erreur d'envoyeur sont les copies exactes de l'horodatage et de l'estimation d'erreur provenant du paquet d'essai de l'envoyeur de session qui correspond à ce paquet d'essai.

Le TTL d'envoyeur est 255 quand il est transmis par l'envoyeur de session. Le TTL d'envoyeur est réglé à la valeur de durée de vie (ou de limite de bonds) du paquet reçu de l'en-tête de paquet IP quand il est transmis par le réflecteur de session.

L'horodatage de réception est l'heure à laquelle le paquet d'essai a été reçu par le réflecteur. La différence entre Horodatage et Horodatage de réception est le temps pendant lequel le paquet a été en transit dans le réflecteur de session. L'estimation d'erreur associée au champ Horodatage s'applique aussi à l'horodatage de réception.

Le numéro de séquence d'envoyeur est une copie du numéro de séquence du paquet transmis par l'envoyeur de session qui est cause que le réflecteur de session a généré et envoyé ce paquet d'essai.

Le champ HMAC dans les paquets TWAMP-Test couvre les mêmes champs que le chiffrement de la norme de chiffrement évolué (AES, *Advanced Encryption Standard*). Donc, dans le mode authentifié, HMAC couvre le premier bloc (16 octets) ; en mode chiffré, HMAC couvre les six premiers blocs (96 octets). Dans TWAMP-Test, le champ HMAC NE DOIT PAS être chiffré.

Le bourrage de paquet dans TWAMP-Test DEVRAIT être pseudo-aléatoire (il DOIT être généré indépendamment de tous autres nombres pseudo-aléatoires mentionnés dans ce document). Cependant, les mises en œuvre DOIVENT fournir un paramètre de configuration, une option, ou un moyen différent de faire que le bourrage de paquet consiste tout en zéros. Le bourrage de paquet NE DOIT PAS être couvert par le HMAC et NE DOIT PAS être chiffré.

La longueur minimum du segment de données des paquets TWAMP-Test en mode non authentifié est 41 octets, et 104 octets dans les deux modes authentifié et chiffré.

Noter que les formats de paquet d'essai du réflecteur de session sont plus grands que les formats de l'envoyeur. Le réflecteur de session DEVRAIT réduire la longueur du bourrage du paquet de l'envoyeur pour réaliser des longueurs égales de charge utile de paquet IP dans chaque direction de transmission, quand un bourrage suffisant est présent. Le réflecteur de session PEUT réutiliser le bourrage de paquet de l'envoyeur (car les exigences pour la génération du bourrage sont les mêmes pour chacune) et dans ce cas, le réflecteur de session DEVRAIT tronquer le bourrage de façon à ce que les octets de plus fort poids soient éliminés.

En mode non authentifié, NI le chiffrement NI l'authentification NE DOIVENT être appliqués.

La disposition du paquet TWAMP-Test est identique dans les modes authentifié et chiffré. L'opération de chiffrement pour un paquet de l'envoyeur de session suit les mêmes règles d'envoi des paquets de session que définies dans OWAMP au paragraphe 4.1.2 de la [RFC4656].

La principale différence entre le mode authentifié et le mode chiffré est la portion des paquets d'essai qui est couverte par HMAC et chiffrée. Le mode authentifié permet d'aller chercher l'horodatage après qu'une portion du paquet est chiffrée, mais en mode chiffré tous les numéros de séquence et les horodatages sont collectés avant le chiffrement afin de fournir le maximum de protection de l'intégrité des données.

En mode authentifié, seul le numéro de séquence dans le premier bloc est chiffré, et les horodatages et numéros de séquence suivants sont envoyés en clair. L'envoi de l'horodatage en clair permet de réduire le temps entre le moment où un horodatage est obtenu par un réflecteur de session et celui où ce paquet est envoyé. Cela améliore potentiellement la précision de l'horodatage, parce que le numéro de séquence peut être chiffré avant d'aller chercher l'horodatage.

En mode chiffré, le réflecteur DOIT aller chercher les horodatages, générer le HMAC, et chiffrer le paquet, puis l'envoyer.

Les méthodes d'obtention des clés et de chiffrement suivent la même procédure que dans OWAMP comme décrit ci-dessous. Chaque session TWAMP-Test a deux clés, une clé de session AES et une clé de session HMAC, et les clés sont déduites des clés de TWAMP-Control et du SID.

La clé de session AES de TWAMP-Test est obtenue comme suit : la clé de session AES de TWAMP-Control (la même clé de session AES qu'utilisée pour la session TWAMP-Control correspondante) est chiffrée avec l'identifiant de session (SID) de 16 octets, en utilisant un chiffrement AES-ECB d'un seul bloc comme clé. Le résultat est la clé de session AES de TWAMP-Test à utiliser pour chiffrer (et déchiffrer) les paquets de la session TWAMP-Test particulière. Noter que la clé de session AES de TWAMP-Test, la clé de session AES de TWAMP-Control, et le SID sont tous de 16 octets.

La clé de session HMAC TWAMP-Test est obtenue comme suit : la clé de session HMAC de TWAMP-Control (la même clé de session HMAC qu'utilisée pour la session TWAMP-Control correspondante) est chiffrée en utilisant le chaînage de bloc de chiffrement AES (AES-CBC, *Cipher Block Chaining*) avec l'identifiant de session de 16 octets comme clé. C'est a un chiffrement CBC à deux blocs qui est toujours effectué avec IV=0. Noter que la clé de session HMAC TWAMP-Test et la clé de session HMAC de TWAMP-Control comportent 32 octets, tandis que le SID fait 16 octets.

En mode authentifié, le premier bloc (16 octets) de chaque paquet TWAMP-Test est chiffré en utilisant le mode AES de dictionnaire électronique (ECB, *Electronic Codebook*). Ce mode n'implique aucun chaînage, et les paquets perdus, dupliqués, ou déclassés ne causent pas de problème au déchiffrement d'un paquet dans une session TWAMP-Test.

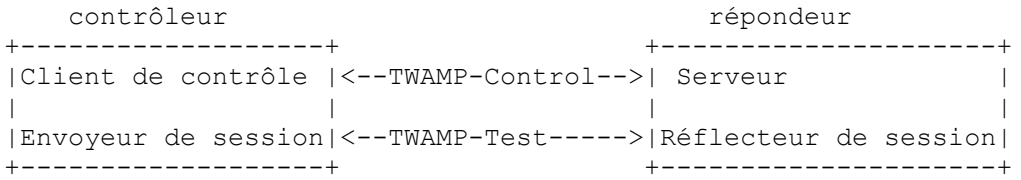
En mode chiffré, les six premiers blocs (96 octets) sont chiffrés en utilisant le mode AES-CBC. La clé de session AES à utiliser est obtenue de la même façon que la clé pour le mode authentifié. Chaque paquet TWAMP-Test est chiffré comme un flux séparé, avec juste une opération de chainage ; le chaînage ne couvre pas plusieurs paquets de sorte que les paquets perdus, dupliqués, ou déclassés ne causent pas de problème. La valeur d'initialisation pour le chiffrement CBC est une valeur dont tous les bits sont à zéro.

Note de mise en œuvre : naturellement, la programmation des clés pour chaque session TWAMP-Test DOIT être établie au plus une fois par session, pas une fois par paquet.

5. Guide de mise en œuvre

Cette section sert de guide pour les mises en œuvre de TWAMP. L'exemple d'architecture présenté ici n'est pas une exigence. Comme pour OWAMP [RFC4656], TWAMP est conçu avec assez de souplesse pour permettre différentes architectures convenant à des exigences de système multiples.

Dans cet exemple, les rôles de client de contrôle et d'envoyeur de session sont mis en œuvre dans un hôte appelé le contrôleur, et les rôles de serveur et de réflecteur de session sont mis en œuvre dans un autre hôte appelé le répondeur.



Cet exemple donne une architecture qui prend en charge toute la norme TWAMP. Le contrôleur établit la session d'essais avec le répondeur par le protocole TWAMP-Control. Après l'établissement de la session, le contrôleur transmet les paquets d'essai au répondeur. Le répondeur suit le comportement de réflecteur de session de TWAMP décrit au paragraphe 4.2.

L'Appendice I donne un exemple à des fins uniquement d'information. Il suggère un chemin incrémentaire pour adopter TWAMP, par qui met en œuvre d'abord le protocole TWAMP-Test.

6. Considérations sur la sécurité

Fondamentalement, TWAMP et OWAMP utilisent le même protocole pour l'établissement des procédures de contrôle et d'essai. La principale différence entre TWAMP et OWAMP est le comportement du réflecteur de session dans TWAMP par rapport au comportement du receveur de session dans OWAMP. Cette différence de comportements n'introduit aucune faiblesse de sécurité connue qui ne soit déjà traitée par les caractéristiques de sécurité de OWAMP. Toutes les considérations sur la sécurité de OWAMP [RFC4656] s'appliquent à TWAMP.

Le message Server-Greeting (définit au paragraphe 3.1 de la [RFC4656] pour OWAMP) inclut un champ Compte pour

spécifier le compte d'itération utilisé dans PKCS n° 5 pour générer les clés à partir des secrets partagés. OWAMP recommande une limite inférieure de 1024 itérations, mais pas de limite supérieure. Le champ Compte donne une opportunité d'attaque de déni de service (DOS) parce qu'il est long de 32 bits. Si un système attaquant règle la valeur maximum dans Compte à 2**32, alors le système attaqué va piétiner pendant un délai significatif en tentant de générer des clés. Donc, les systèmes conformes à TWAMP DEVRAIENT avoir une commande de configuration pour limiter la valeur maximale du champ Compte. La valeur maximale par défaut de Compte DEVRAIT être 32768. Comme suggéré dans OWAMP, cette valeur PEUT être augmentée quand des puissances de calcul supérieures deviendront courantes. Si un client de contrôle reçoit un message Server-Greeting avec un champ Compte supérieur à sa valeur maximum configurée, il DEVRAIT clore la connexion de contrôle.

7. Remerciements

Nous tenons à remercier Nagarjuna Venna, Sharee McNab, Nick Kinraid, Stanislav Shalunov, Matt Zekauskas, Walt Steverson, Jeff Boote, Murtaza Chiba, et Kevin Earnst de leurs commentaires, suggestions, révisions, utiles discussion, et relectures. Lars Eggert, Sam Hartman, et Tim Polk ont contribué à de très utiles relectures au niveau AD, et les auteurs les remercient de leurs contributions à ce mémoire.

8. Considérations relatives à l'IANA

L'IANA a alloué un numéro d'accès TCP bien connu (861) pour la partie OWAMP-Control du protocole OWAMP [RFC4656].

```
+ ...
  owamp-control   861/tcp    OWAMP-Control
  owamp-control   861/udp    OWAMP-Control
  #               [RFC4656]
```

L'IANA a aussi alloué un numéro d'accès TCP/UDP bien connu pour le protocole TWAMP-Control.

```
  ...
  twamp-control   862/tcp    Two-way Active Measurement Protocol (TWAMP) Control
  twamp-control   862/udp    Two-way Active Measurement Protocol (TWAMP) Control
  #               [RFC5357]
  #           863-872    Non alloué
```

Comme TWAMP ajoute une commande Control supplémentaire au delà de la spécification OWAMP-Control et décrit le comportement quand cette commande de contrôle est utilisée, l'IANA a créé un registre pour le champ Numéro de commande TWAMP. Le champ n'est pas nommé explicitement dans la [RFC4656] mais est invoqué pour chaque commande. Ce champ est un mécanisme d'extension reconnu pour TWAMP.

8.1 Spécification de registre

L'IANA a créé un registre Numéro de commande TWAMP-Control. Les commandes TWAMP-Control sont spécifiées par le premier octet dans les messages OWAMP-Control comme le montre le paragraphe 3.5 de la [RFC4656], et modifiées par le présent document. Donc, ce registre peut contenir seize valeurs possibles.

8.2 Gestion de registre

Parce que le registre peut seulement contenir seize valeurs, et parce que OWAMP et TWAMP sont des protocoles de l'IETF, ce registre doit seulement être mis à jour par "Consensus de l'IETF" comme spécifié dans la [RFC5226], RFC qui documente l'utilisation de ce qui est approuvé par l'IESG. On s'attend à ce que de nouvelles valeurs soient allouées comme des entiers à croissance monotone dans la gamme de [0 à 15], sauf sil il y a de bonnes raison de faire autrement.

8.3 Nombres expérimentaux

La [RFC3692] recommande d'allouer un nombre approprié de valeurs pour l'expérimentation et les essais. Les auteurs n'ont pas une idée claire de la quantité de numéros qui pourraient être utiles dans cet espace, ni de si il serait utile qu'on puisse facilement les distinguer ou de les mettre "à la fin" de la gamme des numéros. Deux gammes pourraient être utiles, disons une pour le contrôle de session, et une pour aller chercher la session. Par ailleurs, un seul numéro permettrait une extension illimitée, parce que le format du reste du message pourrait être taillé sur mesure, avec l'allocation des autres numéros faite une fois que leur utilité a été prouvée. Donc, le présent document alloue un numéro (6) comme destiné à l'expérimentation et aux essais.

8.4 Contenu initial du registre

Registre des numéros de commande de TWAMP-Control

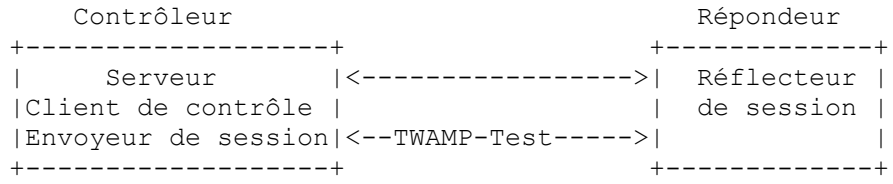
Valeur	Description	Définition sémantique
0	Réservé	
1	Interdit	
2	Débute les sessions	RFC 4656, paragrape 3.7
3	Arrête les sessions	RFC 4656, paragraphe 3.8
4	Réservé	
5	Demande de session TW	ce document, paragraphe 3.5
6	Expérimentation	indéfini, voir le paragraphe 8.3.

9. Considérations d'internationalisation

Le protocole ne porte aucune information en langage naturel, à l'exception éventuelle de KeyID dans TWAMP-Control, qui est codée en UTF-8 [RFC3629], [RFC5198].

Appendice 1. TWAMP léger (information)

Dans cet exemple, les rôles de client de contrôle, serveur, et envoyeur de session sont mis en œuvre dans un hôte appelé le contrôleur, et le rôle de réflecteur de session est mis en œuvre dans un autre hôte appelé le répondeur.



Cet exemple montre une architecture simple pour les répondeurs où leur rôle va être simplement d'agir comme un point d'essai léger dans le réseau. Le contrôleur établit la session d'essais avec le serveur par des moyens non standard. Après l'établissement de la session, le contrôleur transmet les paquets d'essai au répondeur. Le répondeur suit le comportement du réflecteur de session de TWAMP comme décrit au paragraphe 4.2 avec les exceptions suivantes.

Dans le cas de TWAMP léger, le réflecteur de session n'a pas nécessairement connaissance de l'état de session. Si le réflecteur de session n'a pas connaissance de l'état de session, ALORS le réflecteur de session DOIT copier le numéro de séquence du paquet reçu dans le champ Numéro de séquence du paquet reflété. Le contrôleur reçoit les paquets d'essai reflétés et collecte les métriques bidirectionnelles. Cette architecture permet la collecte des métriques bidirectionnelles.

Cet exemple élimine le besoin du protocole TWAMP-Control, et suppose que le réflecteur de session est configuré et communique sa configuration au serveur par des moyens non standard. Le réflecteur de session reflète simplement les paquets entrants au contrôleur tout en copiant les informations nécessaires et en générant les numéros de séquence et valeurs d'horodatage conformément au paragraphe 4.2.1. TWAMP Léger introduit quelques problèmes de sécurité supplémentaires. Les moyens non standard de contrôle du répondeur et d'établissement des sessions d'essai DEVRAIENT offrir les caractéristiques suivantes :

Le protocole de contrôle de répondeur non standard DEVRAIT avoir un mode de fonctionnement authentifié. Le répondeur DEVRAIT être configurable à accepter seulement les sessions de contrôle authentifiées.

Le protocole de contrôle de répondeur non standard DEVRAIT avoir le moyen d'activer les modes authentifié et chiffré du protocole TWAMP-Test.

Quand les sessions d'essai TWAMP Léger opèrent en mode authentifié ou chiffré, le réflecteur de session DOIT avoir un mécanisme pour générer les clés (parce que le protocole TWAMP-Control joue normalement un rôle dans ce processus, mais n'est pas présent ici). La spécification du mécanisme de génération de clés sort du domaine du présent mémoire.

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)

[RFC2474] K. Nichols, S. Blake, F. Baker et D. Black, "Définition du champ Services différenciés (DS) dans les en-têtes IPv4 et IPv6", décembre 1998. *(P.S. ; MàJ par* RFC3168, RFC3260, RFC8436)

[RFC2681] G. Almes, S. Kalidindi, M. Zekauskas, "Métrique de délai d'aller-retour pour IPPM", septembre 1999. *(P.S.)*

[RFC3629] F. Yergeau, "UTF-8, un format de transformation de la norme ISO 10646", STD 63, novembre 2003.

[RFC4656] S. Shalunov et autres, "Protocole de mesures actives unidirectionnelles (OWAMP)", septembre 2006. *(P.S.)*

[RFC5198] J. Klensin, M. Padlipsky, "Format Unicode pour les échanges sur le réseau", mars 2008. (*Remplace* RFC0698, *MàJ* RFC0854) *(P.S.)*

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

Référence pour information

[RFC3692] T. Narten, "L'allocation de numéros expérimentaux et d'essai est considérée comme utile", janvier 2004. (BCP0082)

Adresse des auteurs

Kaynam Hedayat
Brix Networks
285 Mill Road
Chelmsford, MA 01824
USA
mél : khedayat@brixnet.com
URI : http://www.brixnet.com/

Roman M. Krzanowski
Verizon
500 Westchester Ave.
White Plains, NY
USA
mél : roman.krzanowski@verizon.com
URI : http://www.verizon.com/

Al Morton
AT&T Labs
Room D3 - 3C06
200 Laurel Ave. South
Middletown, NJ 07748 USA
mél : acmorton@att.com
URI : http://home.comcast.net/~acmacm/

Kiho Yum
Juniper Networks
1194 Mathilda Ave.
Sunnyvale, CA
USA
mél : kyum@juniper.net
URI : http://www.juniper.com/

Jozef Z. Babiarz
Nortel Networks
3500 Carling Avenue
Ottawa, Ont K2H 8E9
Canada
mél : babiarz@nortel.com
URI : http://www.nortel.com/

Déclaration complète de droits de reproduction

Remerciement
Le financement de la fonction d'édition des RFC est fourni par l'activité de soutien administratif de l'IETF (IASA).