

Groupe de travail Réseau
Request for Comments : 4737
 Catégorie : Sur la voie de la normalisation

A. Morton, AT&T Labs
 L. Ciavattone, AT&T Labs
 G. Ramachandran, AT&T Labs
 S. Shalunov, Internet2
 J. Perser, Veriwave
 novembre 2006

Traduction Claude Brière de L'Isle

Métriques de réarrangement de paquets

Statut du présent mémoire

Le présent document spécifie un protocole de l'Internet en cours de normalisation pour la communauté de l'Internet, et appelle à des discussions et suggestions pour son amélioration. Prière de se référer à l'édition en cours des "Protocoles officiels de l'Internet" (STD 1) pour voir l'état de normalisation et le statut de ce protocole. La distribution du présent mémoire n'est soumise à aucune restriction.

Notice de copyright

Copyright (C) The IETF Trust (2006).

Résumé

Le présent mémoire définit les métriques pour évaluer si un réseau a conservé l'ordre des paquets paquet par paquet. Il donne les motivations des nouvelles métriques et discute les problèmes de mesures, incluant les informations de contexte requises pour toutes les métriques. Le mémoire définit d'abord un singleton réordonné, et ensuite l'utilise comme base de métriques d'échantillons pour quantifier l'étendue des réarrangement dans plusieurs dimensions utiles pour la caractérisation des réseaux ou la conception d'un receveur. Des métriques supplémentaires quantifient la fréquence de réarrangement et la distance entre des occurrences distinctes. On définit ensuite une métrique orientée vers l'assertion des effets du réarrangement sur TCP. Plusieurs exemples d'évaluation utilisant les diverses métriques d'échantillon sont inclus. Un appendice donne des définitions pour évaluer l'ordre avec la fragmentation de paquet.

Table des matières

1. Introduction.....	2
1.1 Motivation.....	2
1.2 Buts et objectifs.....	3
1.3 Contexte exigé pour toute métrique de réarrangement.....	4
2. Conventions utilisée dans le document.....	4
3. Métrique de singleton de paquet réarrangé.....	4
3.1 Nom de métrique.....	4
3.2 Paramètres de métrique.....	5
3.3 Définitions.....	5
3.4 Définition de discontinuité de séquence.....	5
3.5 Évaluation du réarrangement en termes de temps ou d'octets.....	6
3.6 Discussion.....	6
4. Métriques d'échantillons.....	6
4.1 Ratio de paquets réarrangés.....	7
4.1.1 Nom de métrique.....	7
4.1.3 Définition.....	7
4.2 Étendue du réarrangement.....	7
4.3 Décalage de réarrangement de tardiveté.....	8
4.4 Décalage d'octet de réarrangement.....	9
4.5 Trous entre plusieurs discontinuités de réarrangement.....	10
4.6 Cours sans réarrangement.....	11
5. Métriques centrées sur l'assertion du receveur : métrique pertinente pour TCP.....	13
5.1 Nom de métrique.....	13
5.2 Notation des paramètres.....	13
5.3 Définitions.....	13
5.4 Discussion.....	14
6. Problèmes de mesure et de mise en œuvre.....	14
6.1 Considérations de mesure passive.....	16

7. Exemples d'évaluation d'ordre d'arrivée.....	16
7.1 Exemple avec un seul paquet réarrangé.....	16
7.2 Exemple avec deux paquets réarrangés.....	17
7.3 Exemple avec trois paquets réarrangés.....	18
7.4 Exemple avec plusieurs discontinuités de réarrangement de paquet.....	19
8. Considérations sur la sécurité.....	19
8.1 Attaques de déni de service.....	19
8.2 Confidentialité des données d'utilisateur.....	20
8.3 Interférence avec la métrique.....	20
9. Considérations relatives à l'IANA.....	20
10. Références normatives.....	22
11. Références pour information.....	22
12. Remerciements.....	23
Appendice A. Exemple de mise en œuvre en C (information).....	23
Appendice B. Évaluation de l'ordre des fragments (information).....	25
B.1 Nom de métrique.....	26
B.2 Paramètres de métrique supplémentaires.....	26
B.3 Définition.....	26
B.4 Discussion : Notes sur les métriques d'échantillon pour évaluer les fragments.....	27
Appendice C. Responsabilité et licence.....	26
Adresse des auteurs.....	27
Déclaration complète de droits de reproduction.....	27

1. Introduction

L'arrivée dans l'ordre est une propriété qu'on trouve dans les paquets qui ont un chemin en transit, où le numéro de séquence du paquet augmente à chaque nouvelle arrivée et où il n'y a pas d'étape vers l'arrière. Le protocole Internet [RFC0791], [RFC2460], n'a pas de mécanisme pour assurer la livraison ou le séquençage de paquet, et des protocoles de couche supérieure (au-dessus de IP) devraient être prêts à traiter de la perte et du réarrangement. Le présent mémoire définit les métriques de réarrangement.

Un identifiant de séquence unique porté dans chaque paquet, comme un numéro de message entier incrémenté de façon consécutive établit la séquence de source.

La détection du réarrangement à la destination se fonde sur l'ordre d'arrivée des paquets en comparaison avec une valeur de référence non inversable [Cia03].

Cette métrique est cohérente avec la [RFC2330] et classe les paquets arrivant avec des numéros de séquence plus petits que leurs prédécesseurs comme déclassés ou réordonnés. Par exemple, si des paquets numérotés à la suite arrivent comme 1,2,4,5,3, le paquet 3 est alors réordonné. Ceci est équivalent à la définition du réarrangement de Paxson dans [Pax98], où les paquets "tardifs" sont déclarés réordonnés. L'alternative est de désigner des paquets comme "prématurés" (les paquets 4 et 5 dans l'exemple) mais seule l'arrivée du paquet 3 distingue cette circonstance de la perte de paquet. Concentrer son attention sur les paquets tardifs permet de conserver l'orthogonalité avec la métrique de perte de paquets. La construction de la métrique est très similaire à la validation de l'espace de séquence pour les segments reçus dans la [RFC0793]. Les travaux antérieurs pour définir la livraison dans l'ordre incluent [Cia00], [Ben99], [Lou01], [Bel02], [Jai02], et [Cia03].

1.1 Motivation

Une métrique de réarrangement est pertinente pour la plupart des applications, en particulier quand on veut vérifier la prise en charge de flux de supports en temps réel par un réseau. L'étendue du réarrangement peut être suffisante pour causer l'élimination d'un paquet reçu par des fonctions au dessus de la couche IP.

L'ordre des paquets peut changer durant le transfert, et plusieurs caractéristiques de chemin spécifiques peuvent rendre le réarrangement plus probable.

Des exemples en sont :

* Quand deux chemins (ou plus) avec des temps de transfert légèrement différents prennent en charge un seul flux de paquets, les paquets qui traversent le ou les plus longs chemins peuvent arriver déclassés. Plusieurs chemins peuvent

être utilisés pour réaliser l'équilibrage de charge ou peuvent découler de l'instabilité d'acheminement.

- * Pour augmenter la capacité, un appareil du réseau conçu avec de multiples processeurs desservant un seul accès (ou des liaisons parallèles) peuvent réarranger comme effet collatéral.
- * Un protocole de retransmission de couche 2 qui compense une liaison encline à l'erreur peut causer le réarrangement des paquets.
- * Si pour une raison ou une autre, les paquets dans une mémoire tampon ne sont pas sortis dans leur ordre d'arrivée, leur ordre va changer.
- * Si les paquets d'un flux sont alloués à plusieurs mémoires tampons (par exemple, suite à l'évaluation des caractéristiques du trafic) et si les mémoires tampons ont des niveaux d'occupation et/ou taux de service différents, leur ordre va probablement changer.

Quand une ou plusieurs des caractéristiques de chemin ci-dessus sont présentes de façon continue, le réarrangement peut être présent en permanence. La condition permanente de réarrangement cause normalement le réarrangement d'une fraction appréciable des paquets. Cette forme de réarrangement est très facilement détectée en réduisant l'espacement entre les paquets d'essai. Un réarrangement transitoire peut se produire en réponse à l'instabilité du réseau ; des boucles d'acheminement temporaires peuvent causer des périodes de réarrangement extrême. Cette condition est caractérisée par de longs flux en ordre avec des instances occasionnelles de réarrangement, parfois avec une extrême corrélation. Cependant, on n'envisage pas une livraison de paquet dans un ordre complètement aléatoire, où, par exemple, le dernier ou le premier paquet d'un échantillon aient une égale probabilité d'arriver le premier à la destination. Donc, on s'attend au moins à un degré minimal d'ordre dans les arrivées de paquets, comme cela se produit dans les réseaux réels.

La capacité de restaurer l'ordre à la destination va probablement avoir des limites. Les hôtes ont en pratique des mémoires tampons de réception d'une taille finie en termes de paquets, d'octets, ou de temps (comme des mémoires tampons de gigue). Une fois effectuée la détermination initiale du réarrangement, il est utile de quantifier son extension, ou sa tardiveté, dans tous les dimensions significatives.

1.2 Buts et objectifs

Les définitions ci-dessous visent à satisfaire les buts suivants :

1. Déterminer si un réarrangement de paquets a eu lieu ou non.
2. Quantifier le degré de réarrangement. (On définit un certain nombre de métriques pour atteindre ce but, parce que les procédures de réception diffèrent selon les protocoles ou les applications. Comme les effets du réarrangement de paquets varient selon ces procédures, une métrique qui quantifie un aspect clé du comportement d'un receveur pourrait être non pertinente pour un receveur différent. Si toutes les métriques définies ci-dessous sont rapportées, elles donnent une vue panoramique des conditions de réarrangement.)

Les métriques de réarrangement DOIVENT :

- + avoir une ou plusieurs applications, comme une conception de receveur ou une caractérisation de réseau, et une pertinence obligatoire du point de vue de la communauté intéressée ;
- + être calculable "au vol" ;
- + fonctionner même si le flux a des paquets dupliqués ou perdus.

Il est souhaitable que les métriques de réarrangement aient un ou plusieurs des attributs suivants :

- + la capacité d'enchaîner les résultats pour les segments mesurés séparément pour estimer le réarrangement d'un chemin entier ;
- + la simplicité pour une utilisation et compréhension faciles ;
- + la pertinence pour la conception de TCP ;
- + la pertinence pour les performances des applications en temps réel.

Le jeu actuel des métriques satisfait à toutes les exigences ci-dessus et fournit tous les attributs sauf celui d'enchaînement (excepté dans le cas où les mesures des segments de chemin ne montrent pas de réarrangement, et qu'on peut estimer que le chemin complet composé de ces segments ne présente pas non plus de réarrangement). Cependant, satisfaire ces objectifs restreint l'ensemble de métriques à celles qui fournissent des indications claires de la caractérisation du réseau ou conception de receveur. Elles ne sont probablement pas exhaustives dans la couverture des effets du réarrangement sur les

applications, et des mesures supplémentaires sont (peut-être) possibles.

1.3 Contexte exigé pour toute métrique de réarrangement

Un aspect critique de toute métrique du réarrangement est le lien inséparable avec les conditions de mesure. Le réarrangement de paquet n'est pas bien défini si le contexte de mesure complet n'est pas rapporté. Donc, toutes les définitions de métrique de réarrangement comportent les paramètres suivants :

1. Les identifiants "Paquet de type-P" [RFC2330] pour le flux de paquets, incluant les adresses de transport pour la source et la destination, et toute autre information qui peut résulter en un traitement différent du paquet.
2. Le jeu de paramètres de flux pour la discipline d'envoi, comme les paramètres uniques de flux périodiques (comme dans la [RFC3432]) les flux de style TCP (comme dans la [RFC3148]) ou les flux de Poisson (comme dans la [RFC2330]). Les paramètres de flux incluent la taille de paquet, spécifiée comme valeur fixe ou comme schéma de tailles (comme applicable).

Chaque fois qu'une métrique est rapportée, elle DOIT inclure une description de ces paramètres pour fournir le contexte des résultats.

2. Conventions utilisée dans le document

Les mots clés "DOIT", "NE DOIT PAS", "EXIGE", "DEVRA", "NE DEVRA PAS", "DEVRAIT", "NE DEVRAIT PAS", "RECOMMANDÉ", "PEUT", et "FACULTATIF" dans ce document sont à interpréter comme décrit dans la [RFC2119]. Bien que la RFC 2119 ait été écrite pour les protocoles, les mots clés sont utilisés dans le présent document pour des raisons similaires. Ils sont utilisés pour assurer que les résultats des mesures provenant de deux mises en œuvre différentes sont comparables, et pour noter les instances où une mise en œuvre pourrait perturber le réseau.

Dans le présent mémoire, les caractères " \leq " devrait être lus comme "inférieur ou égal à" et " \geq " comme "supérieur ou égal à".

3. Métrique de singleton de paquet réarrangé

Le cadre IPPM [RFC2330] décrit les notions de singleton, d'échantillon, et de statistiques. Pour référence :

"Par métrique de "singleton", on se réfère à des métriques qui sont, en un sens, atomiques. Par exemple, une seule instance de "capacité de débit brut" d'un hôte à un autre pourrait être définie comme un métrique de singleton, même si l'instance implique de mesurer les horaires d'un certain nombre de paquets Internet."

L'évaluation de l'ordre des paquets exige le support de plusieurs concepts. Le premier est un algorithme (fonction) qui produit une série d'identifiants à croissance strictement monotone, appliqués aux paquets à la source pour établir de façon univoque l'ordre de transmission des paquets (où une fonction, $g(x)$, est à croissance strictement monotone si pour tout $x > y$, $g(x) > g(y)$). L'identifiant de séquence univoque peut simplement être un numéro de message entier s'incrémentant consécutivement, ou un numéro de séquence comme utilisé ci-dessous. La notion de roulement de numéro de séquence est discutée à la Section 6.

Le second concept à prendre en charge est une valeur mémorisée qui est le "prochain numéro de paquet attendu". Dans des conditions normales, la valeur de prochain attendu (NextExp, *Next Expected*) est le numéro de séquence du paquet précédent plus 1 pour la numérotation de message. (En général, le receveur reproduit l'algorithme de l'expéditeur et la séquence des identifiants afin que le "prochain attendu" puisse être déterminé.)

Chaque paquet dans un flux de paquets peut être évalué avec cette métrique de singleton d'ordre.

3.1 Nom de métrique

Type-P-Reordered

3.2 Paramètres de métrique

Src : adresse IP de l'hôte émetteur.

Dst : adresse IP de l'hôte de destination.

SrcTime : heure d'émission d'un paquet à la source (ou heure du réseau).

s : numéro de séquence univoque de paquet appliqué à la source, en unités de messages.

NextExp : prochain numéro de séquence attendu à la destination, en unités de messages. La valeur mémorisée dans NextExp est déterminée à partir d'un paquet arrivé précédemment.

Et facultativement :

PayloadSize : nombre d'octets contenus dans le champ Informations et mentionné quand la séquence SrcByte est fondée sur les octets transférés.

SrcByte : numéro de séquence du paquet appliqué à la source, en unités d'octets de charge utile.

3.3 Définitions

Si un paquet s (envoyé à l'instant SrcTime) est trouvé réordonné par comparaison avec la valeur NextExp, son Type-P-Reordered = VRAI ; autrement, Type-P-Reordered = FAUX, comme défini ci-dessous :

La valeur de Type-P-Reordered est définie comme VRAI si $s < \text{NextExp}$ (le paquet est réordonné). Dans ce cas, la valeur NextExp ne change pas.

La valeur de Type-P-Reordered est définie comme FAUX si $s \geq \text{extExp}$ (le paquet est dans l'ordre). Dans ce cas, NextExp est réglé à $s + 1$ pour sa comparaison avec le prochain paquet à arriver.

Comme la valeur NextExp ne peut pas décroître, elle donne un critère d'ordre non réversible pour identifier les paquets réordonnés.

Cette définition peut aussi être spécifiée en pseudo-code.

À l'arrivée avec succès d'un paquet de numéro de séquence s :

```

si  $s \geq \text{NextExp}$  alors                                /* s est dans l'ordre */
    NextExp =  $s + 1$  ;
    Type-P-Reordered = Faux ;
autrement                                             /* quand  $s < \text{NextExp}$  */
    Type-P-Reordered = Vrai

```

3.4 Définition de discontinuité de séquence

Les paquets avec $s > \text{NextExp}$ sont un cas particulier de livraison dans l'ordre. Cette condition indique une discontinuité de séquence, parce que il y a perte ou réarrangement de paquet. Les paquets réordonnés doivent arriver pour que la discontinuité de séquence soit définie comme discontinuité de réarrangement (voir la Section 4).

On définit deux états différents pour les paquets dans l'ordre.

Quand $s = \text{NextExp}$, la séquence d'origine a été conservée, et aucune discontinuité n'est présente.

Quand $s > \text{NextExp}$, certains paquets de la séquence originale ne sont pas encore arrivés, et il y a une discontinuité de séquence associée au paquet s. La taille de la discontinuité est $s - \text{NextExp}$, égale au nombre de paquets présentement manquants, réordonnés ou perdus.

En pseudo-code :

À l'arrivée réussie d'un paquet de numéro de séquence s :

```

si  $s \geq \text{NextExp}$ , alors                                /* s est dans l'ordre */
    si  $s > \text{NextExp}$  alors
        SequenceDiscontinuity = Vrai ;
        SeqDiscontinuitySize =  $s - \text{NextExp}$  ;

```

```

autrement
    SequenceDiscontinuity = Faux ;
NextExp = s + 1 ;
Type-P-Reordered = Faux ;
autrement
    Type-P-Reordered = Vrai;
    SequenceDiscontinuity = Faux ;
/* quand s < NextExp */

```

Qu'une discontinuité de séquence se produise (et sa taille) est déterminé par les conditions qui causent la perte et/ou le réarrangement le long du chemin de mesure. Noter qu'un paquet pourrait être réordonné à un point et ensuite être perdu ailleurs sur le chemin, mais cela ne peut pas être connu par les observations à la destination.

3.5 Évaluation du réarrangement en termes de temps ou d'octets

Il est possible d'utiliser d'autres dimensions de temps ou d'octets de charge utile pour éprouver le réarrangement dans la définition du paragraphe 3.3, car SrcTimes et SrcBytes sont uniques et fiables. Les discontinuités de séquence sont facilement définies et détectées avec la numérotation des messages ; cependant, ce n'est pas si simple dans les dimensions de temps ou d'octets. C'est un inconvénient des autres dimensions parce que la définition de la discontinuité de séquence joue un rôle clé dans les métriques d'échantillon qui suivent.

Il est possible de détecter les discontinuités de séquence par la numérotation des octets de charge utile, mais seulement quand l'appareil d'essais connaît exactement la valeur à allouer à NextExp en réponse à toute arrivée de paquet. C'est possible quand le schéma complet de tailles de charge utile est mémorisé à la destination, ou si le schéma de tailles peut être généré en utilisant un générateur de nombres pseudo aléatoires et un germe partagé. Si la taille de charge utile est constante, la numérotation des octets ajoute une complexité inutile à la numérotation de messages.

Il peut être possible de détecter les discontinuités de séquence avec des flux périodiques et la numérotation de l'heure de source, mais il y a des problèmes pratiques à l'envoi exactement comme programmé avec la fiabilité d'horloge.

Les dimensions de temps et d'octets restent une base importante de la caractérisation de l'étendue du réarrangement, comme décrit aux paragraphes 4.3 et 4.4.

3.6 Discussion

Tout paquet arrivant qui porte un numéro de séquence provenant de la séquence qui établit la valeur NextExp peut être évalué pour déterminer si il est dans l'ordre ou réordonné, sur la base de l'arrivée d'un paquet précédent. Dans le cas où NextExp est indéfini (parce que le paquet arrivant est le premier transfert réussi) le paquet est désigné comme étant dans l'ordre (Type-P-Reordered = FAUX).

Cette métrique suppose le réassemblage des fragments de paquet avant l'évaluation. En principe, il est possible d'utiliser la métrique Type-P-Reordered pour évaluer le réarrangement parmi les fragments de paquet, mais chaque fragment doit contenir des informations de séquence de source. Voir les détails à l'Appendice B, "Évaluation d'ordre des fragments".

Si des paquets dupliqués (plusieurs copies non corrompues) arrivent à destination, ils DOIVENT être notés, et seul le premier arrivé est considéré pour la suite de l'analyse (les copies devraient être déclarées réarrangées en accord avec la définition ci-dessus). Cette exigence a les mêmes implications de mémorisation que les métriques IPPM antérieures et suit le précédent de la [RFC2679]. On donne à la Section 6 "Problèmes de mesure et de mise en œuvre" des suggestions pour minimiser la taille de mémorisation nécessaire.

4. Métriques d'échantillons

Dans cette Section, on définit les métriques applicables à un échantillon de paquets provenant d'un seul système de source de numéros de séquence. Quand un réarrangement se produit, il est très souhaitable de vérifier le degré auquel un paquet est déclassé ou réarrangé par rapport aux autres paquets. Cette Section définit plusieurs métriques qui quantifient l'étendue du réarrangement dans diverses unités de mesure. Chaque métrique met en lumière une utilisation pertinente.

Les métriques dans les paragraphes qui suivent ont une orientation de caractérisation de réseau, mais aussi une pertinence pour la conception du receveur lorsque la compensation du réarrangement présente un intérêt. On commence par une

simple métrique de ratio qui indique la portion réarrangée de l'échantillon.

4.1 Ratio de paquets réarrangés

4.1.1 Nom de métrique

Type-P-Reordered-Ratio-Stream

4.1.2 Paramètres de métrique

L'ensemble de paramètres inclut des paramètres singleton Type-P-Reordered, des paramètres uniques de flux de Poisson (comme dans la [RFC2330]) de flux périodiques (comme dans la [RFC3432]) ou de flux de style TCP (comme dans la [RFC3148]) des paramètres de taille de paquet ou des schémas de taille, et ceux qui suivent :

T0 : une heure de début

Tf : une heure de fin

dT : un temps d'attente pour l'arrivée de chaque paquet, en secondes

K : le nombre total de paquets dans le flux envoyé de la source à la destination

L : le nombre total de paquets reçus (arrivant entre T0 et Tf+dT) sur les K paquets envoyés. On rappelle que les copies identiques (dupliquées) ont été retirées, donc $L \leq K$.

R : le ratio de paquets réarrangés sur les paquets reçus, défini ci-dessous.

Noter que le paramètre dT est effectivement le seuil pour déclarer un paquet comme perdu. La métrique de perte de paquet IPPM [RFC2680] refuse de recommander une valeur pour ce seuil, disant plutôt qu'une "bonne ingénierie, incluant la compréhension de la durée de vie des paquets, sera nécessaire en pratique."

4.1.3 Définition

Étant donné un flux de paquets envoyé d'une source à une destination, le ratio de paquets réarrangés dans l'échantillon est

$$R = (\text{compte de paquets avec Type-P-Reordered} = \text{VRAI}) / (L)$$

Cette fraction peut être exprimée par un pourcentage (multiplier par 100). Noter que dans le cas de paquets dupliqués, seule la première copie est utilisée.

4.1.4 Discussion

Quand Type-P-Reordered-Ratio-Stream est zéro, aucune autre métrique de réarrangement n'a besoin d'être examinée pour cet échantillon. Donc, la valeur de cette métrique est sa simple capacité d'additionner les résultats pour un échantillon sans réarrangement.

4.2 Étendue du réarrangement

Ce paragraphe définit la mesure dans laquelle les paquets sont réordonnés et associe une discontinuité de séquence spécifique à chaque paquet réordonné. Elle utilise les paramètres définis ci-dessus.

4.2.1 Nom de métrique

Type-P-Packet-Reordering-Extent-Stream

4.2.2 Notation et paramètres de métrique

On rappelle que K est le nombre de paquets dans le flux à la source, et L est le nombre de paquets reçus à la destination.

Chaque paquet a reçu un numéro de séquence s, un entier consécutif de 1 à K dans l'ordre transmission des paquets (à la source).

Soit $s[1]$, $s[2]$, ..., $s[L]$ qui représentent les numéros de séquence d'origine associés aux paquets dans l'ordre d'arrivée.

$s[i]$ peut être vu comme un vecteur, où l'indice i est la position d'arrivée du paquet avec le numéro de séquence s . En théorie, tout numéro de séquence de source pourrait apparaître dans toute position d'arrivée, mais ceci est improbable en réalité.

Considérons un paquet réordonné (Type-P-Reordered=VRAI) avec l'indice d'arrivée i et le numéro de séquence de source $s[i]$. Il existe un ensemble d'indices j ($1 \leq j < i$) tels que $s[j] > s[i]$.

Les nouveaux paramètres sont :

i : indice de position d'arrivée, où $i-1$ représente une arrivée plus tôt que i .

j : ensemble d'un ou plusieurs indices d'arrivée, où $1 \leq j < i$.

$s[i]$: numéros de séquence d'origine, s , dans l'ordre d'arrivée.

e : étendue du réarrangement, en unités de paquets, définie ci-dessous.

4.2.3 Définition

L'étendue du réarrangement, e , de paquet $s[i]$ est définie comme étant $i-j$ pour la plus petite valeur de j où $s[j] > s[i]$.

De façon informelle, l'étendue du réarrangement est la distance maximum, en paquets, d'un paquet réordonné au premier paquet reçu avec un plus grand numéro de séquence. Si un paquet est dans l'ordre, son étendue de réarrangement est indéfinie. Le premier paquet à arriver est dans l'ordre par définition et a une étendue de réarrangement indéfinie.

Commentaire sur la définition de étendue : pour certains ordres d'arrivée, l'allocation d'une simple position/distance comme étendue du réarrangement tend à surestimer la mémorisation de receveur nécessaire pour restaurer l'ordre. Une procédure plus précise et complexe pour calculer la mémorisation de paquets serait de soustraire tous les paquets réordonnés antérieurement que le receveur pourrait passer aux couches supérieures (voir la métrique de décalage d'octet). Ayant compris ce biais, cette définition est réputée suffisante, en particulier pour ceux qui demandent un calcul "au vol".

4.2.4 Discussion

Le paquet qui a l'indice j ($s[j]$, identifié dans la définition ci-dessus) est la discontinuité de réarrangement associée au paquet s à l'indice i ($s[i]$). Cette définition est formalisée ci-dessous.

Noter que les K paquets dans le flux pourraient être un sous ensemble d'un flux plus grand, mais L est quand même le nombre total de paquets reçus parmi les K paquets envoyés dans ce sous ensemble.

Si un receveur a l'intention de restaurer l'ordre, sa capacité de mémoire tampon détermine alors sa capacité de traiter les paquets qui sont réordonnés. Pour les cas avec un seul paquet réordonné, l'étendue e donne le nombre de paquets qui doivent être retenus dans la mémoire tampon du receveur en attendant que le paquet réordonné complète la séquence. Pour des scénarios plus complexes, l'étendue peut être une surestimation de la mémorisation requise (voir au paragraphe 4.4 "Décalage d'octet de réarrangement" et les exemples de la Section 7). Aussi, si le receveur purge sa mémoire tampon pour une raison quelconque, la métrique d'étendue ne va pas refléter ce comportement, en supposant plutôt que le receveur va tenter une restauration exhaustive de l'ordre.

Bien que l'étendue de réarrangement quantifie principalement le décalage en termes de position d'arrivée, elle peut aussi être utile pour déterminer la portion de paquets réordonnés qui peut ou non être restaurée dans l'ordre dans une mémoire tampon normale de receveur sur la base de leur ordre d'arrivée seul (et sans l'aide de retransmissions).

Une étendue de réarrangement d'échantillons peut être exprimée comme un histogramme pour résumer facilement la fréquence des diverses étendues.

4.3 Décalage de réarrangement de tardiveté

Les paquets réordonnés peuvent recevoir des valeurs de décalage indiquant leur tardiveté en termes de temps de mémoire tampon qu'un receveur doit posséder pour les accommoder. Les métriques de décalage ne sont calculées que sur les paquets réordonnés, comme identifié par la métrique de singleton de paquet réordonné de la Section 3.

4.3.1 Nom de métrique

Type-P-Packet-Late-Time-Stream

4.3.2 Paramètres de métrique

En plus des paramètres définis pour Type-P-Reordered-Ratio-Stream, on spécifie :

DstTime : heure à laquelle chaque paquet du flux arrive à destination, et peut être associée à l'indice i , ou au paquet $s[i]$.

LateTime($s[i]$) : décalage du paquet $s[i]$ en unités de secondes, défini ci-dessous.

4.3.3 Définition

La tardiveté en temps est calculée en utilisant les heures à la destination. Quand le paquet reçu $s[i]$ est réordonné et a une étendue de réarrangement e , alors :

$$\text{LateTime}(s[i]) = \text{DstTime}(i) - \text{DstTime}(i - e)$$

Autrement, en utilisant une notation similaire à celle du paragraphe 4.2, une définition équivalente est :

$$\text{LateTime}(s[i]) = \text{DstTime}(i) - \text{DstTime}(j), \text{ pour } \min\{j \mid 1 \leq j < i\} \text{ qui satisfait } s[j] > s[i].$$

4.3.4 Discussion

Les métriques de décalage peuvent aider à prédire si des paquets réordonnés vont être utiles à un système général de mémoire tampon de receveur avec des limites finies. La limite peut être l'heure de mémorisation avant un instant d'exécution cyclique (comme avec les mémoires tampons de gigue).

Noter que la variation de délai de paquet IP (IPDV, *IP Packet Delay Variation*) [RFC3393] unidirectionnelle donne la variation de délai pour un paquet par rapport au paquet précédent dans la séquence de source. La tardiveté et l'IPDV donnent une indication de si une mémoire tampon à la destination a une mémorisation suffisante pour s'accommoder du comportement du réseau et restaurer l'ordre. Quand un paquet antérieur dans la séquence de source est perdu, l'IPDV va nécessairement être indéfini pour les paquets adjacents, et LateTime peut fournir le seul moyen d'évaluer l'utilité d'un paquet.

Dans le cas de mémoires tampons de gigue, il y a des circonstances où le receveur emploie la dissimulation de perte au moment prévu pour l'exécution d'un paquet en retard. Cependant, si ce paquet arrive décalé, le paramètre "Late Time" détermine si le paquet est encore utile. IPDV ne s'applique plus, parce que le receveur établit un nouveau programme d'exécution avec un délai supplémentaire de mémoire tampon pour s'accommoder d'événements similaires dans le futur (cela n'exige qu'un traitement très minimal).

La combinaison de pertes et de réarrangement influence la métrique de LateTime. Si elle est présentée avec la séquence d'arrivées 1, 10, 5 (où les paquets 2, 3, 4, et 6 à 9 sont perdus) LateTime ne va pas indiquer exactement de combien le paquet 5 est "en retard" par rapport à sa position d'arrivée prévue. IPDV [RFC3393] ne va pas non plus rendre cela, à cause de l'absence des paires de paquets adjacents. En supposant un flux périodique [RFC3432], une heure d'arrivée attendue pourrait être définie pour tous les paquets, mais c'est essentiellement une métrique de variation de délai d'un seul point (comme défini dans les Recommandations UIT-T [I.356] et [Y.1540]) et non une métrique de réarrangement.

Des résultats de LateTime d'échantillons peuvent être exprimés comme un histogramme pour résumer la fréquence des temps de mémoire tampon nécessaires pour s'accommoder des paquets réordonnés et permettre le réglage de la mémoire tampon sur cette base. Une fonction de distribution cumulative (CDF, *cumulative distribution function*) avec les temps de mémoire tampon contre le pourcentage de paquets réordonnés accommodés peut donner des informations.

4.4 Décalage d'octet de réarrangement

Des valeurs de décalage peuvent être allouées aux paquets réordonnés pour indiquer la mémorisation en octets qu'un receveur doit posséder pour s'en accommoder. Les métriques de décalage sont calculées seulement sur les paquets réordonnés, comme identifié par la métrique de singleton de paquet réordonné de la Section 3.

4.4.1 Nom de métrique

Type-P-Packet-Byte-Offset-Stream

4.4.2 Paramètres de métrique

On utilise les mêmes paramètres que définis plus tôt, incluant les paramètres facultatifs SrcByte et PayloadSize, et on définit :

ByteOffset(s[i]) : décalage de paquet s[i] en octets

4.4.3 Définition

Le décalage de flux d'octet pour le paquet réordonné s[i] est la somme des tailles de charge utile des paquets qualifiés par les critères suivants :

- * L'arrivée est avant celle du paquet réordonné, s[i], et
- * Le numéro de séquence d'envoi, s, est supérieur à s[i].

Les paquets qui remplissent ces deux conditions sont normalement mis en mémoire tampon jusqu'à ce que la séquence derrière eux soit complète. Noter que ces critères s'appliquent aux paquets en ordre et aux paquets réordonnés.

Pour le paquet réordonné s[i] avec une étendue de réarrangement e :

$$\begin{aligned} \text{ByteOffset}(s[i]) &= \text{Somme}[\text{paquets qualifiés}] \\ &= \text{Somme}[\text{Taille de charge utile}(\text{paquet à } i-1 \text{ si qualifié}), \\ &\quad \text{Taille de charge utile}(\text{paquet à } i-2 \text{ si qualifié}), \dots \\ &\quad \text{Taille de charge utile}(\text{paquet à } i-e \text{ toujours qualifié})] \end{aligned}$$

En utilisant notre notation antérieure :

$$\text{ByteOffset}(s[i]) = \text{Somme}[\text{charges utiles de } s[j] \text{ où } s[j] > s[i] \text{ et } i > j \geq i - e]$$

4.4.4 Discussion

On note que les estimations de taille de mémoire tampon due au réarrangement dépendent largement du flux d'essai, en termes d'espacement entre les paquets d'essai et de leur taille, en particulier quand la taille de paquet est variable. Dans ces circonstances, et dans d'autres, il peut être très utile de caractériser le décalage en termes de taille de charge utile des paquets mémorisés, en utilisant la métrique Type-P-packet-Byte-Offset-Stream.

La métrique de décalage d'octet peut aider à prédire si les paquets réordonnés vont être utiles dans un système général de mémoire tampon de receveur avec des limites finies. La limite est exprimée par le nombre d'octets que la mémoire tampon peut mémoriser.

Les résultats d'un décalage d'octets d'échantillon peuvent être exprimés par un histogramme pour résumer la fréquence des longueurs de mémoire tampon nécessaires pour s'accommoder des paquets réordonnés et permettre le réglage de la mémoire tampon sur cette base. Une CDF avec la taille de mémoire tampon contre le pourcentage de paquets réordonnés accommodés peut apporter des informations.

4.5 Trous entre plusieurs discontinuités de réarrangement

4.5.1 Noms des métriques

Type-P-Packet-Reordering-Gap-Stream

Type-P-Packet-Reordering-GapTime-Stream

4.5.2 Paramètres

On utilise les mêmes paramètres définis précédemment, mais en ajoutant la convention que l'indice i' est supérieur à

l'indice i , de même que $j' > j$, et on définit :

$\text{Gap}(s[j'])$: le trou de réarrangement du paquet $s[j']$ en unités de messages entiers

et le paramètre FACULTATIF :

$\text{GapTime}(s[j'])$: le trou de réarrangement du paquet $s[j']$ en unités de secondes.

4.5.3 Définition de discontinuité de réarrangement

Tous les paquets réordonnés sont associés à un paquet à une discontinuité de réarrangement, définie comme le paquet dans l'ordre $s[j]$ qui est arrivé à la valeur minimum de j ($1 \leq j < i$) pour lequel $s[j] > s[i]$.

Noter que $s[j]$ aura été trouvé causer une discontinuité de séquence, où $s > \text{NextExp}$ quand il est évalué avec la métrique de singleton réordonné décrite au paragraphe 3.4.

On rappelle que $i - e = \min(j)$. Les paquets réordonnés suivants peuvent être associés au même $s[j]$, ou à une discontinuité différente. Ce fait est utilisé dans la définition du trou de réarrangement, ci-dessous.

4.5.4 Définition de trou de réarrangement

Un trou de réarrangement est la distance entre des discontinuités de réarrangement successives. La métrique Type-P-Packet-Reordering-Gap-Stream alloue une valeur pour $\text{Gap}(s[j'])$ à tous les paquets d'un flux (et une valeur pour $\text{GapTime}(s[j'])$, quand il est rapporté).

Si le paquet $s[j']$ se trouve être une discontinuité de réarrangement, sur la base de l'arrivée du paquet réordonné $s[i']$ avec l'étendue e' , et si une discontinuité de réarrangement antérieure $s[j]$, fondée sur l'arrivée du paquet réordonné $s[i]$ avec l'étendue e avait déjà été détecté, et $i' > i$, et si il n'y a pas de discontinuité de réarrangement entre j et j' , alors le trou de réarrangement pour le paquet $s[j']$ est la différence entre les positions d'arrivée des discontinuités de réarrangement, comme montré ci-dessous :

$$\text{Gap}(s[j']) = j' - j$$

Les trous PEUVENT aussi être exprimés en temps :

$$\text{GapTime}(s[j']) = \text{DstTime}(j') - \text{DstTime}(j)$$

Autrement :

$\text{Gap}(s[j'])$ (et $\text{GapTime}(s[j'])$) pour le paquet $s[j']$ est 0.

4.5.5 Discussion

Quand des discontinuités de réarrangement séparées peuvent être distinguées, un compte peut aussi être rapporté (ainsi que la description de la discontinuité, comme le nombre de paquets réordonnés associés à cette discontinuité et leurs étendues et décalages). Les trous entre les discontinuités de réarrangement d'un échantillon peuvent être exprimés par un histogramme pour résumer facilement la fréquence des divers trous. Rapporter le mode, la moyenne, la gamme, etc., peut aussi résumer les distributions.

La métrique de trous peut aider à corréliser la fréquence des discontinuités de réarrangement avec leur cause. Les longueurs de trous sont aussi porteuses d'informations pour les concepteurs de receveurs, en révélant la période des discontinuités de réarrangement. La combinaison des trous de réarrangement et de l'étendue révèle si les receveurs vont devoir traiter des cas de paquets réordonnés qui se chevauchent.

4.6 Cours sans réarrangement

Ce paragraphe définit une métrique fondée sur un compte des paquets dans l'ordre consécutifs entre les paquets réordonnés.

4.6.1 Noms des métriques

Type-P-Packet-Reordering-Free-Run-x-numruns-Stream

Type-P-Packet-Reordering-Free-Run-q-squruns-Stream
 Type-P-Packet-Reordering-Free-Run-p-numpkts-Stream
 Type-P-Packet-Reordering-Free-Run-a-accpkts-Stream

4.6.2 Paramètres

On utilise les mêmes paramètres définis antérieurement et on définit :

r : compteur de cours.

x : nombre de cours, aussi nombre de paquets réordonnés.

a : accumulateur de paquets dans l'ordre.

p : nombre de paquets (quand le flux est complet, $p = (x + a) = L$)

q : somme des carrés des cours comptés.

4.6.3 Définition

Lorsque les paquets d'un échantillon arrivent à destination, le compte des paquets dans l'ordre entre des paquets réordonnés est cours sans réarrangement. Noter que la longueur minimum de cours est zéro selon cette définition. Voici un exemple de pseudo code :

```
r = 0 ;                /* r est le compteur de cours */
x = 0 ;                /* x est le nombre de cours */
a = 0 ;                /* a est l'accumulateur de paquets dans l'ordre */
p = 0 ;                /* p est le nombre de paquets */
q = 0 ;                /* q est la somme des carrés des cours comptés */
```

quand (les paquets arrivent avec le numéro de séquence s)

```
{
  p++;
  si (s ≥ NextExp)      /* s est dans l'ordre */
    alors r++ ;
    a++ ;
  autrement             /* s est réordonné */
    q+ = r*r ;
    r = 0 ;
    x++ ;
}
```

Chaque arrivée dans l'ordre incrémente le compteur de cours et l'accumulateur de paquets dans l'ordre ; chaque paquet réordonné remet à zéro le compteur de cours après son ajout à la somme des carrés des longueurs.

Chaque arrivée d'un paquet réordonné donne un nouveau compte de cours. Les longs cours accompagnent les périodes où l'ordre a été conservé, tandis que les cours courts indiquent des réarrangements fréquents ou de multiples paquets.

Le pourcentage de paquets dans l'ordre est $100*a/p$

La longueur moyenne de cours sans réarrangement est a/x

Le compteur q donne une indication de la variation des cours sans réarrangement à partir de la moyenne en comparant q/a à a/x ($(q/a)/(a/x)$).

4.6.4 Discussion et illustration

Les paramètres Type-P-Packet-Reordering-Free-Run-Stream donnent un bref résumé des caractéristiques de réarrangement du flux incluant la moyenne des longueurs de cours sans réarrangement, et la variation des longueurs de cours ; donc, une application clé de cette métrique est l'évaluation du réseau.

Pour 36 paquets avec 3 cours de 11 paquets dans l'ordre, on a :

$p = 36$

$x = 3$
 $a = 33$
 $q = 3 * (11 * 11) = 363$
 moyenne des cours sans réarrangement = 11
 $q/a = 11$
 $(q/a)/(a/x) = 1,0$

Pour 36 paquets avec 3 cours, 2 cours de longueur 1, et un de longueur 31, on a :

$p = 36$
 $x = 3$
 $a = 33$
 $q = 1 + 1 + 961 = 963$
 moyenne des cours sans réarrangement = 11
 $q/a = 29.18$
 $(q/a)/(a/x) = 2.65$

La variabilité de la longueur de cours est dominante dans la différence entre les valeurs de q (somme des carrés des longueurs de cours) et en comparant la longueur moyenne de cours aux ratios $(q/a)/(a/x)$ (égaux à 1 quand tous les cours ont la même longueur).

5. Métriques centrées sur l'assertion du receveur : métrique pertinente pour TCP

Cette section décrit une métrique qui porte des informations associées aux effets du réarrangement sur TCP. Cependant, afin de pouvoir déduire quelque chose sur les performances de TCP, le flux d'essais DOIT avoir une proche ressemblance avec l'envoyeur TCP intéressé. La [RFC3148] fait la liste des aspects spécifiques des algorithmes de contrôle de l'encombrement qui doivent être spécifiés. De plus, la RFC 3148 recommande que les métriques de capacité de transfert en vrac DEVRAIENT avoir des instruments pour distinguer trois cas de réarrangement de paquets (au paragraphe 3.3). Les métriques d'échantillon définies ci-dessus satisfont les exigences de classement des paquets qui sont légèrement ou en gros déclassés. La métrique de cette section ajoute la capacité d'estimer si le réarrangement pourrait causer le dépassement du seuil DUP-ACK, ce qui causerait l'invocation de l'algorithme de retransmission rapide. D'autres instruments du cœur de TCP sont mentionnés dans [Mat03].

5.1 Nom de métrique

Type-P-Packet-n-Reordering-Stream

5.2 Notation des paramètres

Soit n un entier positif (un paramètre). Soit k un entier positif égal au nombre de paquets envoyés (taille de l'échantillon). Soit l un entier non négatif représentant le nombre de paquets reçus déclassés sur les k paquets envoyés. (Noter qu'il n'y a pas de relation entre k et l : d'un côté, les pertes peuvent faire que l soit inférieur à k ; d'un autre côté, des doublés peuvent rendre l supérieur à k .) On alloue à chaque paquet envoyé un numéro de séquence, de 1 à k , dans l'ordre de l'émission des paquets.

Soient $s[1], s[2], \dots, s[l]$ les numéros de séquence d'origine des paquets reçus, dans l'ordre d'arrivée.

5.3 Définitions

Définition 1 : le nombre de paquets reçus i ($n < i \leq l$) avec le numéro de séquence source $s[i]$, est réordonné de n si et seulement si pour tout j tel que $i - n \leq j < i$, $s[j] > s[i]$.

Remarque : Si, par cette définition, un paquet est réordonné de n et si $0 < n' < n$, alors le paquet est aussi réordonné de n' .

Note : cette définition est illustrée en code C dans l'Appendice A. Le code détermine et rapporte le réarrangement de n pour n de 1 à un paramètre spécifié (MAXN dans le code, réglé à 100). La valeur de n conjecturée comme pertinente pour TCP est le seuil de ACK dupliqué de TCP (réglé à la valeur de 3 à l'alinéa 2 du paragraphe 3.2 de la [RFC2581]).

Cette définition n'alloue pas un n à tous les paquets réordonnés comme défini par la métrique de singleton, en particulier quand des blocs de paquets successifs sont réordonnés. (Dans la séquence d'arrivée $s = \{1,2,3,7,8,9,4,5,6\}$, les paquets 4, 5, et 6 sont réordonnés, mais seul le paquet 4 est réordonné de n , avec $n = 3$.)

Définition 2 : le degré de réarrangement de n d'un échantillon est m/l , où m est le nombre de paquets réordonnés de n dans l'échantillon.

Définition 3 : le degré de réarrangement monotone d'un échantillon est son degré de réarrangement de 1.

Définition 4 : un échantillon est dit n'avoir pas de réarrangement si son degré de réarrangement monotone est 0.

Note : comme il résulte de la remarque ci-dessus, si le réarrangement monotone d'un échantillon est 0, alors le réarrangement de n de l'échantillon est 0 pour tout n .

5.4 Discussion

Le degré de réarrangement de n peut être exprimé par un pourcentage, et dans ce cas le nombre de la définition 2 est multiplié par 100.

La métrique de réarrangement de n est utile pour faire correspondre le réglage de seuil de ACK dupliqués pour un chemin donné. Par exemple, si un chemin ne présente pas plus de 5 réarrangements, un seuil de DUP-ACK de 6 peut éviter des retransmissions inutiles.

Les cas particuliers importants sont $n = 1$ et $n = 3$:

- Pour $n = 1$, l'absence de réarrangement de 1 signifie que les numéros de séquence que voit le receveur sont de croissance monotone par rapport au précédent paquet arrivant.
- Pour $n = 3$, un envoyeur TCP NewReno va retransmettre 1 paquet en réponse à une instance de réarrangement de 3 et donc considérer ce paquet comme perdu pour les besoins du contrôle d'encombrement (l'envoyeur va diviser par deux sa fenêtre d'encombrement, voir la [RFC2581]). Trois est le seuil par défaut pour le protocole de transport de commandes de flux (SCTP, *Stream Control Transport Protocol*) [RFC2960], et le protocole de contrôle d'encombrement de datagrammes (DCCP, *Datagram Congestion Control Protocol*) [RFC4340] quand ils sont utilisés avec le profil d'identifiant 2 de protocole de contrôle d'encombrement de datagrammes : contrôle d'encombrement de style TCP [RFC4341].

Un réarrangement de n d'un échantillon peut être exprimé par un histogramme pour résumer la fréquence pour chaque valeur de n .

On note que la définition du réarrangement de n ne peut pas prédire le nombre exact de paquets inutilement retransmis par un envoyeur TCP dans certaines circonstances, comme les cas de singletons réarrangés avec un faible espacement. Le temps et la position influencent tous deux le comportement de l'envoyeur.

La désignation du réarrangement de n d'un paquet est parfois égale à son étendue de réarrangement, e . Le réarrangement de n est différent de la façon suivante :

1. n est un compte des paquets antérieurs avec des positions d'arrivée consécutives chez le receveur.
2. Les paquets réordonnés (Type-P-Reordered=VRAI) peuvent n'être pas réarrangés de n , mais vont avoir une étendue, e (voir les exemples).

6. Problèmes de mesure et de mise en œuvre

Les résultats des essais vont dépendre de l'intervalle entre les paquets de mesure (à la fois à la source, et durant le transport où l'espacement peut changer). Il est clair que les paquets lancés peu fréquemment (par exemple, 1 toutes les 10 secondes) ont peu de chances d'être réordonnés.

Afin de jauger le réarrangement pour une application en accord avec les métriques définies dans le présent mémoire, il est RECOMMANDÉ d'utiliser le même schéma d'envoi que l'application intéressée. Dans tous les cas, la méthode exacte de génération de paquet DOIT être rapportée avec les résultats des mesures, incluant tous les paramètres du flux.

+ Pour faire des déductions sur des applications qui utilisent TCP, il est EXIGÉ d'utiliser des flux de style TCP, comme dans la [RFC3148]

+ Pour les applications en temps réel, il est RECOMMANDÉ d'utiliser des flux périodiques comme dans la [RFC3432]

Il est acceptable de rapporter les métriques des Sections 3 et 4 avec les autres métriques IPPM utilisant des flux de Poisson [RFC2330]. Les flux de Poisson représentent un "échantillon non biaisé" de performances du réseau pour les métriques de perte de paquet et de délai. Cependant, il serait incorrect de faire des déductions sur les catégories d'application ci-dessus en utilisant des métriques de réarrangement mesurées avec des flux de Poisson.

Les concepteurs de flux d'essais peuvent préférer utiliser un intervalle d'envoi périodique afin de conserver un biais temporel connu et permettre une analyse simplifiée des résultats (comme décrit dans la [RFC3432]). Dans ce cas, il est RECOMMANDÉ que l'intervalle d'envoi périodique soit choisi de façon à reproduire l'espacement de paquet de source le plus proche de celui attendu. Ceux qui font les essais doivent reconnaître que les flux envoyés à la limite de série de la vitesse de la liaison DOIVENT avoir une durée limitée et DOIVENT considérer la perte de paquet comme une indication que le flux a causé de l'encombrement, et suspendre la poursuite de l'essai.

Quand on a l'intention de comparer des mesures indépendantes de réarrangement, il est RECOMMANDÉ d'utiliser les mêmes paramètres de flux d'essai dans chaque système de mesure.

Les longueurs de paquet devraient aussi être variées pour tenter de détecter des instances de traitement parallèles (elles peuvent causer un réarrangement d'état permanent). Par exemple, une salve à la vitesse de ligne des plus longs paquets (longueur de la MTU) suivis par une salve des plus courts paquets possibles peut être un schéma efficace de détection. D'autres schémas de taille sont possibles.

Le critère d'ordre non réversible et toutes les métriques décrites ci-dessus restent valides et utiles quand un flux de paquets subit des pertes, ou à la fois des pertes et un réarrangement. En d'autres termes, les pertes seules ne causent pas le déclassement des paquets suivants.

Comme cette définition de métrique peut utiliser des numéros de séquence dans une gamme finie, il est possible que les numéros de séquence atteignent la fin de la gamme et reviennent à zéro durant une mesure. Par définition, la valeur NextExp ne peut pas décroître, et tous les paquets reçus après un retour à zéro seraient déclarés réordonnés. Le roulement de numéro de séquence peut être évité en utilisant des combinaisons de taille de compteur et de durée d'essais où le roulement est impossible (et la séquence est remise à zéro au départ). Aussi, le numérotage fondé sur le message résulte en une plus lente consommation de la séquence. Il peut quand même y avoir des cas où une atténuation méthodologique de ce problème est désirable (par exemple, des essais à long terme). Les éléments d'atténuation sont :

1. Il doit y avoir un essai pour détecter si un roulement s'est produit. Il serait presque impossible que les numéros de séquence de paquets successifs sautent de plus de la moitié de la gamme totale, de sorte que ces grandes discontinuités sont désignées comme un roulement.
2. Tous les numéros de séquence utilisés dans les calculs sont représentés avec une précision suffisamment grande. Une correction s'applique aux numéros (équivalente à ajouter un chiffre significatif) chaque fois qu'un retour à zéro est détecté.
3. Les paquets réordonnés coïncidant avec les numéros de séquence qui atteignent la fin de la gamme doivent aussi être détectés pour une application appropriée du facteur de correction.

Idéalement, l'instrument d'essai aurait la capacité d'utiliser tous les paquets antérieurs en tout point du flux d'essai. En pratique, la capacité de déterminer l'étendue du réarrangement sera limitée, à cause des exigences de mémorisation des paquets antérieurs. Sauvegarder seulement les paquets qui indiquent des discontinuités (et leurs positions d'arrivée) réduit le volume de mémorisation.

Une autre solution est d'utiliser une fenêtre glissante d'historique des paquets, où la taille de fenêtre serait déterminée par une limite supérieure de l'étendue de réarrangement utile. Cette limite pourrait être de plusieurs paquets ou de plusieurs secondes de paquets, selon l'analyse prévue. Quand on élimine toutes les informations de flux au delà de la fenêtre, l'étendue du réarrangement ou le degré de réarrangement de n peut devoir être exprimé comme plus grand que la longueur de la fenêtre si des informations de discontinuité de réarrangement ont été éliminées, et les calculs de trous ne seront pas possibles.

L'exigence d'ignorer les paquets dupliqués rend aussi obligatoire la mémorisation. Ici, le retraçage des numéros de séquence des paquets manquants peut minimiser la taille de mémorisation. Les paquets manquants peuvent finalement être déclarés perdus ou être réordonnés si ils arrivent. La liste des paquets manquants et le plus grand numéro de séquence reçu jusqu'alors (NextExp - 1) sont des informations suffisantes pour déterminer si un paquet est un dupliqué (en supposant une taille de mémorisation gérable pour les paquets qui manquent par suite de perte).

Il est important de noter que les réseaux IP pratiques ont aussi une capacité limitée de "mémoriser" les paquets, même

quand des boucles d'acheminement apparaissent temporairement. Donc, la mémorisation maximale pour les métriques de réarrangement (et leur complexité) approcherait seulement le nombre de paquets dans l'échantillon, K, quand le temps d'envoi pour K paquets est petit par rapport au plus grand temps de transfert possible du réseau. Une autre limitation possible sur la mémorisation est la longueur maximum du champ de numéro de séquence, en supposant que la plupart des flux d'essais ne dépassent pas en pratique cette longueur.

Enfin, on note que déterminer l'étendue de réarrangement et des trous est risqué quand il y a des événements qui se chevauchent ou sont incorporés. La complexité de l'instrument d'essais et la complexité du réarrangement sont directement corrélées.

6.1 Considérations de mesure passive

Comme avec les autres métriques IPPM, les définitions ont été construites principalement pour des mesures actives.

En supposant que les informations de séquence nécessaires (numéro de message) sont incluses dans la charge utile de paquet (éventuellement dans les en-têtes d'application comme RTP) les métriques de réarrangement peuvent être évaluées dans un arrangement de mesures passives. Il est aussi possible d'évaluer l'ordre en tout point le long d'un chemin entre source et destination, en reconnaissant que les mesures intermédiaires peuvent différer de celles faites à la destination (où l'effet du réarrangement sur les applications peut être déduit).

Il est possible d'appliquer ces métriques pour évaluer le réarrangement dans un flux d'expéditeur TCP. Dans ce cas, les numéros de séquence de source vont se fonder sur le flux d'octets ou le dénombrement de segments. Comme le flux peut inclure des retransmissions dues aux pertes ou au réarrangement, il faut prendre soin d'éviter de déclarer réordonnés les paquets retransmis. La référence de séquence supplémentaire de *s* ou *SrcTime* aide à éviter cette ambiguïté dans les mesures actives, ou le champ facultatif *Horodatage TCP* [RFC1323] dans les mesures passives.

7. Exemples d'évaluation d'ordre d'arrivée

Cette section donne des exemples pour illustrer comment fonctionne le critère d'ordre non réversible, comment le réarrangement de *n* fonctionne dans les comparaisons, et la valeur de la quantification du réarrangement dans toutes les dimensions de temps, d'octets, et de positions.

Dans cette Section, on se réfère aux paquets par leur numéro de séquence de source, sauf lorsque noté autrement. Ainsi "paquet 4" se réfère au paquet dont le numéro de séquence de source est 4, et le lecteur devrait se référer aux tableaux dans chaque exemple pour déterminer le numéro d'indice d'arrivée du paquet 4, si nécessaire.

7.1 Exemple avec un seul paquet réarrangé

Le Tableau 1 donne un cas simple de réarrangement, où un paquet est réordonné, le paquet 4. La liste des paquets est dans l'ordre d'arrivée, et la numérotation de message est utilisée. Tous les paquets contiennent *PayloadSize* = 100 octets, avec *SrcByte* = (*s* x 100) - 99 pour *s* = 1,2,3,4,...

Tableau 1 : exemple avec le paquet 4 réordonné, ordre d'envoi (*s @Src*) : 1,2,3,4,5,6,7,8,9,10

<i>s @Dst</i>	<i>NextExp</i>	Temps source	Temps dest.	Délai	IPDV	Ordre dest.	Décal. octets	Tardiveté
1	1	0	68	68	1			
2	2	20	88	68	0	2		
3	3	40	108	68	0	3		
5	4	80	148	68	-82	4		
6	6	100	168	68	0	5		
7	7	120	188	68	0	6		
8	8	140	208	68	0	7		
4	9	60	210	150	82	8	400	62
9	9	160	228	68	0	9		
10	10	180	248	68	0	10		

Chaque colonne donne les informations suivantes :

s : numéro de séquence du paquet à la source.

NextExp : valeur de NextExp quand le paquet arrive (avant mise à jour).

Temps source : horodatage du paquet à la source, en ms.

Temps dest : horodatage du paquet à la destination, en ms.

Délai : délai unidirectionnel du paquet, en ms.

IPDV : variation du délai de paquet IP, en ms. $IPDV = \text{Délai}(\text{SrcNum}) - \text{Délai}(\text{SrcNum}-1)$

Ordre dest : ordre d'arrivés du paquet à destination.

Décal octet : décalage d'octet du paquet réordonné, en octets.

LateTime : tardiveté d'un paquet réordonné, en ms.

On peut voir que quand le paquet 4 arrive, NextExp = 9, et il est déclaré réordonné. On calcule l'étendue du réarrangement comme suit :

En utilisant la notation $\langle s[1], \dots, s[i], \dots, s[L] \rangle$, les paquets reçus sont représentés comme :

$$\begin{array}{c} \vee \\ s = 1, 2, 3, 5, 6, 7, 8, 4, 9, 10 \\ i = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 \\ \wedge \end{array}$$

En appliquant la définition de Type-P-Packet-Reordering-Extent-Stream :

quand $j = 7, 8 > 4$, donc l'étendue de réarrangement est 1 ou plus.

quand $j = 6, 7 > 4$, donc l'étendue de réarrangement 2 ou plus.

quand $j = 5, 6 > 4$, donc l'étendue de réarrangement 3 ou plus.

quand $j = 4, 5 > 4$, donc l'étendue de réarrangement 4 ou plus.

quand $j = 3$, mais $3 < 4$, et 4 est l'étendue maximum, $e = 4$ (en supposant qu'il n'y a pas de discontinuité de séquence antérieure, comme dans cet exemple).

De plus, on peut calculer le temps de retard ($210 - 148 = 62$ ms en utilisant le temps de destination) comparé à l'arrivée du paquet 5. Si le receveur a une mémoire tampon de gigue qui contient plus de 4 paquets, ou au moins 62 ms de mémorisation, le paquet 4 peut être utile. Noter que le délai unidirectionnel et l'IPDV indiquent un comportement inhabituel pour le paquet 4. Aussi, si le paquet 4 était arrivé au moins 62 ms plus tôt, il aurait été dans l'ordre, dans cet exemple.

Si tous les paquets contenaient des charges utiles de 100 octets, alors le décalage d'octet est égal à 400 octets.

Suivant les définitions du paragraphe 5.1, le paquet 4 est désigné comme réarrangé de 4.

7.2 Exemple avec deux paquets réarrangés

Tableau 2 : exemple avec les paquets 5 et 6 réordonnés, ordre d'envoi (s @Src) : 1,2,3,4,5,6,7,8,9,10

s @Dst	NextExp	Temps source	Temps dest.	Délai	IPDV	Ordre dest.	Décal. octets	Tardiveté
1	1	0	68	68	1			
2	2	20	88	68	0	2		
3	3	40	108	68	0	3		
4	4	60	128	68	0	4		
7	5	120	188	68	-22	5		
5	8	80	189	109	41	6	100	1
6	8	100	190	90	-19	7	100	2
8	8	140	208	68	0	8		
9	9	160	228	68	0	9		
10	10	180	248	68	0	10		

La Tableau 2 montre un cas où les paquets 5 et 6 arrivent juste derrière le paquet 7, donc 5 et 6 sont tous deux réordonnés. Les tardivetés ($189 - 188 = 1$, $190 - 188 = 2$) sont petites.

En utilisant la notation $\langle s[1], \dots, s[i], \dots, s[L] \rangle$, les paquets reçus sont représentés comme :

$$\begin{array}{c}
 \vee \vee \\
 s = 1, 2, 3, 4, 7, 5, 6, 8, 9, 10 \\
 i = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 \\
 \wedge \wedge
 \end{array}$$

Considérant d'abord le paquet 5 :
 quand $j = 5, 7 > 5$, donc l'étendue de réarrangement est 1 ou plus.
 quand $j = 4, 4 < 5$, donc 1 est son étendue maximum, et $e = 1$.

Considérant ensuite le paquet 6 :
 quand $j = 6, 5 < 6$, l'étendue n'est pas encore définie.
 quand $j = 5, 7 > 6$, donc l'étendue de réarrangement est $i - j = 2$ ou plus.
 quand $j = 4, 4 < 6$, et on trouve que 2 est son étendue maximum, et $e = 2$.

On peut aussi associer chacun de ces paquets réordonnés avec une discontinuité de réarrangement. On trouve le minimum $j = 5$ (pour les deux paquets) conformément au paragraphe 4.2.3. Donc le paquet 6 est associé à la même discontinuité de réarrangement que le paquet 5, la discontinuité de réarrangement au paquet 7.

C'est un cas où l'étendue de réarrangement va surestimer la mémorisation de paquet requise pour restaurer l'ordre. Une seule mémorisation de paquet est requise (pour contenir le paquet 7) mais $e = 2$ pour le paquet 6.

Suivant les définitions de la Section 5, le paquet 5 est désigné comme réarrangé de 1, mais le paquet 6 n'est pas désigné comme réarrangé de n.

Une paire hypothétique d'expéditeur/receveur peut retransmettre inutilement le paquet 5, car il est réarrangé de 1 (en accord avec la métrique de singleton). Bien que le paquet 6 ne puisse pas être inutilement retransmis, le receveur ne peut pas avancer le paquet 7 aux couches supérieures avant que le paquet 6 arrive. Donc, la métrique de singleton a déterminé correctement que le paquet 6 est réordonné.

7.3 Exemple avec trois paquets réarrangés

Tableau 3 : exemple avec les paquets 4, 5, et 6 réordonnés ; ordre d'envoi (s @Src) : 1,2,3,4,5,6,7,8,9,10,11

s @Dst	NextExp	Temps source	Temps dest.	Délai	IPDV	Ordre dest.	Décal. octets	Tardiveté
1	1	0	68	68	1			
2	2	20	88	68	0	2		
3	3	40	108	68	0	3		
7	4	120	188	68	-88	4		
8	8	140	208	68	0	5		
9	9	160	228	68	0	6		
10	10	180	248	68	0	7		
4	11	60	250	190	122	8	400	62
5	11	80	252	172	-18	9	400	64
6	11	100	256	156	-16	10	400	68
11	11	200	268	68	0	11		

Le cas du Tableau 3 est lorsque trois paquets dans la séquence ont de longs temps de transit (les paquets avec $s = 4, 5$, et 6). Le délai, la tardiveté, et le décalage d'octets capturent très bien cela, et indiquent la variation de l'étendue de réarrangement, tandis que IPDV indique que l'espacement entre les paquets 4,5,et 6 a changé.

L'histogramme des étendues de réarrangement (e) serait :

Bin	1	2	3	4	5	6	7
Fréquence	0	0	0	1	1	1	0

En utilisant la notation $\langle s[1], \dots, s[i], \dots, s[l] \rangle$, les paquets reçus sont représentés comme :

$$\begin{array}{l}
 s = 1, 2, 3, 7, 8, 9, 10, 4, 5, 6, 11 \\
 i = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11
 \end{array}$$

On calcule d'abord le réarrangement de n. On considère d'abord le paquet 4 :

quand $n = 1$, $7 \leq j < 8$, et $10 > 4$, donc le paquet est réarrangé de 1.
 quand $n = 2$, $6 \leq j < 8$, et $9 > 4$, donc le paquet est réarrangé de 2.
 quand $n = 3$, $5 \leq j < 8$, et $8 > 4$, donc le paquet est réarrangé de 3.
 quand $n = 4$, $4 \leq j < 8$, et $7 > 4$, donc le paquet est réarrangé de 4.
 quand $n = 5$, $3 \leq j < 8$, mais $3 < 4$, et 4 est le maximum de réarrangement de n.

Considérant ensuite le paquet 5[9] :

quand $n = 1$, $8 \leq j < 9$, mais $4 < 5$, donc le paquet à $i = 9$ n'est pas désigné comme réarrangé de n. On trouve le même résultat pour le paquet 6.

On considère maintenant si les paquets réordonnés 5 et 6 sont associés à la même discontinuité de réarrangement que le paquet 4. En utilisant l'essai du paragraphe 4.2.3, on trouve le minimum $j = 4$ pour les trois paquets. Ils sont tous associés à la discontinuité de réarrangement du paquet 7.

Cet exemple montre encore que la définition du réarrangement de n identifie un seul paquet (4) avec un degré suffisant de réarrangement de n qui pourrait causer une retransmission de paquet inutile par l'envoyeur New Reno TCP (avec le seuil de DUP-ACK = 3 ou 4). Aussi, l'arrivée réordonnée des paquets 5 et 6 va permettre au processus receveur de passer les paquets 7 à 10 à la pile de protocoles (le singleton Type-P-Reordered = VRAI pour les paquets 5 et 6, et ils sont tous associés à une seule discontinuité de réarrangement).

7.4 Exemple avec plusieurs discontinuités de réarrangement de paquet

**Tableau 4 : Exemple avec plusieurs discontinuité de réarrangement de paquet ; ordre d'envoi (s @Src) :
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16**

	Discontinuité	Discontinuité
	-----Trou-----	
s =	1, 2, 3, 6, 7, 4, 5, 8, 9, 10, 12, 13, 11, 14, 15, 16	
i =	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	
r =	1, 2, 3, 4, 5, 0, 0, 1, 2, 3, 4, 5, 0, 1, 2, 3, ...	
nombre de tours, n =	1 2	3
Fin de r comptes =	5 0	5

(Ces valeurs sont calculées après l'arrivée du paquet.)

Le paquet 4 a une étendue $e = 2$, le paquet 5 a l'étendue $e = 3$, et le paquet 11 a $e = 2$. Il y a deux différentes discontinuités de réarrangement, une au paquet 6 (où $j = 4$) et une au paquet 12 (où $j' = 11$).

Selon la définition de trou de réarrangement :

Trou(s[j']) = (j') - (j)

Trou(paquet 12) = (11) - (4) = 7

On a aussi trois cours sans réarrangement de longueurs 5, 0, et 5.

Les différences entre ces deux métriques multi événements sont évidentes ici. Les trous sont la distance entre les discontinuités de séquence qui sont ensuite définies comme des discontinuités de réarrangement, tandis que les cours sans réarrangement capturent la distance entre les paquets réordonnés.

8. Considérations sur la sécurité

8.1 Attaques de déni de service

Cette métrique exige un flux de paquets envoyés d'un hôte (source) à un autre hôte (destination) à travers des réseaux intermédiaires. Cette méthode pourrait être utilisée pour des attaques de déni de service dirigées contre le ou les réseaux de destination et/ou intermédiaires.

Les administrateurs des réseaux de source, de destination, et intermédiaires devraient établir des accords bilatéraux ou

multilatéraux concernant le moment, la taille, et la fréquence de collecte des métriques d'échantillons. L'utilisation de cette méthode au delà des termes des accords entre les participants peut causer un rejet immédiat ou l'élimination des paquets ou autres procédures définies entre les parties affectées.

8.2 Confidentialité des données d'utilisateur

L'utilisation active de cette méthode génère des paquets pour un échantillon, plutôt que de prendre des échantillons sur la base des données d'utilisateur, et ne menace pas la confidentialité des données d'utilisateur. Les mesures passives doivent se restreindre aux en-têtes intéressants. Comme les charge utiles d'utilisateur peuvent être temporairement mémorisées pour une longue analyse, des précautions convenables DOIVENT être prises pour garder ces informations en sécurité et confidentielles. Dans la plupart des cas, une fonction de hachage va produire une valeur convenable pour les comparaisons de charges utiles.

8.3 Interférence avec la métrique

Il est possible d'identifier qu'un certain paquet ou flux de paquets fait partie d'un échantillon. Avec cette connaissance à la destination et/ou aux réseaux intermédiaires, il est possible de changer le traitement des paquets (par exemple, en augmentant ou en diminuant le délai) qui peut fausser les performances mesurées. Il est aussi possible de générer des paquets supplémentaires qui paraissent faire partie de la métrique d'échantillons. Ces paquets supplémentaires vont probablement perturber les résultats de la mesure de l'échantillon. Les conséquences probables d'une injection de paquets sont que les paquets supplémentaires vont être déclarés dupliqués, ou que les paquets originaux vont être vus comme des dupliqués (si ils arrivent après les paquets injectés correspondants) causant des mesures invalides sur les paquets injectés.

Les exigences pour la résistance de la collecte de données aux interférences par des parties malveillantes et les mécanismes pour réaliser une telle résistance sont disponibles dans les autres documents IPPM. Un ensemble d'exigences pour un protocole de collecte de données se trouve dans la [RFC3763], et une spécification de protocole pour le protocole de mesures actives unidirectionnelles (OWAMP, *One-Way Active Measurement Protocol*) est dans la [RFC4656]. Les sections de considérations sur la sécurité des deux documents OWAMP sont extensives et devraient être consultées pour les détails.

9. Considérations relatives à l'IANA

Les métriques définies dans le présent mémoire ont été enregistrées dans le registre de l'IANA Métriques IPPM comme décrit dans la version initiale du registre [RFC4148].

L'IANA a enregistré les métriques suivantes dans IANA-IPPM-METRICS-REGISTRY-MIB :

```
ietfReorderedSingleton OBJECT-IDENTITY
STATUT : courant
DESCRIPTION : "Type-P-Reordered"
RÉFÉRENCE : "Reference RFC 4737, Section 3" ::= { ianaIppmMetrics 34 }

ietfReorderedPacketRatio OBJECT-IDENTITY
STATUT : courant
DESCRIPTION : "Type-P-Reordered-Ratio-Stream"
RÉFÉRENCE : "Reference RFC 4737, Section 4.1" ::= { ianaIppmMetrics 35 }

ietfReorderingExtent OBJECT-IDENTITY
STATUT : courant
DESCRIPTION : "Type-P-Packet-Reordering-Extent-Stream"
RÉFÉRENCE : "Reference RFC 4737, Section 4.2" ::= { ianaIppmMetrics 36 }

ietfReorderingLateTimeOffset OBJECT-IDENTITY
STATUT : courant
DESCRIPTION : "Type-P-Packet-Late-Time-Stream"
RÉFÉRENCE : "Reference RFC 4737, Section 4.3" ::= { ianaIppmMetrics 37 }

ietfReorderingByteOffset OBJECT-IDENTITY
```

STATUT : courant
 DESCRIPTION : "Type-P-Packet-Byte-Offset-Stream"
 RÉFÉRENCE : "Reference RFC 4737, Section 4.4" ::= { ianaIppmMetrics 38 }

ietfReorderingGap OBJECT-IDENTITY
 STATUT : courant
 DESCRIPTION : "Type-P-Packet-Reordering-Gap-Stream"
 RÉFÉRENCE : "Reference RFC 4737, Section 4.5" ::= { ianaIppmMetrics 39 }

ietfReorderingGapTime OBJECT-IDENTITY
 STATUT : courant
 DESCRIPTION : "Type-P-Packet-Reordering-GapTime-Stream"
 RÉFÉRENCE : "Reference RFC 4737, Section 4.5" ::= { ianaIppmMetrics 40 }

ietfReorderingFreeRunx OBJECT-IDENTITY
 STATUT : courant
 DESCRIPTION : "Type-P-Packet-Reordering-Free-Run-x-numruns-Stream"
 RÉFÉRENCE : "Reference RFC 4737, Section 4.6" ::= { ianaIppmMetrics 41 }

ietfReorderingFreeRunq OBJECT-IDENTITY
 STATUT : courant
 DESCRIPTION : "Type-P-Packet-Reordering-Free-Run-q-sqruns-Stream"
 RÉFÉRENCE : "Reference RFC 4737, Section 4.6" ::= { ianaIppmMetrics 42 }

ietfReorderingFreeRunp OBJECT-IDENTITY
 STATUT : courant
 DESCRIPTION : "Type-P-Packet-Reordering-Free-Run-p-numpkts-Stream"
 RÉFÉRENCE : "Reference RFC 4737, Section 4.6" ::= { ianaIppmMetrics 43 }

ietfReorderingFreeRuna OBJECT-IDENTITY
 STATUT : courant
 DESCRIPTION : "Type-P-Packet-Reordering-Free-Run-a-accpkts-Stream"
 RÉFÉRENCE : "Reference RFC 4737, Section 4.6" ::= { ianaIppmMetrics 44 }

ietfReordering OBJECT-IDENTITY
 STATUT : courant
 DESCRIPTION : "Type-P-Packet-n-Reordering-Stream"
 RÉFÉRENCE : "Reference RFC 4737, Section 5" ::= { ianaIppmMetrics 45 }

10. Références normatives

- [RFC0791] J. Postel, éd., "Protocole Internet - Spécification du [protocole du programme Internet](#)", STD 5, septembre 1981.
- [RFC2119] S. Bradner, "[Mots clés à utiliser](#) dans les RFC pour indiquer les niveaux d'exigence", BCP 14, mars 1997. (MàJ par [RFC8174](#))
- [RFC2330] V. Paxson, G. Almes, J. Mahdavi, M. Mathis, "[Cadre pour la mesure des performances](#) d'IP", mai 1998. (Information ; MàJ par [RFC8468](#))
- [RFC2460] S. Deering et R. Hinden, "Spécification du [protocole Internet, version 6](#) (IPv6)", décembre 1998. (MàJ par [5095](#), [6564](#) ; D.S ; Remplacée par [RFC8200](#), STD 86)
- [RFC3148] M. Mathis, M. Allman, "Cadre de définition de métrique empirique de capacité de transfert brute", juillet 2001. (Info.)
- [RFC3432] V. Raisen et autres, "Mesure des [performances réseau avec des flux périodiques](#)", novembre 2002. (P.S.)
- [RFC3763] S. Shalunov, B. Teitelbaum, "Exigences pour le protocole de mesures actives en unidirectionnel (OWAMP)", avril 2004. (Information)

[RFC4148] E. Stephan, "Registre des métriques de mesure des performances d'IP (IPPM)", août 2005. [BCP0108](#) ; *Obs, voir RFC6248*)

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

11. Références pour information

[Bel02] J. Bellardo and S. Savage, "Measuring Packet Reordering," Proceedings of the ACM SIGCOMM Internet Measurement Workshop 2002, 6-8 novembre, Marseille, France.

[Ben99] J.C.R. Bennett, C. Partridge, and N. Shectman, "Packet Reordering is Not Pathological Network Behavior," IEEE/ACM Transactions on Networking, vol. 7, no. 6, pp. 789-798, décembre 1999.

[Cia00] L. Ciavattone et A. Morton, "Out-of-Sequence Packet Parameter Definition (for Y.1540)", Contribution T1A1.3/2000-047, 30 octobre 2000, <http://home.comcast.net/~acmacm/IDcheck/0A130470.doc> .

[Cia03] L. Ciavattone, A. Morton, and G. Ramachandran, "Standardized Active Measurements on a Tier 1 IP Backbone," IEEE Communications Mag., pp. 90-97, juin 2003.

[I.356] Recommandation UIT-T I.356, "Performances de transfert de cellules à la couche ATM du RNIS-LB", Union Internationale des Télécommunications, Genève, mars 2000.

[Jai02] S. Jaiswal and al., "Measurement et Classification of Out-of-Sequence Packets in a Tier-1 IP Backbone," Proceedings of the ACM SIGCOMM Internet Measurement Workshop, 6-8 novembre 2002, Marseille, France.

[Lou01] D. Loguinov and H. Radha, "Measurement Study of Low- bitrate Internet Video Streaming", Proceedings of the ACM SIGCOMM Internet Measurement Workshop, 1-2 novembre 2001, San Francisco, USA.

[Mat03] M. Mathis, J. Heffner, and R. Reddy, "Web100: Extended TCP Instrumentation for Research, Education and Diagnosis", ACM Computer Communications Review, Vol 33, Num 3, juillet 2003, <http://www.web100.org/docs/mathis03web100.pdf> .

[Pax98] V. Paxson, "Measurements and Analysis of End-to-End Internet Dynamics", Ph.D. dissertation, U.C. Berkeley, 1997, <ftp://ftp.ee.lbl.gov/papers/vp-thesis/dis.ps.gz> .

[RFC0793] J. Postel (éd.), "Protocole de [commande de transmission](#) – Spécification du protocole du programme Internet DARPA", STD 7, septembre 1981.

[RFC1323] V. Jacobson, R. Braden et D. Borman, "[Extensions TCP](#) pour de bonnes performances", mai 1992.

[RFC2581] M. Alman, V. Paxson et W. Stevens, "[Contrôle d'encombrement avec TCP](#)", avril 1999. (*Obsolète, voir RFC5681*)

[RFC2679] G. Almes, S. Kalidindi, M. Zekauskas, "[Métrique de délai unidirectionnel pour IPPM](#)", septembre 1999. (*P.S. ; Remplacée par RFC7679*, STD 81)

[RFC2680] G. Almes, S. Kalidindi, M. Zekauskas, "[Métrique de perte de paquet unidirectionnelle pour IPPM](#)", septembre 1999. *P.S. ; Remplacée par RFC7680*)

[RFC2960] R. Stewart et autres, "Protocole de transmission de commandes de flux", octobre 2000. (*Obsolète, voir RFC4960*) (*P.S.*)

[RFC3393] C. Demichelis, P. Chimento, "[Métrique de variation de délai de paquet IP](#) pour la mesure des performances IP (IPPM)", novembre 2002. (*P.S.*)

[RFC4340] E. Kohler et autres, "[Protocole de contrôle d'encombrement](#) de datagrammes (DCCP)", mars 2006. (*P.S.*) (*MàJ par 6773*)

[RFC4341] S. Floyd, E. Kohler, "[Profil d'identifiant 2 de protocole](#) de contrôle d'encombrement de datagrammes

(DCCP) : Contrôle d'encombrement de style TCP", mars 2006. (P.S. ; MàJ par [RFC8311](#))

[TBABAJ02] T. Banka, A. Bare, A. P. Jayasumana, "Metrics for Degree of Reordering in Packet Sequences", Proc. 27th IEEE Conference on Local Computer Networks, Tampa, FL, novembre 2002.

[Y.1540] Recommandation UIT-T Y.1540, "Service de communication de données au protocole Internet – Transfert de paquet IP et paramètre de performances de disponibilité", Union Internationale des Télécommunications, Genève, décembre 2002.

12. Remerciements

Les auteurs tiennent à remercier de nombreuses discussions utiles Matt Zekauskas, Jon Bennett (qui est l'auteur des paragraphes sur le fonctionnement sans réarrangement) et Matt Mathis. Merci à David Newman, Henk Uijterwaal, Mark Allman, Vern Paxson, et Phil Chimento de leur relecture et suggestions, et à Michal Przybylski qui a partagé avec nous les expériences de mise en œuvre sur la liste de diffusion ippm. Anura Jayasumana et Nischal Piratla nous ont apporté leur récent travail en cours [TBABAJ02]. Nous reconnaissons notre dette à l'égard des auteurs du cadre pour les performances de IP [RFC2330].

Appendice A. Exemple de mise en œuvre en C (information)

On donne ci-après deux exemples de mise en œuvre de code c des définitions de réarrangement :

Exemple 1 réarrangement de n =====

```
#include <stdio.h>

#define MAXN 100
#define min(a, b) ((a) < (b)? (a): (b))
#define loop(x) ((x) >= 0? x: x + MAXN)

/* Lire le nouveau numéro de séquence et le retourner. Retourner une valeur sentinelle de Fin de fichier (EOF) (au moins
une fois) quand il n'y a plus de numéro de séquence. Dans cet exemple, les numéros de séquence viennent de stdin ; dans
un essai réel, ils viendraient du réseau. */

int
read_sequence_number()
{
    int res, rc;
    rc = scanf("%d\n", &res);
    si (rc == 1) retourner res;
    autrement retourner EOF;
}

int
main()
{
    int m[MAXN]; /* on a m[j-1] == nombre de paquets réarrangés de j. */
    int ring[MAXN]; /* Derniers numéros de séquence vus. */
    int r = 0; /* Pointeur de ring pour la prochaine écriture. */
    int l = 0; /* Nombre de numéros de séquence lus. */
    int s; /* Dernier numéro de séquence lu. */
    int j;

    pour (j = 0; j < MAXN; j++) m[j] = 0;
    pour (;(s = read_sequence_number()) != EOF; l++, r=(r+1)%MAXN) {
        pour (j=0; j<min(l, MAXN)&& s<ring[loop(r-j-1)]; j++) m[j]++;
    }
}
```

```

    ring[r] = s;
}

pour (j = 0; j < MAXN && m[j]; j++)
    printf("%d-reordering = %f%%\n", j+1, 100.0*m[j]/(1-j-1));
si (j == 0) printf("pas de reordering\n");
autrement si (j < MAXN) printf("pas de %d-reordering\n", j+1);
autrement printf("seulement jusqu'au traitement de %d-reordering \n", MAXN);
exit(0);
}

```

/* Exemple 2 : comparaison de singleton et réarrangement de n =====

Auteur : Jerry Perser 7-2002 (modifié par acm 12-2004)

Compile : \$ gcc -o jpboth file.c

Usage : \$ jpboth 1 2 3 7 8 4 5 6 (séquence de paquets donnée sur la ligne cmd)

Note pour les coupeurs/colleurs : la ligne 59 peut devoir être réparée */

```

#include <stdio.h>

#define MAXN 100
#define min(a, b) ((a) < (b)? (a): (b))
#define loop(x) ((x) >= 0? x: x + MAXN)

/* Compteurs globaux */
int receive_paquets=0;          /* nombre de paquets reçus */
int reorder_paquets_A1=0;      /* nombre de paquets réordonnés (singleton) */
int reorder_paquets_Stas=0;    /* nombre de paquets réordonnés (réarrangement de n)*/

/* La fonction à tester si le paquet courant a été réordonné retourne 0 pour non réordonné et 1 pour réordonné */

int testorder1(int seqnum) // A1
{
    static int NextExp = 1;
    int iReturn = 0;

    si (seqnum >= NextExp) {
        NextExp = seqnum+1;
    } autrement {
        iReturn = 1;
    }
    return iReturn;
}

int testorder2(int seqnum) // Stanislav
{
    static int ring[MAXN];      /* Derniers numéros de séquence vus. */
    static int r = 0;          /* Pointeur ring pour la prochaine écriture */
    int l = 0;                 /* Nombre de numéros de séquence lus. */
    int j;
    int iReturn = 0;

    l++;
    r = (r+1) % MAXN;
    for (j=0; j<min(l, MAXN) && seqnum<ring[loop(r-j-1)]; j++)
        iReturn = 1;
    ring[r] = seqnum;
    return iReturn;
}

int main(int argc, char *argv[])
{

```

```

int i, packet;
for (i=1; i< argc; i++) {
    receive_packets++;
    paquet = atoi(argv[i]);
    reorder_packets_AI += testorder1(packet); // singleton
    reorder_packets_Stas += testorder2(packet); //n-reord.
}
printf("paquets reçus = %d, Singleton Reordered = %d, n-
reordered = %d\n", receive_packets, reorder_packets_AI,
reorder_packets_Stas );
exit(0);
}

```

Référence : ISO/IEC 9899:1999 (E), tel qu'amendé par ISO/IEC 9899:1999/Cor.1:2001 (E). Aussi publié comme : The C Standard: Incorporating Technical Corrigendum 1, British Standards Institute, ISBN: 0-470-84573-2, Hardcover, 558 pages, septembre 2003.

Appendice B. Évaluation de l'ordre des fragments (information)

La Section 3 déclare que le réassemblage de fragments est supposé avoir lieu avant l'évaluation de l'ordre, mais que des procédures similaires pourraient être appliquées avant le ré-assemblage. Cet appendice donne les définitions et procédures pour identifier le réarrangement dans un flux de paquets qui inclut des fragmentations.

B.1 Nom de métrique

La métrique conserve le même nom, Type-P-Reordered, mais des paramètres supplémentaires sont nécessaires.

Cet appendice suppose que l'appareil qui divise un paquet en fragments les envoie selon un décalage de fragment ascendant. Les anciens systèmes d'exploitation Linux envoyaient les fragments en ordre inverse, dont il vaut la peine de vérifier cette possibilité.

B.2 Paramètres de métrique supplémentaires

MoreFrag : état du fanion Plus de fragments dans l'en-tête IP.

FragOffset : décalage depuis le début d'un paquet fragmenté, en unités de 8 octets (aussi à partir de l'en-tête IP).

FragSeq# : numéro de séquence à partir de l'en-tête IP d'un paquet fragmenté en cours d'évaluation pour réarrangement.

Réglé à zéro, l'évaluation de fragment n'est pas en cours.

NextExpFrag : prochain décalage de fragment attendu à la destination, en unités de 8 octets. Réglé à zéro quand l'évaluation de fragment n'est pas en cours.

Le numéro de séquence de paquet, s, est supposé être le même que le numéro de séquence de l'en-tête IP. Aussi, la valeur de NextExp ne change pas avec l'arrivée dans l'ordre des fragments. NextExp n'est mis à jour que quand arrive un dernier fragment ou un paquet complet.

Noter que les paquets avec des fragments manquants DOIVENT être déclarés perdus, et l'état de réarrangement de tout fragment qui arrive bien DOIT être exclu des métriques d'échantillon.

B.3 Définition

La valeur de Type-P-Reordered est normalement fausse (le paquet est dans l'ordre) quand

* le numéro de séquence $s \geq \text{NextExp}$, ET

* le décalage de fragment $\text{FragOffset} \geq \text{NextExpFrag}$

Cependant, il est plus efficace de définir exactement les conditions réordonnées et de désigner Type-P-Reordered comme Faux autrement.

La valeur de Type-P-Reordered est définie comme Vrai (le paquet est réordonné) dans les conditions ci-dessous. Dans ce cas, la valeur NextExp ne change pas.

Cas 1 : si $s < \text{NextExp}$
 Cas 2 : si $s < \text{FragSeq\#}$
 Cas 3 : si $s \geq \text{NextExp}$ ET $s = \text{FragSeq\#}$ ET $\text{FragOffset} < \text{NextExpFrag}$

Cette définition peut aussi être illustrée en pseudo-code. Une version du code suit, et des simplifications sont possibles. L'entretien des nouveaux paramètres va être un défi.

```

NextExp=0;
NextExpFrag=0;
FragSeq#=0;

while(paquets arrive avec s, MoreFrag, FragOffset)
{
si (s>=NextExp ET MoreFrag==0 ET s>=FragSeq#){
/* un paquet normal ou un dernier fragment d'un paquet est arrivé dans l'ordre */
  NextExp = s+1;
  FragSeq# = 0;
  NextExpFrag = 0;
  Reordering = Faux;
}
si (s>=NextExp ET MoreFrag==1 ET s>FragSeq# ≥ 0){
/* un fragment d'un nouveau paquet est arrivé, éventuellement avec un numéro de séquence supérieur au paquet fragmenté en cours */
  FragSeq# = s;
  NextExpFrag = FragOffset+1;
  Reordering = Faux;
}
si (s>=NextExp ET MoreFrag==1 ET s==FragSeq#){
/* un fragment du "paquet en cours s" est arrivé */

  si (FragOffset ≥ NextExpFrag){
    NextExpFrag = FragOffset+1;
    Reordering = Faux;
  }
  autrement{
    Reordering = Vrai;          /* fragment réordonné */
  }
}
si (s>=NextExp ET MoreFrag==1 ET s < FragSeq#){
/* cas où un fragment en retard est arrivé, seulement pour illustration, redondant avec le "autrement" ci-dessous */
  Reordering = Vrai;
}
autrement { /* quand s < NextExp, ou MoreFrag==0 ET s < FragSeq# */
  Reordering = Vrai;
}
}

```

Un version opérationnelle du code comporterait une vérification pour s'assurer que tous les fragments d'un paquet arrivent avant d'utiliser l'état de réarrangement, comme des métriques d'échantillon.

B.4 Discussion : Notes sur les métriques d'échantillon dans l'évaluation de fragments

Tous les fragments avec le même numéro de séquence de source sont affectés à la même heure de source. L'évaluation avec une numérotation de flux d'octets peut être simplifiée si le décalage de fragment est simplement ajouté au SourceByte du premier paquet (avec le décalage de fragment = 0) en gardant en mémoire les unités de 8 octets du décalage.

Appendice C. Responsabilité et licence

En ce qui concerne le présent document en tout ou partie (incluant le pseudo-code et le code C) les auteurs ne donnent

aucune garantie et ne sont responsables d'aucun dommage résultant de son utilisation. Les auteurs accordent à quiconque une permission irrévocable de l'utiliser, le modifier, et le distribuer de toutes façons qui ne diminuent pas les droits de tout autre de l'utiliser, le modifier, et le distribuer, pourvu que les travaux dérivés redistribués ne contiennent pas d'informations trompeuses sur les auteurs ou la version. Les travaux dérivés n'ont pas besoin d'être licenciés sous des termes similaires.

Adresse des auteurs

Al Morton
AT&T Labs
Room D3 - 3C06
200 Laurel Ave. South
Middletown, NJ 07748 USA
téléphone : +1 732 420 1571
mél : acmorton@att.com

Len Ciavattone
AT&T Labs
Room A2 - 4G06
200 Laurel Ave. South
Middletown, NJ 07748 USA
téléphone : +1 732 420 1239
mél : lencia@att.com

Gomathi Ramachandran
AT&T Labs
Room C4 - 3D22
200 Laurel Ave. South
Middletown, NJ 07748 USA
téléphone : +1 732 420 2353
mél : gomathi@att.com

Stanislav Shalunov
Internet2
1000 Oakbrook DR STE 300
Ann Arbor, MI 48104
téléphone : +1 734 995 7060
mél : shalunov@internet2.edu

Jerry Perser
Veriwave
8770 SW Nimbus Ave.
Suite B
Beaverton, OR 97008 USA
téléphone : +1 818 338 4112
mél : jperser@veriwave.com

Déclaration complète de droits de reproduction

Copyright (C) The Internet Society (2006)

Le présent document est soumis aux droits, licences et restrictions contenus dans le BCP 78, et sauf pour ce qui est mentionné ci-après, les auteurs conservent tous leurs droits.

Le présent document et les informations contenues sont fournies sur une base "EN L'ÉTAT" et le contributeur, l'organisation qu'il ou elle représente ou qui le/la finance (s'il en est), la INTERNET SOCIETY, le IETF TRUST et la INTERNET ENGINEERING TASK FORCE déclinent toutes garanties, exprimées ou implicites, y compris mais non limitées à toute garantie que l'utilisation des informations encloses ne viole aucun droit ou aucune garantie implicite de commercialisation ou d'aptitude à un objet particulier.

Propriété intellectuelle

L'IETF ne prend pas position sur la validité et la portée de tout droit de propriété intellectuelle ou autres droits qui pourraient être revendiqués au titre de la mise en œuvre ou l'utilisation de la technologie décrite dans le présent document ou sur la mesure dans laquelle toute licence sur de tels droits pourrait être ou n'être pas disponible ; pas plus qu'elle ne prétend avoir accompli aucun effort pour identifier de tels droits. Les informations sur les procédures de l'ISOC au sujet des droits dans les documents de l'ISOC figurent dans les BCP 78 et BCP 79.

Des copies des dépôts d'IPR faites au secrétariat de l'IETF et toutes assurances de disponibilité de licences, ou le résultat de tentatives faites pour obtenir une licence ou permission générale d'utilisation de tels droits de propriété par ceux qui mettent en œuvre ou utilisent la présente spécification peuvent être obtenues sur le répertoire en ligne des IPR de l'IETF à <http://www.ietf.org/ipr>.

L'IETF invite toute partie intéressée à porter son attention sur tous copyrights, licences ou applications de licence, ou autres droits de propriété qui pourraient couvrir les technologies qui peuvent être nécessaires pour mettre en œuvre la présente norme. Prière d'adresser les informations à l'IETF à ietf-ipr@ietf.org.

Remerciement

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