Groupe de travail Réseau
**Request for Comments : 5475**
Catégorie : Sur la voie de la normalisation

T. Zseby, Fraunhofer FOKUS
M. Molina, DANTE
N. Duffield, AT&T Labs - Research
S. Niccolini, NEC Europe Ltd.
F. Raspall, EPSC-UPC

Traduction Claude Brière de L'Isle
mars 2009

# Techniques d'échantillonnage et de filtrage pour le choix de paquet IP

**Statut du présent mémoire**

Le présent document spécifie un protocole de l'Internet sur la voie de la 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.



**Résumé**

Le présent document décrit les techniques d'échantillonnage et de filtrage pour le choix de paquet IP. Il fait une catégorisation des schémas et définit les paramètres nécessaires pour décrire les schémas de sélection les plus courants. De plus, il montre comment ces techniques peuvent être combinées pour construire des sélecteurs de paquet plus élaborés. Le document donne les bases de la définition des modèles d'information pour les techniques de choix de configuration dans les processus de mesure et pour rapporter la technique utilisée à un collecteur.

## Table des matières

## 1. Introduction

Il y a deux principaux pilotes pour l'évolution des infrastructures de mesures et leurs technologies sous-jacentes. D'abord, les débits de données des réseaux augmentent, avec une croissance concomitante des données de mesure. Ensuite, la croissance est constituée par la demande des applications fondées sur la mesure pour des mesures de trafic d'une granularité de plus en plus fine. Les appareils qui effectuent les mesures exigent des capacités de mesure de plus en plus sophistiquées et consommatrices de ressources, incluant la capture des champs d'en-tête de paquet ou même de parties de la charge utile, et la classification pour l'analyse de flux. Tous ces facteurs peuvent conduire à une quantité écrasante de données de mesure, résultant en de fortes demandes de ressources de mesure, mémorisation, transfert, et traitement ultérieur.

La collecte soutenue de trafic du réseau au débit de ligne peut être effectuée par un matériel de mesure spécialisé. Cependant, le coût du matériel et l'infrastructure de mesures requis pour accommoder les mesures empêchent cette approche d'être généralisée. Une certaine forme de réduction des données au point de mesure est plutôt nécessaire.

Cela peut être réalisé par un choix intelligent de paquets par l'échantillonnage ou le filtrage. Une autre façon de réduire la quantité de données est d'utiliser des techniques d'agrégation (non traitées dans ce document). La motivation de l'échantillonnage est de choisir un sous ensemble représentatif des paquets qui permette de former des estimations précises des propriétés du trafic non échantillonné. La motivation du filtrage est de supprimer tous les paquets qui ne sont pas intéressants. L'agrégation combine les données et permet une vue compacte pré-définie du trafic. Des exemples d'applications qui bénéficient de la sélection de paquet sont données dans la [RFC5474]. Les techniques d'agrégation sortent du domaine d'application du présent document.

### 1.1 Conventions utilisées dans ce document

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

## 2. Vue d'ensemble des documents PSAMP

Le présent document fait partie d'une série de documents du groupe PSAMP.

[RFC5474] : "Cadre de travail pour la sélection et le rapport de paquet" décrit le cadre de PSAMP pour que les éléments de réseau choisissent des sous ensembles de paquets par des méthodes statistiques et autres, et pour exporter un flux de rapports sur les paquets choisis à un collecteur.

RFC 5475 (ce document) : "Techniques d'échantillonnage et de filtrage pour la sélection de paquets IP" décrit l'ensemble

de techniques de sélection de paquets prise en charge par PSAMP.

[RFC5476] : "Spécification du protocole d'échantillonnage de paquet (PSAMP)" spécifie l'exportation des informations de paquet d'un processus d'exportation PSAMP à un processus de collecte PSAMP.

[RFC5477] : "Modèle d'information pour l'exportation d'échantillonnage de paquet" définit un modèle d'information et de données pour PSAMP.

## 3. Terminologie

La terminologie de PSAMP définie ici est pleinement cohérente avec tous les termes mentionnés dans la [RFC5474] mais inclut des termes supplémentaires nécessaires pour la description des méthodes de choix de paquet. Une vue d'ensemble de l'architecture et des configurations possibles des éléments de PSAMP se trouve dans la [RFC5474]. La terminologie de PSAMP vise aussi à la cohérence avec les termes utilisés dans la [RFC3917]. Les relations entre les termes de PSAMP et de IPFIX sont décrites dans la [RFC5474].

Dans les documents PSAMP, tous les termes définis pour PSAMP sont écrits avec une majuscule. Le présent document utilise la même convention *(seulement pour le texte anglais d'origine)*.

### 3.1 Points d'observation, flux de paquets, et contenu de paquet

Point d'observation [RFC5101] : c'est une localisation dans le réseau où les paquets peuvent être observés. Des exemples incluent :
(i) une ligne à laquelle est attachée une sonde ;
(ii) un support partagé, comme un LAN fondé sur Ethernet ;
(iii) un seul accès d'un routeur, ou un ensemble d'interfaces (physiques ou logiques) d'un routeur ;
(iv) un sous système incorporé de mesures au sein d'une interface.
Noter qu'un point d'observation peut être un sur-ensemble de plusieurs autres points d'observation. Par exemple, un point d'observation peut être une carte de ligne entière. Cela serait le sur-ensemble des points d'observation individuels aux interfaces de la carte de ligne.

Flux de paquets observé : c'est l'ensemble de tous les paquets observés au point d'observation.

Flux de paquets : ensemble de paquets provenant du flux de paquets observé qui s'écoule après un point spécifié dans le processus de mesure. Un exemple de flux de paquets est le résultat du processus de sélection. Noter que les paquets choisis dans un flux, par exemple, par échantillonnage, ne possèdent pas nécessairement une propriété qui peut les distinguer des paquets qui n'ont pas été choisis. Pour cette raison, le terme de "flux" est préféré à celui de "courant", qui est défini comme un ensemble de paquets avec des propriétés communes [RFC3917].

Contenu de paquet : il note l'union de l'en-tête de paquet (qui inclut des champs d'en-tête de couche de liaison, de couche réseau, et autre couche d'encapsulation) et la charge utile du paquet. À certains points d'observation, les informations d'en-tête de couche de liaison peuvent n'être pas disponibles.

### 3.2 Processus de choix

Processus de choix : il prend en entrée le flux de paquets observé et choisit un sous ensemble de ce flux comme résultat.

État de sélection : un processus de choix peut maintenir des informations d'état à utiliser par le processus de choix. À un moment donné, l'état de sélection peut dépendre des paquets observés à ce moment ou avant, et d'autres variables. Des exemples incluent :
(i) les numéros de séquence des paquets à l'entrée des sélecteurs ;
(ii) un horodatage de l'observation du paquet au point d'observation ;
(iii) des itérateurs pour les générateurs de nombres pseudo aléatoires ;
(iv) des valeurs de hachage calculées durant la sélection ;
(v) des indicateurs de si le paquet a été choisi par un sélecteur donné.

Les processus de choix peuvent changer des portions de l'état de sélection par suite du traitement d'un paquet. L'état de sélection pour un paquet est de refléter l'état après le traitement du paquet.

Sélecteur : un sélecteur définit quelle sorte d'action effectue un processus de choix sur un seul paquet de son entrée. Si il est choisi, le paquet devient un élément du flux de paquets de sortie.
Le sélecteur peut utiliser les informations suivantes pour déterminer si un paquet est choisi :
(i) le contenu du paquet ;
(ii) des informations déduites du traitement du paquet au point d'observation ;
(iii) tout état de sélection qui peut être maintenu par le processus de choix.

Sélecteur composite : c'est une composition ordonnée de sélecteurs, dans laquelle le flux de paquets de sortie provenant d'un sélecteur forme le flux de paquets d'entrée du sélecteur suivant.

Sélecteur primitif : un sélecteur est primitif si il n'est pas un sélecteur composite.

Séquence de sélection : à partir de tous les paquets observés à un point d'observation, seuls quelques paquets sont choisis par un ou plusieurs sélecteurs. La séquence de sélection est une valeur unique par domaine d'observation décrivant le point d'observation et les identifiants de sélecteur à travers lesquels les paquets sont choisis.

### 3.3 Rapports

Rapports de paquets : ils comprennent un sous ensemble configurable d'une entrée de paquets au processus de sélection, incluant le contenu de paquet, les informations relatives à son traitement (par exemple, l'interface de sortie) et son état de sélection associé (par exemple, un hachage du contenu du paquet).

Interprétation de rapport : elle comporte des informations subsidiaires, relatives à un ou plusieurs paquets, qui sont utilisées pour l'interprétation de leurs rapports de paquet. Des exemples incluent des paramètres de configuration du processus de choix.

Flux de rapports : c'est le résultat du processus de mesures, comprenant deux types distinctifs d'informations : les rapports de paquet et l'interprétation de rapport.

### 3.4 Processus de mesures

Un processus de mesure choisit les paquets à partir du flux de paquets observé en utilisant un processus de choix, et produit comme résultat un flux de rapport concernant les paquets choisis.

Le processus de mesure PSAMP peut être vu comme l'analogue du processus de mesure IPFIX [RFC5101], qui produit des enregistrements de flux comme résultat, avec la différence que le processus de mesure PSAMP contient toujours un processus de choix. La relation entre PSAMP et IPFIX est décrite dans les [RFC5477] et [RFC5474].

### 3.5 Processus d'exportation

Processus d'exportation : il envoie, sous la forme de paquets d'exportation, le résultat d'un ou plusieurs processus de mesure à un ou plusieurs collecteurs.

Paquet d'exportation : combinaison d'interprétations de rapport et/ou d'un ou plusieurs rapports de paquet qui sont groupés par le processus d'exportation dans un paquet d'export à exporter à un collecteur.

### 3.6 Appareil PSAMP

Appareil PSAMP : appareil qui héberge au moins un point d'observation, un processus de mesure (qui inclut un processus de choix) et un processus d'exportation. Normalement, le ou les points d'observation, le ou les processus de mesure, et le ou les processus d'exportation correspondants sont colocalisés dans cet appareil, par exemple, dans un routeur.

### 3.7 Collecteur

Collecteur : un collecteur reçoit un flux de rapport exporté par un ou plusieurs processus d'exportation. Dans certains cas, l'hôte des processus de mesure et/ou d'exportation peut aussi servir de collecteur.

### 3.8 Méthodes de choix

Filtrage : un filtre est un sélecteur qui choisit un paquet de façon déterministe sur la base du contenu du paquet, ou de son traitement, ou de leurs fonctions qui se produisent dans l'état de sélection. Deux exemples sont :
(i) Le filtrage par correspondance de propriété : un paquet est choisi si un champ spécifique du paquet est égal à une valeur prédéfinie.
(ii) Choix fondé sur le hachage : une fonction de hachage est appliquée au contenu du paquet, et le paquet est choisi si le résultat tombe dans la gamme spécifiée.

Échantillonnage : un sélecteur qui n'est pas un filtre est appelé une opération d'échantillonnage. Cela reflète la notion intuitive que si le choix d'un paquet ne peut pas être déterminé à partir de son seul contenu, il doit y avoir lieu à un certain type d'échantillonnage. Les opérations d'échantillonnage peuvent être divisées en deux sous-types :
(i) Échantillonnage indépendant du contenu, qui n'utilise pas le contenu du paquet pour prendre les décisions d'échantillonnage. Des exemples incluent l'échantillonnage systématique, et l'échantillonnage pseudo aléatoire uniforme conduit par un nombre pseudo aléatoire dont la génération est indépendante du contenu du paquet. Noter que dans l'échantillonnage indépendant du contenu, il n'est pas nécessaire d'accéder au contenu de paquet pour prendre la décision de sélection.
(ii) Échantillonnage dépendant du contenu, dans lequel le contenu du paquet est utilisé pour prendre les décisions de choix. Une application est un choix pseudo aléatoire conformément à une probabilité qui dépend du contenu d'un champ du paquet, par exemple, l'échantillonnage de paquets avec une probabilité dépendant de leur numéro d'accès TCP/UDP. Noter que cela n'est pas un filtre.

Domaine de hachage : c'est un sous ensemble du contenu de paquet et du traitement de paquet, vu comme une chaîne de N bits pour un entier positif N.

Gamme de hachage : c'est un ensemble de chaînes de M bits pour un entier positif M qui définit la gamme de valeurs que peut prendre le résultat de l'opération de hachage.

Fonction de hachage : elle définit une transposition déterministe du domaine de hachage à la gamme de hachage.

Gamme de choix de hachage : c'est un sous ensemble de la gamme de hachage. Le paquet est choisi si l'action de la fonction de hachage sur le domaine de hachage pour le paquet donne un résultat dans la gamme de choix de hachage.

Choix fondé sur le hachage : c'est un filtrage spécifié par un domaine de hachage, une fonction de hachage, une gamme de hachage, et une gamme de choix de hachage.

Choix approximatif : les sélecteurs dans toutes les catégories ci-dessus peuvent être approximés par des opérations dans la même catégorie ou une autre pour les besoins de la mise en œuvre. Par exemple, l'échantillonnage pseudo aléatoire uniforme peut être approximé par le choix fondé sur le hachage, en utilisant une fonction de hachage et un domaine de hachage convenables. Dans ce cas, la précision de l'approximation dépend du choix de la fonction de hachage et du domaine de hachage.

Population : c'est un flux de paquets ou un sous ensemble d'un flux de paquets. Une population peut être considérée comme un ensemble de base à partir duquel les paquets sont choisis. Un exemple est "tous les paquets dans le flux de paquets observés qui sont observés dans un intervalle de temps spécifié".

Taille de population : c'est le nombre total de paquets dans la population.

Taille d'échantillon : c'est le nombre de paquets choisis dans la population par un sélecteur.

Fraction configurée de choix : c'est le ratio attendu de la taille d'échantillon sur la taille de population, telle que fondée sur les paramètres de choix configurés.

Fraction de choix atteinte : c'est le ratio de la taille d'échantillon réelle sur la taille de population. Pour certaines méthodes d'échantillonnage, la fraction de choix atteinte peut différer de la fraction de choix configurée du fait, par exemple, de la variabilité statistique inhérente des décisions d'échantillonnage d'échantillonnage probabiliste et de choix fondé sur le hachage. Néanmoins, pour les grandes tailles de population et des sélecteurs configurés de façon appropriée, la fraction de choix atteinte approche généralement la fraction configurée de choix.

## 4. Catégorisation des techniques de choix de paquet

Les techniques de choix de paquet génèrent un sous ensemble de paquets dans un flux de paquets observés à un point d'observation. On distingue l'échantillonnage et le filtrage.

L'échantillonnage vise au choix d'un sous ensemble de paquets représentatif. Le sous ensemble est utilisé pour déduire des connaissances sur l'ensemble complet de paquets observés sans les traiter tous. Le choix peut dépendre de la position du paquet, et/ou du contenu du paquet, et/ou de décisions (pseudo) aléatoires.

Le filtrage choisit un sous ensemble avec des propriétés communes. C'est utilisé si seulement un sous ensemble des paquets est intéressant. Les propriétés peuvent être directement déduites du contenu du paquet, ou dépendre du traitement donné par le routeur au paquet. Le filtrage est une opération déterministe. Il dépend du contenu du paquet ou du traitement du routeur. Il ne dépend jamais de la position du paquet ou de décisions (pseudo) aléatoires.

Noter qu'une technique courante pour choisir les paquets est de calculer une fonction de hachage sur certains bits de l'en-tête de paquet et/ou son contenu et de le choisir si la valeur de hachage tombe dans la gamme de choix de hachage. Comme le hachage est une opération déterministe sur le contenu de paquet, c'est une technique de filtrage selon notre catégorisation. Néanmoins, les fonctions de hachage sont parfois utilisées pour émuler l'échantillonnage aléatoire. Selon les bits d'entrée choisis, la fonction de hachage, et la gamme de choix de hachages, cette technique peut être utilisée pour émuler le choix aléatoire de paquets avec une certaine probabilité p. C'est aussi une technique puissante pour choisir de façon cohérente le même sous ensemble de paquets à plusieurs points d'observation [DuGr00].

Le tableau suivant donne une vue d'ensemble des schémas décrits dans ce document et leur catégorisation. X signifie que la caractéristique s'applque au schéma de sélection. (X) note les schémas pour lesquels des variantes dépendantes du contenu et indépendantes du contenu existent. Par exemple, le filtrage par correspondance de propriété est normalement fondé sur le contenu de paquet et est donc dépendant du contenu. Mais comme expliqué au paragraphe 6.1, il peut aussi dépendre de l'état du routeur et va alors être indépendant du contenu. Il peut être facilement vu que seuls les schémas avec les deux propriétés, dépendance du contenu et choix déterministe, sont considérés comme des filtres.

| **Schéma de sélection** | **Sélection déterministe** | **Dépendant du contenu** | **Catégorie** |
|---|---|---|---|
| Systématique fondé sur le compte | X | - | Échantillonnage |
| Systématique fondé sur le temps | X | - | Échantillonnage |
| Aléatoire n parmi N | - | - | Éhantillonnage |
| Aléatoire probabiliste uniforme | - | - | Échantillonnage |
| Aléatoire probabiliste non uniforme | - | (X) | Échantillonnage |
| Aléatoire état de flux non uniforme | - | (X) | Échantillonnage |
| Filtrage de correspondance de propriété | X | X) | Filtrage |
| Fonction de hachage | X | X | Filtrage |

La catégorisation qu'on vient d'introduire est surtout utile pour la définition d'un modèle d'information décrivant les sélecteurs de primitive. Des techniques de choix plus complexes peuvent être décrites par la composition d'opérations d'échantillonnage et de filtrage en cascade. Par exemple, un choix de paquet qui pondère la probabilité de choix sur la base de la longueur de paquet peut être décrite comme une cascade d'un schéma de filtrage et d'échantillonnage. Cependant, cette approche descriptive n'est pas destinée à être rigide : si une pratique courante et consolidée de sélection se révèle trop complexe pour être décrite comme une composition des blocs de construction mentionnés, une description ad hoc peut être spécifiée à la place et ajoutée comme nouveau schéma au modèle d'information.

## 5. Échantillonage

Le déploiement des techniques d'échantillonnage vise à la fourniture d'informations sur une caractéristique spécifique de la population parente à un moindre coût que ce que demanderait un recensement complet. Afin de planifier une stratégie d'échantillonnage convenable, il est donc crucial de déterminer à l'avance le type d'informations nécessaire et le degré désiré de précision.

Tout d'abord, il est important de connaître le type de métrique qui devrait être estimée. La métrique intéressante peut aller du simple compte de paquets [JePP92] jusqu'à l'estimation des distributions complètes des caractéristiques de flux (par exemple, les tailles de paquets) [ClPB93].

Ensuite, la précision requise des informations et avec cela, la confiance qui est visée, devrait être connue à l'avance. Par exemple, pour la comptabilité fondée sur l'usage, la confiance requise pour l'estimation des compteurs de paquets peut dépendre de la valeur monétaire qui correspond au transfert d'un paquet. Cela signifie qu'une confiance supérieure pourrait être requise pour des flux de paquets coûteux (par exemple, le service IP premium) que pour des flux moins coûteux (par exemple, au mieux). Les exigences de précision pour valider une qualité acceptée précédemment peuvent aussi varier extrêmement avec les demandes des consommateurs. Ces exigences sont généralement déterminées par des accords de niveau de service (SLA, *Service Level Agreement*).

La méthode d'échantillonnage et les paramètres utilisés doivent être clairement communiqués à toutes les applications qui utilisent les données de mesure. C'est seulement avec cette connaissance qu'une interprétation correcte des résultats de mesure peut être assurée.

Les méthodes d'échantillonnage peuvent être caractérisées par l'algorithme d'échantillonnage, le type de déclencheur utilisé pour débuter l'intervalle d'échantillonnage, et la longueur de l'intervalle d'échantillonnage. Ces paramètres sont décrits en détails ici. L'algorithme d'échantillonnage décrit le processus de base du choix des échantillons. En accord avec [AmCa89] et [ClPB93], on définit le processus de base d'échantillonnage.

### 5.1 Échantillonnage systématique

L'échantillonnage systématique décrit le processus de choix des points de départ et la durée des intervalles de choix en accord avec une fonction déterministe. Ce peut être par exemple le choix périodique de tout kième élément d'une trace mais aussi la sélection de tous les paquets qui arrivent à un instant prédéfini. Même si le processus de sélection ne suit pas une fonction périodique (par exemple, si le temps entre les intervalles d'échantillonnage varie dans le temps) on considère cela comme un échantillonnage systématique tant que le choix est déterministe.

L'utilisation de l'échantillonnage systématique implique toujours le risque de biaiser le résultat. Si la systématique dans le processus d'échantillonnage ressemble à celle du processus stochastique observé (l'occurrence des caractéristiques intéressantes dans le réseau) il y a une forte probabilité que l'estimation soit biaisée. La systématique du processus observé pourrait n'être pas connue à l'avance.

On considère seulement ici les schémas à espacement égal, où les déclencheurs de l'échantillonnage sont périodiques, soit dans le temps, soit en compte de paquets. Tous les paquets arrivant dans un intervalle de sélection (en temps ou compte de paquets) au delà du déclencheur sont retenus.

Systématique fondé sur le compte : dans l'échantillonnage systématique fondé sur le compte, les déclencheurs de début et d'arrêt pour l'intervalle d'échantillonnage sont définis en accord avec la position spatiale du paquet (compte de paquets).

Systématique fondé sur le temps : dans l'échantillonnage systématique fondé sur le temps, des déclencheurs de début et d'arrêt fondés sur le temps sont utilisés pour définir les intervalles d'échantillonnage. Sont choisis tous les paquets qui arrivent au point d'observation dans les intervalles de temps définis par les déclencheurs de début et d'arrêt (c'est-à-dire, l'heure d'arrivée du paquet est postérieure à l'heure de début et antérieure à l'heure d'arrêt).

Les deux schémas sont des schémas de choix indépendants du contenu. Les sélecteurs déterministes dépendants du contenu sont catégorisés comme des filtres.

### 5.2 Échantillonnage aléatoire

L'échantillonnage aléatoire choisit les points de début des intervalles d'échantillonnage en accord avec un processus aléatoire. Le choix des éléments est une expérience indépendante. Avec cela, des estimations non biaisées peuvent être réalisées. À la différence de l'échantillonnage systématique, l'échantillonnage aléatoire exige la génération de nombres aléatoires. On peut différencier les deux méthodes d'échantillonnage aléatoire en échantillonnage de n parmi N et en échantillonnage probabiliste.

#### 5.2.1 Échantillonnage de n parmi N

Dans l'échantillonnage de n parmi N, n éléments sont choisis dans la population parente qui consiste en N éléments. Un exemple serait de générer n différents nombres aléatoires dans la gamme [1,N] et de choisir tous les paquets qui ont une position de paquet égale à un des nombres aléatoires. Pour cette sorte d'échantillonnage, la taille de l'échantillon n est fixée.

### 5.2.2 Échantillonnage probabiliste

Dans l'échantillonnage probabiliste, la décision de choisir ou non un élément est prise en accord avec une probabilité de choix prédéfinie. Un exemple serait jeter une pièce de monnaie pour chaque paquet et de choisir tous les paquets pour lesquels la pièce était "face". Pour cette sorte d'échantillonnage, la taille d'échantillon peut varier pour les différents essais. La probabilité de choix ne doit pas nécessairement être la même pour chaque paquet. Donc, on distingue l'échantillonnage probabiliste uniforme (avec la même probabilité de choix pour tous les paquets) et l'échantillonnage probabiliste non uniforme (où la probabilité de choix peut varier pour les différents paquets).

#### 5.2.2.1 Échantillonnage probabiliste uniforme

Pour l'échantillonnage probabiliste uniforme, les paquets sont choisis indépendamment avec une probabilité p uniforme. Cet échantillonnage peut être fondé sur le compte, et est parfois appelé un échantillonnage aléatoire géométrique, parce que la différence de compte entre les paquets choisis successifs est une variable aléatoire indépendante avec une distribution géométrique de moyenne 1/p. Un échantillonnage aléatoire exponentiel analogique fondé sur le temps a une distribution exponentielle du temps entre les déclencheurs.

Les deux échantillonnages aléatoire géométrique et exponentiel sont des exemples de ce qui est appelé l'échantillonnage additif aléatoire, défini comme l'échantillonnage où les intervalles ou comptes entre échantillons successifs sont des variables aléatoires indépendantes identiquement distribuées.

#### 5.2.2.2 Échantillonnage probabiliste non uniforme

C'est une variante de l'échantillonnage probabiliste dans lequel les probabilités de l'échantillonnage peuvent dépendre de l'entrée du processus de sélection. Cela peut être utilisé pour pondérer les probabilités de l'échantillonnage afin, par exemple, de gonfler les chances des paquets de l'échantillonnage qui sont rares mais réputés importants. Des estimateurs non biaisés pour des statistiques quantitatives sont récupérés en re-normalisant les valeurs des échantillons ; voir [HT52].

#### 5.2.2.3 Échantillonnage non uniforme dépendant de l'état du flux

Un autre type d'échantillonnage qui peut être classé comme probabiliste non uniforme est en relation étroite avec le concept de flux défini dans la [RFC3917], et il est seulement utilisé conjointement avec une fonction de surveillance de flux (le processus de mesures IPFIX). Les paquets sont choisis selon un état de sélection. L'important ici est que l'état de sélection est déterminé aussi par l'état du flux auquel le paquet appartient et/ou par l'état des autres flux actuellement surveillés par la fonction de surveillance de flux associée. Un exemple d'un tel algorithme est la méthode "échantillonne et garde" décrite dans [EsVa01] :

- Si un paquet entre en compte pour un enregistrement de flux qui existe déjà dans le processus d'enregistrement de flux IPFIX, il est choisi (c'est-à-dire, l'enregistrement de flux est mis à jour).
- Si un paquet n'entre en compte pour aucun enregistrement de flux existant, il est choisi avec une probabilité p. Si il a été choisi, un nouvel enregistrement de flux doit être créé.

Un autre algorithme qui entre dans la catégorie de l'échantillonnage non uniforme dépendant de l'état du flux est décrit dans [Moli03].

Ce type d'échantillonnage est dépendant du contenu parce que l'identification du flux auquel le paquet appartient exige d'analyser une partie du contenu du paquet. Si le paquet est choisi, alors il est passé en entrée à la fonction de surveillance IPFIX (qui est appelée "Local Export" dans la [RFC5474]). Choisir le paquet selon l'état d'une antémémoire de flux est utile quand les ressources de mémoire de la fonction de surveillance de flux sont restreintes (c'est-à-dire, quand il n'y a pas assez de place pour garder tous les flux dont la surveillance a été programmée).

#### 5.2.2.4 Configuration d'échantillonnage non uniforme probabiliste et dépendant de l'état du flux

De nombreuses méthodes spécifiques différentes peuvent être groupées sous les termes d'échantillonnage non uniforme probabiliste et d'état de flux. Selon le but de l'échantillonnage et le schéma mis en œuvre, un nombre et type différents de paramètres d'entrée sont nécessaires pour configurer un tel schéma.

Il existe des propositions concrètes pour de telles méthodes dans la communauté de la recherche (par exemple, [EsVa01], [DuLT01], [Moli03]). Certaines de ces propositions sont encore peu avancées et ont besoin d'investigations supplémentaires pour prouver leur utilité et leur applicabilité. Notre but n'est pas d'indiquer des préférences parmi ces méthodes. On va plutôt seulement décrire ici les méthodes de base et laisser la spécification des schémas explicites et leurs paramètres aux fabricants (par exemple, comme extension du modèle d'information).

# 6. Filtrage

Le filtrage est le choix déterministe de paquets sur la base du contenu du paquet, du traitement du paquet au point d'observation, ou de fonctions déterministes de leur occurrence dans l'état de sélection. Le paquet est choisi si ces quantités entrent dans une gamme spécifiée. Le rôle du filtrage, comme le mot le suggère lui-même, est de séparer tous les paquets qui ont une certaine propriété de ceux qui ne l'ont pas. Une caractéristique distinctive de l'échantillonnage est que la décision du choix ne dépend pas de la position du paquet dans le temps ou dans l'espace, ou d'un processus aléatoire.

On identifie et décrit les deux techniques de filtrage dans les paragraphes qui suivent.

## 6.1 Filtrage de correspondance de propriété

Avec cette méthode de filtrage, un paquet est sélectionné si des champs spécifiques du paquet et/ou des propriétés de l'état du routeur sont égaux à une valeur prédéfinie. Les champs de filtre possibles sont tous des attributs de flux IFIX spécifiés dans la [RFC5102]. D'autres champs peuvent être définis en proposant de nouveaux éléments d'information ou en définissant des extensions spécifiques de fabricant.

Un paquet est choisi si Field=Value. Les gabarits et les gammes ne sont pris en charge que dans la mesure où la [RFC5102] les permet, par exemple, en fournissant des champs explicites comme les gabarits réseau des adresses de source et destination.

Des opérations ET sont possibles en enchaînant des filtres, produisant donc une opération de choix composite. Dans ce cas, l'ordre dans lequel le filtrage se produit est implicitement défini (les filtres les plus externes viennent après les filtres les plus internes). Cependant, tant que l'enchaînement est seulement sur les filtres, le résultat de la cascade de filtres est indépendant de l'ordre, mais l'ordre peut être important pour les besoins de la mise en œuvre, car le premier filtre va devoir travailler à un plus fort taux. Dans tous les cas, une mise en œuvre n'est pas contrainte à l'égard de l'ordre des filtres, tant que le résultat est le même, et elle peut même effectuer le filtrage composite en une seule étape.

Les opérations OU ne sont pas prises en charge dans ce modèle de base. Des filtres plus sophistiqués (par exemple, qui supportent des gabarits binaires, des gammes, ou des opérations OU) peuvent être réalisés comme schémas spécifiques de fabricant.

Tous les attributs de flux IPFIX définis dans la [RFC5102] peuvent être utilisés pour le filtrage par correspondance de propriété. D'autres éléments d'information peuvent être facilement définis. Les opérations de correspondance de propriété devraient être disponibles pour les différentes portions de protocole de l'en-tête de paquet :

(i) en-tête IP (excluant les options dans IPv4, les champs d'en-tête en pile dans IPv6)
(ii) en-tête de protocole de transport (par exemple, TCP, UDP)
(iii) champs d'en-tête d'encapsulation (par exemple, la pile d'étiquettes MPLS, si elle est présente)

Quand l'appareil PSAMP offre le filtrage par correspondance de propriété, et, dans sa capacité normale autre que d'effectuer des fonctions PSAMP, identifie ou traite des informations provenant de IP, du protocole de transport ou de protocoles d'encapsulation, alors les informations devraient être rendues disponibles pour le filtrage. Par exemple, quand un appareil PSAMP achemine sur la base de l'adresse IP de destination, ce champ devrait être rendu disponible pour le filtrage. À l'inverse, un appareil PSAMP qui n'achemine pas n'est pas supposé être capable de localiser une adresse IP dans un paquet, ou la rendre disponible pour le filtrage, bien qu'il puisse le faire.

Comme le chiffrement de paquet dissimule les valeurs réelles de champs chiffrés, le filtrage par correspondance de propriété doit être configurable à ignorer les paquets chiffrés, quand ils sont détectés.

Le processus de choix peut prendre en charge le filtrage fondé sur les propriétés de l'état du routeur :

(i) l'interface d'entrée à laquelle arrive un paquet est égale à une valeur spécifiée
(ii) l'interface de sortie à laquelle un paquet est acheminé est égale à une valeur spécifiée
(iii) le paquet a violé une liste de contrôle d'accès (ACL) sur le routeur
(iv) échec de transmission sur le chemin inverse (RPF, *Reverse Path Forwarding*)
(v) échec du protocole de réservation de ressource (RSVP, *Resource Reservation Protocol*)
(vi) pas de chemin trouvé pour le paquet
(vii) le système autonome (AS) d'origine du protocole de routeur frontière (BGP, *Border Gateway Protocol*) est égal à une valeur spécifiée ou tombe dans une gamme donnée
(viii) l'AS BGP de destination est égal à une valeur spécifiée ou tombe dans une gamme donnée

Les paquets qui correspondent à une condition d'échec de transmission sur le chemin inverse (RPF) sont des paquets pour lesquels le filtrage d'entrée a échoué, comme défini dans la [RFC3704].

Les paquets qui correspondent à une condition d'échec du protocle de réservation de ressource (RSVP) sont des paquets qui ne respectent pas pleinement la spécification de RSVP, comme définie dans la [RFC2205].

Des considérations architecturales de routeur peuvent empêcher certaines informations concernant le traitement du paquet d'être disponibles au débit de ligne pour le choix des paquets. Par exemple, le processus de choix peut n'être pas mis en œuvre dans le chemin rapide qui est capable d'accéder à l'état du routeur au taux de ligne. Cependant, quand le filtrage suit l'échantillonnage (ou une autre opération de sélection) dans un sélecteur composite, le taux du résultat du flux de paquets provenant de l'échantillonneur et entré au filtre peut être suffisamment lent pour que le filtre puisse choisir sur la base de l'état du routeur.

### 6.2 Filtrage fondé sur le hachage

Une fonction de hachage h se transpose en le contenu de paquet c, ou une certaine portion du paquet, sur une gamme de hachage R. Le paquet est choisi si h(c) est un élément de S, qui est un sous ensemble de R appelé la gamme de choix de hachages. Donc, le choix fondé sur le hachage est un cas particulier de filtrage. L'objet est sélectionné si c est dans inv(h(S)). Mais pour les fonctions de hachage désirables, l'image inverse inv(h(S)) va être extrêmement complexe, et donc h ne va pas être exprimable comme, disons, un filtre fondé sur la correspondance de propriété ou une simple combinaison de ceux-ci.

Le choix fondé sur le hachage est principalement utilisé pour réaliser un choix de paquets coordonné. Cela signifie que les mêmes paquets sont choisis à différents points d'observation. Cela est utile par exemple pour observer le chemin (trajectoire) qu'a pris un paquet à travers le réseau ou pour appliquer le choix de paquets à des mesures unidirectionnelles passives.

Une condition préalable pour que cette méthode fonctionne et pour assurer l'interopérabilité est que la même fonction de hachage avec les mêmes paramètres (par exemple, valeur d'entrée) soit utilisée aux points d'observation.

Un choix de paquets cohérent est aussi possible avec le filtrage sur la base de la correspondance de propriété. Néanmoins,le choix fondé sur le hachage peut être utilisé pour approximer un choix aléatoire. Les propriétés statistiques désirées sont discutées au paragraphe 6.2.2.

Dans les paragraphes qui suivent, on donne des exemples d'application de choix de paquets coordonné.

#### 6.2.1 Exemples d'application pour le choix de paquets coordonné

##### 6.2.1.1 Échantillonnage de trajectoire

L'échantillonnage de trajectoire est le choix cohérent d'un sous ensemble de paquets soit à tous les points d'observation d'un ensemble, soit à aucun d'entre eux. L'échantillonnage de trajectoire est réalisé par le choix fondé sur le hachage si tous les points d'observation de l'ensemble utilisent une fonction de hachage, un domaine de hachage, et une gamme de choix, communs. Le domaine de hachage comprend tout ou partie du contenu de paquet qui est invariant le long du chemin du paquet. Des champs comme la durée de vie (TTL) qui sont décrémentés à chaque bond, et le CRC d'en-tête [RFC1624], qui est recalculé à chaque bond, sont donc exclus du domaine de hachage. Le domaine de hachage doit être plus large que juste une clé de flux, si les paquets sont à choisir de façon quasi aléatoire au sein des flux.

La trajectoire (ou chemin) suivie par un paquet est reconstruite à partir des rapports PSAMP sur lui qui atteignent un collecteur. Les rapports sur un paquet donné qui ont pour origine les différents points d'observation sont associés en faisant correspondre une étiquette dans les rapports. L'étiquette peut comprendre la portion invariante du contenu de paquet rapporté, ou éventuellement un résumé du contenu invariant de paquet, qui est inséré dans le rapport de paquet au point d'observation. Un tel résumé peut être construit en appliquant une seconde fonction de hachage au contenu invariant de paquet. La reconstruction des trajectoires et les méthodes pour traiter de possibles ambiguïtés dues à des collisions d'étiquettes (étiquettes identiques rapportées pour des paquets différents) et d'éventuelles pertes de rapports dans la transmission sont traitées dans [DuGr00], [DuGG02], et [DuGr04].

Les applications d'échantillonnage de trajectoire incluent (i) l'estimation de la matrice des chemins du réseau, c'est-à-dire, les intensités de trafic selon les chemins du réseau, classées par clés de flux ; (ii) la détection de boucles d'acheminement,

comme indiqué par des intersections de trajectoires ; (iii) des mesures de performances passives : des trajectoires qui se terminent prématurément indiquent des pertes de paquet, le délai unidirectionnel de paquet peut être déterminé si les rapports incluent des horodatages (synchronisés) des arrivées de paquet au point d'observation ; et (iv) le traçage des attaques du réseau, du chemin réel pris par les paquets d'attaque avec des adresses de source usurpées.

#### 6.2.1.2 Mesures unidirectionnelles passives

Le choix de paquets coordonné peut être appliqué par exemple aux mesures de délai unidirectionnel afin de réduire les ressources nécessaires. Dans les mesures de délai unidirectionnel, les paquets sont collectés à différents points d'observation dans le réseau. Un résumé de paquet est généré pour chaque paquet pour aider à identifier le paquet. Le résumé de paquet et l'heure d'arrivée au point d'observation sont rapportés à un processus qui calcule le délai. Le délai est calculé en soustrayant l'heure d'arrivée du même paquet aux points d'observation (par exemple, [ZsZC01]). Avec des débits de données élevés, capturer tous les paquets peut exiger une certaine quantité de ressources pour la mémorisation, le transfert, et le traitement. Pour réduire la consommation de ressources, des méthodes de choix de paquet peuvent être appliquées. Mais pour de telles techniques de choix, on doit s'assurer que les mêmes paquets sont collectés aux différents points d'observation. Le choix fondé sur le hachage fournit cette caractéristique.

#### 6.2.1.3 Génération de nombres pseudo aléatoires

Bien que des générateurs de nombres pseudo aléatoires avec des propriétés bien comprises aient été développés, cela peut ne pas être la méthode choisie dans des situations où les ressources de calcul sont rares. Une solution de remplacement convenable est d'utiliser des fonctions de hachage de contenu de paquet comme source d'aléa. Le hachage (convenablement re-normalisé) est une variable pseudo aléatoire dans l'intervalle [0,1]. D'autres schémas peuvent utiliser les champs du paquet dans des itérateurs de nombres pseudo aléatoires. Cependant, les propriétés statistiques d'une loi idéale de choix de paquet (comme l'échantillonnage indépendant pour des paquets différents, ou l'indépendance au contenu de paquet) peuvent n'être pas rendues exactement par une mise en œuvre, mais seulement de façon approximative.

L'utilisation du contenu de paquet pour générer des variables pseudo aléatoires partage avec l'échantillonnage probabiliste non uniforme (voir au paragraphe 5.2.2.2 ci-dessus) la propriété que les décisions de choix dépendent du contenu de paquet. Cependant, il y a une différence fondamentale entre les deux. Dans le premier cas, le contenu détermine des variables pseudo aléatoires. Dans le dernier cas, le contenu détermine seulement les probabilités de choix : la sélection pourrait alors se faire, par exemple, en utilisant les variables aléatoires obtenues par un générateur de nombres pseudo aléatoires indépendant.

### 6.2.2 Propriétés désirées des fonctions de hachage

On formule ici les propriétés désirées des fonctions de hachage. Pour cela, on doit distinguer si une fonction de hachage est utilisée pour le choix de paquet ou juste comme résumé de paquet. L'objet principal du présent document est le choix de paquets. Néanmoins, on fournit aussi des exigences pour l'utilisation des fonctions de hachage comme résumé de paquet.

Tout d'abord, on doit définir les champs de résultat convenables provenant du paquet. En accord avec [DuGr00], le champ d'entrée devrait être :
- invariant sur le chemin
- variable entre les paquets

C'est seulement si les champs d'entrée sont les mêmes aux différents points d'observation qu'il est possible de reconnaître le paquet. Les champs d'entrée devraient être variables parmi les paquets afin de répartir le résultat du hachage sur la gamme de sélection.

#### 6.2.2.1 Exigences pour le choix de paquet

En accord avec les considérations de [MoND05] et [Henk08], on définit les propriétés désirées suivantes des fonctions de hachage utilisées pour le choix de paquet :

(i) Vitesse : la fonction de hachage doit être appliquée à chaque paquet qui traverse le point d'observation. Donc, elle doit être rapide afin de s'accommoder des hauts débits de paquets. Idéalement, l'opération de hachage ne devrait pas influencer les performances de l'appareil PSAMP.

(ii) Uniformité : la fonction de hachages h devrait avoir de bonnes propriétés de mixage, en ce sens que de petits changements dans l'entrée (par exemple, la variation d'un seul bit) causent de grands changements dans le résultat (changement de nombreux bits). Alors tout bloc local de valeurs de c est largement étalé sur R par h, et ainsi la

distribution de h(c) est bien uniforme même si la distribution de c ne l'est pas. Alors la fraction de sélection atteinte devient proche de la fraction de sélection configurée (#S/#R) qui peut être réglée par le choix de S.

(iii) Non biaisé : La décision de choix devrait être aussi indépendante des attributs de paquet que possible. L'ensemble des paquets choisi ne devrait pas être biaisé à l'égard d'un type spécifique de paquets.

(iv) Représentativité de l'échantillon : l'échantillon devrait être aussi représentatif que possible pour le trafic observé.

(v) Non linéarité : la fonction ne devrait pas être linéaire. Cela augmente les propriétés de mixage (critère d'uniformité). En plus de cela, cela diminue la prévisibilité du résultat et donc la vulnérabilité aux attaques.

(vi) Robustesse contre les vulnérabilités : la fonction de hachage devrait être robuste contre les attaques. Les vulnérabilités potentielles sont décrites au paragraphe 6.2.3.

#### 6.2.2.2 Exigences pour le résumé de paquet

Pour résumer le contenu de paquet à inclure dans une étiquette de rapport, la propriété la plus importante est une faible fréquence de collision. Une exigence secondaire est la capacité d'accepter des entrées de longueur variable, afin de permettre l'inclusion d'une quantité maximale de paquets comme entrées. La vitesse d'exécution est d'importance secondaire, car le résumé a seulement besoin d'être formé à partir des paquets choisis.

### 6.2.3 Considérations sur la sécurité des fonctions de hachage

Un problème du choix fondé sur le hachage est si un grand ensemble de paquets en relation pourrait être échantillonné de façon disproportionnée, c'est-à-dire, que la fraction de sélection atteinte soit significativement différente de la fraction de sélection configurée. Cela peut arriver :
(i) par un comportement imprévu de la fonction de hachage, ou
(ii) parce que les paquets ont été délibérément construits pour avoir cette propriété.

Le premier point souligne l'importance de l'utilisation d'une fonction de hachage avec de bonnes propriétés de mixage. Pour cela, les propriétés statistiques des fonctions de hachage candidates doivent être évaluées. Comme le résultat du hachage dépend du mixage du trafic, l'évaluation devrait être faite de préférence sur des traces de paquet mises à jour à partir du réseau dans lequel le choix fondé sur le hachage va être déployé.

Cependant, les fonctions de hachage qui fonctionnent bien sur le trafic normal peuvent n'être pas suffisamment fortes pour soutenir les attaques spécifiquement ciblés contre elles. De telles attaques potentielles ont été décrites dans [GoRe07].

Dans les paragraphes qui suivent, on souligne différents scénarios potentiels d'attaque. On encourage à l'utilisation de fonctions de hachage normalisées. Donc, on suppose que la fonction de hachage elle-même est publique et donc connue d'un attaquant.

Néanmoins, on suppose aussi la possibilité d'utiliser un paramètre d'entrée privé pour la fonction de hachage, qui est gardé secret. Un tel paramètre d'entrée peut par exemple être rattaché à l'entrée de hachage avant l'application de l'opération de hachage. Avec cela, au moins des parties de l'opération de hachage restent secrètes.

Pour les scénarios d'attaque, on suppose qu'un attaquant utilise sa connaissance de la fonction de hachage pour construire des paquets qui sont alors envoyés, soit au titre de l'attaque elle-même, soit pour récolter plus d'informations qui peuvent être utilisée pour raffiner l'attaque.

Deux scénarios sont considérés. Dans le premier scénario, l'attaquant n'a pas connaissance de si les paquets construits sont choisis ou non. Dans le second scénario, l'attaquant utilise sa connaissance du résultat de l'échantillonnage. Les moyens par lesquels cela pourrait être acquis sont discutés ci dessous. Des attaques supplémentaires qui impliquent d'altérer les paquets d'export en transit, par opposition à l'attaque de l'appareil PSAMP, sont discutées dans [GoRe07].

#### 6.2.3.1 Vulnérabilités du choix fondé sur le hachage sans connaissance du résultat du choix

(i) La fonction de hachage n'utilise pas de paramètre privé.
Si aucun paramètre d'entrée privé n'est utilisé, les attaquants potentiels peuvent facilement calculer quels paquets résultent en quelles valeurs de hachage. Si la gamme de sélection est publique, un attaquant peut construire des paquets dont les propriétés de choix sont connues à l'avance. Si la gamme de choix est privée, un attaquant ne peut pas

déterminer si un paquet fabriqué est choisi. Cependant, en calculant le hachage sur différents paquets d'essai fabriqués, et en choisissant ceux qui donnent une certaine valeur de hachage, l'attaquant peut construire un ensemble arbitrairement grand de paquets distincts avec des propriétés de sélection communes, c'est-à-dire, les paquets qui vont être soit tous sélectionnés ou pas du tout sélectionnés. Cela peut être fait quelle que soit la force de la fonction de hachage.

(ii) La fonction de hachage n'est pas cryptographiquement forte.
Si la fonction de hachage n'est pas cryptographiquement forte, il est possible de construire des séquences de paquets distincts avec la propriété de sélection commune même si un paramètre privé est utilisé.

Un exemple est la fonction de hachage standard CRC-32 utilisée avec un module privé (mais sans chaîne privée rajoutée à l'entrée). Elle a des propriétés de mixage faibles pour les bits de moindre poids. Par conséquent, en incrémentant simplement l'entrée de hachage, on obtient des paquets distincts dont les hachages tombent pour la plupart dans une gamme étroite, et donc ont une forte probabilité d'être couramment choisis ; voir [GoRe07].

Un choix convenable des paramètres de la fonction de hachage peut rendre de telles attaques plus difficiles. Par exemple, rajouter à la fin de l'entrée une chaîne privée avant de hacher avec le CRC-32 va donner de plus fortes propriétés de mixage sur tous les bits de l'entrée. Cependant, avec une fonction de hachage, comme avec le CRC-32, qui n'est pas fort cryptographiquement, la possibilité de découvrir une méthode de construction d'ensembles de paquets avec la propriété de sélection commune ne peut pas être exclue, même quand un module privé ou l'ajout à la fin d'une chaîne est utilisé.

#### 6.2.3.2 Vulnérabilités du choix fondé sur le hachage en utilisant la connaissance du résultat du choix

La connaissance du résultat de sélection de paquets fabriqués peut être utilisée par un attaquant pour construire plus facilement des ensembles de paquets qui sont échantillonnés de façon disproportionnée et/ou sont couramment choisis. Pour cela, l'attaquant n'a pas besoin d'une connaissance à priori de la fonction de hachage ou de la gamme de choix.

Un attaquant pourrait acquérir cette connaissance du résultat de sélection de plusieurs façons :

(i) Rapports de facturation : si les échantillons sont utilisés à des fins de facturation, alors le résultat de la sélection des paquets peut être déduit en corrélant un flux de paquets fabriqués avec les rapports de facturation qu'il génère. Cependant, le taux auquel la connaissance des résultats de sélection peut être acquis dépend de la granularité temporelle et spatiale des rapports de facturation ; étant plus lent quand les rapports sont plus agrégés.

(ii) Retours d'un système de détection d'intrusion : par exemple, un adversaire maître d'un robot apprend si ses paquets ont été détectés par le système de détection d'intrusion en voyant si un de ses robots est bloqué par le réseau.

(iii) Observation du flux de rapports : les paquets d'export envoyés sur un réseau public peuvent être espionnés par un adversaire. Le chiffrement des paquets d'export fournit seulement une défense partielle, car it est possible de déduire les résultats de sélection des paquets en corrélant un flux de paquets fabriqués avec l'occurrence (pas le contenu) des paquets dans le flux d'export qu'il génère. Le taux auquel une telle connaissance pourrait être acquise est limité par la résolution temporelle à laquelle les rapports peuvent être associés aux paquets, par exemple, du fait de la variabilité du traitement et de la propagation, et de la difficulté de distinguer le rapport sur les paquets d'attaque de ceux sur le trafic de fond, si il est présent. L'association entre les paquets et leurs rapports dont elle dépend pourrait être supprimée en bourrant les paquets d'export à une longueur constante et en les envoyant à un taux constant.

On se tourne maintenant vers les attaques qui peuvent exploiter la connaissance des résultats de sélection. D'abord, avec une fonction de hachage non cryptographique, la connaissance des résultats de sélection pour un flux d'essai peut être utilisée pour mieux fabriquer un ensemble de paquets avec la propriété de sélection commune. Cela a été démontré pour le hachage modulaire $f(x) = a\ x + b \bmod k$, pour les paramètres privés a, b, et k. Avec le taux d'échantillonnage p, la connaissance du résultat de l'échantillonnage d'en gros 2/p est suffisant pour que l'attaque réussisse, indépendamment des valeurs de a, b, et k. Avec la connaissance des résultats de la sélection d'un plus grand nombre de paquets, les paramètres a, b, et k peuvent être déterminés ; voir [GoRe07].

Une fonction de hachage cryptographique employant un paramètre privé et fonctionnant dans les modes de fonction pseudo aléatoire spécifiés ci-dessus n'est pas vulnérable à ces attaques, même si la gamme de choix est connue.

#### 6.2.3.3 Vulnérabilités aux attaques en répétition

Comme le choix fondé sur le hachage est déterministe, tout paquet ou ensemble de paquets avec des propriétés de sélection connues peut être répété dans le réseau et rencontrer les même résultats de sélection pourvu que la fonction de hachage et

ses paramètres ne soient pas changés. La répétition d'un seul paquet peut être remarquée par d'autres méthodes de mesure si elles sont employées (par exemple, une collection de statistiques de flux) tandis qu'un ensemble de paquets distincts qui apparaît statistiquement similaire au trafic régulier peut être moins remarquable.

Les attaques en répétition peuvent être atténuées par des changements répétés des paramètres de la fonction de hachage. Cela empêche aussi les attaques qui exploitent la connaissance du résultat de l'échantillonnage, au moins si les paramètres sont changés au moins aussi vite que la connaissance peut être acquise par l'attaquant. Afin de préserver la capacité d'effectuer l'échantillonnage de trajectoire, le changement de paramètre devrait être simultané (ou approximativement) sur tous les points d'observation.

### 6.2.4 Choix de la fonction de hachage

Le choix spécifique de fonction de hachage représente un compromis entre complexité et facilité de mise en œuvre. Idéalement, une fonction de hachage cryptographiquement forte employant un paramètre privé et fonctionnant en mode de fonction pseudo aléatoire comme spécifié ci-dessus serait utilisée, donnant une bonne émulation d'un choix aléatoire de paquet au taux d'échantillonnage cible, et donnant une robustesse maximale contre les attaques décrites au paragraphe précédent. Malheureusement, il n'y a pas actuellement une seule fonction de hachage qui satisfasse toutes ces exigences.

Comme précisé au paragraphe 6.2.3, seules les fonctions de hachage cryptographique employant un paramètre privé fonctionnent en mode de fonction pseudo aléatoire sont suffisamment fortes pour couvrir la gamme des attaques concevables. Par exemple, des entrées de longueur fixe ou variable pourraient être hachées en utilisant un chiffrement de bloc (comme de la norme de chiffrement évolué (AES, *Advanced Encryption Standard*)) en mode de chaînage de bloc de chiffrement. Des entrées de longueur fixe pourraient aussi être hachées en utilisant l'itération d'une fonction de hachage cryptographique (comme MD5 ou SHA1) avec une valeur privée initiale. Pour les entrées de longueur variable, une itération de fonction de hachage cryptographique (comme MD5 ou SHA1) devrait employer l'ajout d'une chaîne privée à la fin des données en plus d'une valeur privée initiale. Pour les détails, voir la construction "append-cascade" de [BeCK96]. On encourage l'utilisation de telles fonctions de hachage cryptographiquement fortes chaque fois que possible.

Cependant, un problème de l'utilisation de ces fonctions est celui des faibles performances. Comme le montre par exemple [Henk08], les temps de calcul pour MD5 et SHA sont environ 7 à 10 fois plus élevés comparés aux fonctions non cryptographiques. La différence s'accroît pour les petites longueurs d'entrée de hachage.

Donc, on ne peut pas supposer que tous les appareils PSAMP vont être capables d'appliquer une fonction de hachage cryptographiquement forte à chaque paquet au taux de ligne. Pour cette raison, les fonctions de hachage mentionnées dans ce paragraphe vont être d'une variété plus faible. De futures extensions de protocole qui emploient de plus fortes fonctions de hachage sont les bienvenues.

On trouvera des comparaisons de fonctions de hachage pour le choix de paquet et le résumé de paquet à l'égard de divers critères dans [MoND05] et [Henk08].

#### 6.2.4.1 Fonctions de hachage pour le choix de paquet

Si le choix de paquet fondé sur le hachage est appliqué, la fonction BOB DOIT être utilisée pour les opérations de choix de paquets afin d'être conforme à PSAMP. La spécification de BOB est donnée dans l'appendice. Le paramètre (valeur initiale) et la gamme de sélection devraient tous deux rester privés. La valeur initiale de la fonction de hachage DOIT être configurable hors bande pour empêcher des problèmes de sécurité comme l'exposition du contenu de la valeur initiale.

D'autres fonctions, comme CRC-32 et IPSX, PEUVENT être utilisées. La fonction IPSX est décrite dans l'appendice, et la fonction CRC-32 est décrite dans la [RFC1141]. Si CRC-32 est utilisé, l'entrée devrait d'abord être complétée en fin d'une chaîne privée agissant comme paramètre, et le modulo du CRC devrait aussi rester privé.

IPSX est simple à mettre en œuvre et a été en conséquence d'un ordre de grandeur plus rapide à exécuter par paquet que BOB ou CRC-32 [MoND05].

L'évaluation de ces trois fonctions de hachage a montré une relativement pauvre uniformité avec une entrée de 16 octets tirée seulement des champs invariants dans les champs d'en-tête IP et TCP/UDP (c'est-à-dire, les champs d'en-tête qui ne changent pas d'un bond à l'autre). IPSX est par nature limité à 16 octets.

BOB et CRC-32 présentent une uniformité notablement meilleure quand 4 octets ou plus de la charge utile sont aussi inclus dans l'entrée [MoND05]. BOB a eu aussi d'assez bonnes performances avec d'autres critères [Henk08].

Bien que les caractéristiques aient été vérifiées pour différents profils de trafic, les résultats ne peuvent pas être généralisés à du trafic arbitraire. Comme le choix fondé sur le hachage est une fonction déterministe sur le contenu de paquet, il peut toujours être biaisé à l'égard de paquets qui ont des attributs spécifiques. De plus, on devrait noter que toutes les fonctions de hachage ont seulement été évaluées pour IPv4.

Aucune de ces fonctions de hachage n'est recommandée pour des besoins de cryptographie. Noter aussi que l'utilisation d'un paramètre privé réduit seulement légèrement la vulnérabilité aux attaques. Comme on le montre au paragraphe 6.2.3, les fonctions qui ne sont pas cryptographiquement fortes (par exemple, BOB et CRC) ne peuvent pas empêcher des attaquants de fabriquer des paquets qui soient choisis de façon disproportionnée même si un paramètre privé est utilisé et si la gamme de sélection reste secrète.

Comme décrit au paragraphe 6.2.2, les octets d'entrée à la fonction de hachage doivent être invariants le long du chemin que traverse le paquet. C'est seulement comme cela qu'il est assuré que les mêmes paquets sont choisis aux différents points d'observation. De plus, il devrait y avoir une forte variabilité entre les différents paquets pour générer une forte variation dans la gamme de hachage. Une évaluation de la variabilité des différents champs d'en-tête de paquet se trouve dans [DuGr00], [HeSZ08], et [Henk08].

Si un choix fondé sur le hachage avec la fonction BOB est utilisé sur du trafic IPv4, les octets d'entrée suivants DOIVENT être utilisés.
- champ d'identification IP
- champ Fanions
- décalage de fragment
- adresse IP de source
- adresse IP de destination
- un nombre configurable d'octets de la charge utile IP, commençant à un décalage configurable.

Du fait du manque de traces de paquet IPv6 convenables, toutes les fonctions de hachage candidates dans [DuGr00], [MoND05], et [Henk08] ont été évaluées seulement pour IPv4. Du fait de la structure des champs d'en-tête et adresse IPv6, on s'attend à ce qu'il y ait moins d'aléa dans les champs d'en-tête de paquet IPv6 que dans les champs d'en-tête IPv4. Néanmoins, l'aléa du trafic IPv6 n'a pas encore été suffisamment évalué pour obtenir aucune certitude. De plus, les profils de trafic IPv6 peuvent changer significativement à l'avenir quand IPv6 sera utilisé par une communauté plus large.

Si un choix fondé sur le hachage avec la fonction BOB est utilisé sur du trafic IPv6, les octets d'entrée suivants DOIVENT être utilisés.
- longueur de charge utile (2 octets)
- les octets numéro 10, 11, 14, 15, 16 de l'adresse de source IPv6
- les octets numéro 10, 11, 14, 15, 16 de l'adresse de destination IPv6
- un nombre configurable d'octets de la charge utile IP, commençant à un décalage configurable. Il est recommandé d'utiliser au moins 4 octets provenant de la charge utile IP.

La charge utile elle-même ne change pas sur le chemin. Même si certains routeurs traitent certains champs d'en-tête d'extension, ils ne vont pas le supprimer du paquet. Donc, la longueur de charge utile est invariante le long du chemin. De plus, elle diffère généralement pour les différents paquets. L'adresse IPv6 fait 16 octets. La première partie est la partie réseau et contient peu de variation. La second partie est la partie hôte et contient plus de variation. Donc, la seconde partie de l'adresse est utilisée. Néanmoins, l'uniformité n'a pas été vérifiée pour le trafic IPv6.

#### 6.2.4.2 Fonctions de hachage convenables pour le résumé de paquet

La fonction BOB DEVRAIT aussi être utilisée à cette fin. D'autres fonctions (comme CRC-32) PEUVENT être utilisées. Parmi les fonctions capables d'opérer avec des entrées de longueur variable, BOB et CRC-32 ont l'exécution la plus rapide, BOB étant légèrement plus rapide. IPSX n'est pas recommandé pour le résumé parce que il a un taux de collision significativement plus élevé et prend seulement une entrée de longueur fixe.

## 7. Paramètres pour la description des techniques de choix

Cette section donne une vue générale des différents schéma de sélection et de leurs paramètres nécessaires. Afin d'être conforme à PSAMP, au moins un des schémas proposés DOIT être mis en œuvre.

La décision de choisir ou non un paquet est fondée sur une fonction qui est effectuée quand le paquet arrive au processus de choix. Les schémas de choix de paquet diffèrent par les paramètres d'entrée pour le processus de choix et les fonctions qu'ils exigent pour faire le choix de paquet. Le tableau suivant les récapitule.

| **Schéma** | **Paramètres d'entrée** | **Fonctions** |
|---|---|---|
| Systématique fondé sur le compte | Schéma d'échantillonnage de position de paquet | compteur de paquets |
| Systématique fondé sur le temps | Schéma d'échantillonnage d'heure d'arrivée | horloge ou temporisateur |
| Aléatoire de n parmi N | Schéma d'échantillonnage de position de paquet (liste de nombres aléatoires) | compteur de paquets, nombres aléatoires |
| Probabiliste uniforme | Probabilité d'échantillonnage | fonction aléatoire |
| Probabiliste non uniforme | Par exemple, position de paquet , contenu (parties) de paquet | fonction de sélection, calcul de probabilité |
| État de flux non uniforme | Par exemple, état de flux contenu (parties) de paquet | fonction de sélection, calcul de probabilité |
| Correspondance de propriété | Contenu (parties) de paquet ou état de routeur | fonction de filtre ou découverte d'état |
| Fondé sur le hachage | Contenu (parties) de paquet | fonction de hachage |

### 7.1 Description des techniques d'échantillonnage

Dans cette section, on définit quels éléments sont nécessaires pour décrire les techniques d'échantillonnage les plus courantes. Ici la fonction de sélection est prédéfinie et donnée par l'identifiant de sélecteur.

Description d'échantillonneur :
SELECTOR_ID
SELECTOR_TYPE
SELECTOR_PARAMETERS

où :

SELECTOR_ID : identifiant univoque pour l'échantillonneur de paquets.

SELECTOR_TYPE : pour les processus d'échantillonnage, le type de sélecteur définit quel algorithme d'échantillonnage est utilisé.
Valeurs : Systématique fondé sur le compte | Systématique fondé sur le temps| Aléatoire | n parmi N | Probabiliste uniforme | Probabiliste non uniforme | État de flux non uniforme

SELECTOR_PARAMETERS : pour les processus d'échantillonnage, SELECTOR PARAMETERS définit les paramètres d'entrée pour le processus. La longueur d'intervalle dans l'échantillonnage systématique signifie que tous les paquets qui arrivent dans cet intervalle sont choisis. Le paramètre d'espacement définit le temps d'espacement ou le nombre de paquets entre la fin d'un intervalle d'échantillonnage et le début du prochain intervalle suivant.

Cas de n parmi N : taille de population N, taille d'échantillon n

Cas du systématique fondé sur le temps : longueur d'intervalle (en micro s) espacemennt (en micro s)

Cas du systématique fondé sur le compte : longueur d'intervalle (en paquets) espacement (en paquets)

Cas du probabiliste uniforme (avec probabilité égale par paquet) : probabilité d'échantillonnage p

Cas probabiliste non uniforme : fonction de calcul pour la probabilité d'échantillonnage p (voir aussi le paragraphe 5.2.2.4)

Cas de l'état de flux : les informations rapportées de l'échantillonnage d'état de flux ne sont pas définies dans ce document (voir aussi au paragraphe 5.2.2.4)

### 7.2 Description des techniques de filtrage

Dans ce paragraphe, on définit quels éléments sont nécessaire pour décrire les techniques de filtrage les plus courante. La structure suit étroitement celle présentée pour les techniques d'échantillonnage.

Description de filtre :
SELECTOR_ID
SELECTOR_TYPE
SELECTOR_PARAMETERS

où :

SELECTOR_ID : identifiant univoque pour le filtre de paquet. L'identifiant peut être calculé en considération de la séquence de sélection et d'un identifiant local.

SELECTOR_TYPE : pour les processus de filtrage, le type de sélecteur définit quel type de filtrage est utilisé.
Valeurs : Correspondance | Hachage | État de routeur.

SELECTOR_PARAMETERS : pour les processus de filtrage, paramètres du sélecteur définit formellement les propriétés communes du paquet filtré. Pour les filtres de type correspondance et hachage, les définitions ont beaucoup de points en commun.
Valeurs :
Cas de correspondance :
- Élément d'information (d'après la [RFC5102])
- Valeur (type conforme à la [RFC5102])

En cas de plusieurs critères de correspondance, plusieurs "case matching" doivent êt reliés par un ET logique.

Cas de hachage :
- Domaine hachage (bits d'entrée provenant du paquet)
  - <type d'en-tête = IPv4>
  - <spécification de bits d'entrée, partie en-tête>
  - <type d'en-tête = IPv6>
  - <spécification de bits d'entrée, partie en-tête>
  - <nombre d'octets de charge utile N>
  - <spécification de bits d'entrée, partie en-tête>
- fonction de hachage
  - nom de fonction de hachage
  - longueur de clé d'entrée (élimine 0x octets)
  - valeur de résultat (longueur M et gabarit binaire)
  - gamme de choix de hachages, comme liste d'intervalles sans chevauchement [valeur de début, valeur de fin] où la valeur est dans [0,2^M-1]
  - les paramètres supplémentaires dépendent de la fonction de hachage spécifique (par exemple, les bits d'entrée du hachage (germe))

Notes sur les bits d'entrée pour le cas de hachage :
- Les bits d'entrée peuvent provenir seulement de la partie en-tête, de la partie charge utile seulement, ou des deux.
- La spécification des bits, pour la partie en-tête, peut être spécifiée seulement pour IPv4 ou pour IPv6, ou les deux.
- Dans le cas de IPv4, la spécification des bits est une séquence de 20 chiffres hexadécimaux [00 à FF] spécifiant un gabarit binaire de 20 octets à appliquer à l'en-tête.
- Dans le cas de IPv6, c'est une séquence de 40 nombres hexadécimaux [00 à FF] spécifiant un gabarit binaire de 40 octets à appliquer à l'en-tête.
- La spécification des bits, pour la partie charge utile, est une séquence de nombres hexadécimaux [00 à FF] spécifiant le gabarit binaire à appliquer aux N premiers octets de la charge utile, comme spécifié dans le champ précédent. Dans le cas où la séquence de nombres hexadécimaux est plus longue que N, seuls les N premiers nombres sont considérés.
- Dans le cas où la charge utile est plus courte que N, la fonction de hachage ne peut pas être appliquée. D'autres options, comme le bourrage avec des zéros, pourront être considérées à l'avenir.
- Une fonction de hachage ne peut pas être définie sur les champs d'options de l'en-tête IPv4, ni sur les champs d'en-tête en pile de IPv6.
- La gamme de choix de hachages définit une gamme de valeurs de hachage (parmi tous les résultats possibles de l'opération de hachage). Si le résultat du hachage pour un paquet spécifique tombe dans cette gamme, le paquet est choisi. Si la valeur est en dehors de la gamme, le paquet n'est pas choisi. Par exemple, si la spécification de l'intervalle de sélection est [1:3], [6:9] tous les paquets pour lesquels le résultat de hachage est 1,2,3,6,7,8, ou 9 sont choisis. Dans tous les autres cas, le paquet n'est pas choisi.

Cas de l'état de routeur :

- L'interface d'entrée à laquelle le paquet arrive est égale à une valeur spécifiée
- L'interface de sortie à laquelle le paquet est acheminé est égale à une valeur spécifiée
- Le paquet a violé la liste de contrôle d'accès (ACL) sur le routeur
- La transmission sur le chemin inverse (RPF) a échoué pour le paquet
- La réservation de ressource est insuffisante pour le paquet
- Aucun chemin n'est trouvé pour le paquet
- L'AS d'origine est égal à une valeur spécifiée ou se tient dans une gamme donnée
- L'AS de destination est égal à une valeur spécifiée ou se tient dans une gamme donnée.

Notes sur le cas de l'état de routeur :
- Toutes les entrées d'état de routeur peuvent être liées par des opérateurs ET logiques.

## 8. Techniques composites

Des schémas composites sont réalisés en combinant les identifiants de sélecteur dans une séquence de sélection. La séquence de sélection contient tous les identifiants de sélecteur qui sont appliqués ensuite au flux de paquets. Des exemples de schémas composites sont donnés ci-dessous.

### 8.1 Filtrage en cascade->échantillonnage ou échantillonnage->filtrage

Si un filtre précède un processus d'échantillonnage, le rôle du filtrage est de créer un ensemble de "populations parentes" à partir d'un seul flux qui peut alors être alimenté indépendamment à différentes fonctions d'échantillonnage, avec différents paramètres réglés pour la population elle-même (par exemple, si des flux de différentes intensités résultent du filtrage, il peut être bon d'avoir des taux d'échantillonnage différents). Si le filtrage suit un processus d'échantillonnage, les mêmes fraction et type de sélection sont appliqués à tout le flux, indépendamment de la taille relative des flux résultant de la fonction de filtrage. De plus, les paquets non destinés à être sélectionnés dans l'opération de filtrage vont aussi "charger" la fonction d"échantillonnage. Donc, en principe, le filtrage avant l'échantillonnage permet un réglage plus précis de la procédure d'échantillonnage, mais si les filtres sont trop complexes à faire fonctionner au plein taux de ligne (par exemple, parce que ils doivent accéder à des informations d'état de routeur) l'échantillonnage avant le filtrage peut être nécessaire.

### 8.2 Échantillonnage stratifié

L'échantillonnage stratifié est un exemple d'utilisation d'une technique composite. L'idée de base de l'échantillonnage stratifié est d'augmenter la précision de l'estimation en utilisant des informations a priori sur les corrélations de la caractéristique recherchée avec d'autres caractéristiques plus faciles à obtenir. Les informations a priori sont utilisées pour effectuer un groupement intelligent des éléments de la population parente. De cette manière, une meilleure précision de l'estimation peut être réalisée avec la même taille d'échantillon ou la taille d'échantillon peut être réduite sans réduire la précision de l'estimation.

L'échantillonnage stratifié divise le processus d'échantillonnage en plusieurs étapes. D'abord, les éléments de la population parente sont groupés en sous ensembles en accord avec une caractéristique donnée. Ce groupement peut être fait en plusieurs étapes. Ensuite les échantillons sont pris dans chaque sous ensemble.

Plus forte est la corrélation entre la caractéristique utilisée pour diviser la population parente (stratification variable) et la caractéristique intéressante (pour laquelle une estimation est ensuite recherché) plus facile est le processus d'échantillonnage consécutif et plus fort est le gain de la stratification. Par exemple, si la caractéristique de division était égale à la caractéristique recherchée, chaque élément du sous groupe serait un représentant parfait de cette caractéristique. Dans ce cas, il serait suffisant de prendre un élément arbitraire de chaque sous groupe pour avoir la distribution réelle de la caractéristique dans la population parente. Donc, l'échantillonnage stratifié peut réduire les coûts du traitement d'échantillonnage (c'est-à-dire, le nombre des échantillons nécessaires pour réaliser un niveau de confiance donné).

Pour l'échantillonnage stratifié, on doit spécifier des règles de classification pour les groupement des éléments dans les sous groupes et le schéma d'échantillonnage qui est utilisé dans les sous groupes. Les règles de classification peuvent être exprimées par plusieurs filtres. Pour le schéma d'échantillonnage au sein des sous groupes, les paramètres doivent être spécifiés comme décrit ci-dessus. L'utilisation de méthodes d'échantillonnage stratifié pour les mesures est décrit par exemple dans [ClPB93] et [Zseb03].

## 9. Considérations sur la sécurité

Les considérations sur la sécurité concernant le choix d'une fonction de hachage pour le choix fondé sur la hachage ont été discutées au paragraphe 6.2.3. Cette section discute un certain nombre d'attaques potentielles pour fabriquer des flux de paquets qui sont détectés de façon disproportionnée et/ou pour découvrir les paramètres des fonctions de hachage, les vulnérabilités des différentes fonctions de hachage à ces attaques, et les pratiques pour minimiser ces vulnérabilités.

En plus de cela, un utilisateur peut obtenir des connaissances sur les déclencheurs de début et de fin dans l'échantillonnage systématique fondé sur le temps, par exemple, en envoyant des paquets d'essai. Cette connaissance pourrait permettre à des utilisateurs de modifier leur programmation d'envoi de façon que leurs paquets soient choisis ou non choisis de façon disproportionnée [GoRe07].

Pour l'échantillonnage aléatoire, un générateur de nombre aléatoires cryptographiquement fort devrait être utilisé afin d'empêcher qu'un adversaire puisse prédire les décisions de sélection [GoRe07].

D'autres menaces pour la sécurité peuvent survenir quand les paramètres d'échantillonnage sont configurés ou communiqués à d'autres entités. La configuration et le rapport des paramètres d'échantillonnage sortent du domaine d'application du présent document. Donc, les menaces pour la sécurité qui ont pour origine cette sorte de communication ne peuvent pas être attestées avec les information données dans ce document.

Certaines de ces menaces peuvent probablement être traitées en gardant les informations de configuration confidentielles et en authentifiant les entités qui configurent l'échantillonnage. Néanmoins, une analyse et une vérification complète des menaces pour la configuration et les rapports doivent être faites si des méthodes de configuration ou de rapport sont proposées.

## 10. Contributeurs

Sharon Goldberg a contribué aux considérations sur la sécurité pour le choix fondé sur le hachage.

Sharon Goldberg
Department of Electrical Engineering
Princeton University
F210-K EQuad
Princeton, NJ 08544,
USA
mél : goldbe@princeton.edu

## 11. Remerciements

Nous tenons à remercier le groupe PSAMP, et en particulier Benoît Claise et Stewart Bryant, pour les discussions fructueuses et la relecture du document. Nous remercions Sharon Goldberg de son apport sur les questions de sécurité concernant le choix fondé sur le hachage.

## 12. Références

### 12.1 Références normatives

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

### 12.2 Références pour information

[AmCa89] Paul D. Amer, Lillian N. Cassel, "Management of Sampled Real-Time Network Measurements", 14th

Conference on Local Computer Networks, octobre 1989, Minneapolis, pages 62-68, IEEE, 1989.

[BeCK96] M. Bellare, R. Canetti and H. Krawczyk, "Pseudorandom Functions Revisited: The Cascade Construction and its Concrete Security", Symposium on Foundations of Computer Science, 1996.

[ClPB93] K.C. Claffy, George C. Polyzos, Hans-Werner Braun, "Application of Sampling Methodologies to Network Traffic Characterization", ACM SIGCOMM'93, San Francisco, CA, USA, 13-17 septembre 1993.

[DuGG02] N.G. Duffield, A. Gerber, M. Grossglauser, "Trajectory Engine: A Backend for Trajectory Sampling", IEEE Network Operations and Management Symposium 2002, Florence, Italy, 15-19 avril 2002.

[DuGr00] N.G. Duffield, M. Grossglauser, "Trajectory Sampling for Direct Traffic Observation", Proceedings of ACM SIGCOMM 2000, Stockholm, Sweden, 28 août, 1er septembre 2000.

[DuGr04] N.G. Duffield and M. Grossglauser "Trajectory Sampling with Unreliable Reporting", Proc IEEE Infocom 2004, Hong Kong, mars 2004.

[DuLT01] N.G. Duffield, C. Lund, and M. Thorup, "Charging from Sampled Network Usage", ACM Internet Measurement Workshop IMW 2001, San Francisco, USA, 1-2 novembre 2001.

[EsVa01] C. Estan and G. Varghese, "New Directions in Traffic Measurement and Accounting", ACM SIGCOMM Internet Measurement Workshop 2001, San Francisco CA, novembre 2001.

[GoRe07] S. Goldberg, J. Rexford, "Security Vulnerabilities and Solutions for Packet Sampling", IEEE Sarnoff Symposium, Princeton, NJ, mai 2007.

[HT52] D.G. Horvitz and D.J. Thompson, "A Generalization of Sampling without replacement from a Finite Universe" J. Amer. Statist. Assoc. Vol. 47, pp. 663-685, 1952.

[Henk08] Christian Henke, "Evaluation of Hash functions for Multipoint Sampling in IP Networks", Thèse de doctorat, TU Berlin, avril 2008.

[HeSZ08] Christian Henke, Carsten Schmoll, Tanja Zseby, "Evaluation of Header Field Entropy for Hash-Based Packet Selection", Proceedings of Passive and Active Measurement Conference PAM 2008, Cleveland, Ohio, USA, avril 2008.

[Jenk97] B. Jenkins, "Algorithm Alley", Dr. Dobb's Journal, septembre 1997. http://burtleburtle.net/bob/hash/doobs.html.

[JePP92] Jonathan Jedwab, Peter Phaal, Bob Pinna, "Traffic Estimation for the Largest Sources on a Network, Using Packet Sampling with Limited Storage", HP technical report, Managemenr, Mathematics and Security Department, HP Laboratories, Bristol, mars 1992, http://www.hpl.hp.com/techreports/92/HPL-92-35.html.

[Moli03] M. Molina, "A scalable and efficient methodology for flow monitoring in the Internet", International Teletraffic Congress (ITC-18), Berlin, septembre 2003.

[MoND05] M. Molina, S. Niccolini, N.G. Duffield, "A Comparative Experimental Study of Hash functions Applied to Packet Sampling", International Teletraffic Congress (ITC-19), Beijing, août 2005.

[RFC1141] T. Mallory et A. Kullberg, "Mise à jour incrémentaire de la somme de contrôle Internet", janvier 1990.

[RFC1624] A. Rijsinghani, éditeur, "Calcul de la somme de contrôle Internet via une mise à jour incrémentaire", mai 1994. *(Info.)*

[RFC2205] R. Braden, éd., L. Zhang, S. Berson, S. Herzog, S. Jamin, "Protocole de réservation de ressource (RSVP) -- version 1, spécification fonctionnelle", septembre 1997. (*MàJ par* RFC2750, RFC3936, RFC4495, RFC6780)) *(P.S.)*

[RFC3704] F. Baker, P. Savola, "Filtrage d'entrée pour réseaux à rattachements multiples", mars 2004. (BCP0084) (*MàJ par* RFC8704)

[RFC3917] J. Quittek, T. Zseby, B. Claise, S. Zander, "Exigences pour l'exportation d'informations de flux IP (IPFIX)", octobre 2004. *(Information)*

[RFC4271] Y. Rekhter, T. Li et S. Hares, "Protocole de routeur frontière version 4 (BGP-4)", janvier 2006. *(D.S.) (MàJ par* RFC6608, RFC8212, RFC9072)

[RFC5101] B. Claise, éd., "Spécification du protocole d'exportation d'informations de flux IP (IPFIX) pour l'échange d'informations de flux de trafic IP", janvier 2008. *(P.S.) (Obsolète, voir* RFC7011, STD77)

[RFC5102] J. Quittek et autres, "Modèle d'informations pour l'exportation d'informations de flux IP", janvier 2008. *(P.S.) (Remplacée par* RFC7012)

[RFC5474] N. Duffield et autres, "Cadre de travail pour la sélection et le rapport de paquet", mars 2009. *(Information)*

[RFC5476] B. Claise et autres "Spécification du protocole d'échantillonnage de paquet (PSAMP)", mars 2009. *(P.S.)*

[RFC5477] T. Dietz et autres, "Modèle d'information pour l'exportation d'échantillonnage de paquet", mars 2009. *(P.S.)*

[Zseb03] T. Zseby, "Stratification Strategies for Sampling-based Non-intrusive Measurement of One-way Delay", Proceedings of Passive and Active Measurement Workshop (PAM 2003), La Jolla, CA, USA, pp. 171-179, avril 2003.

[ZsZC01] Tanja Zseby, Sebastian Zander, Georg Carle. "Evaluation of Building Blocks for Passive One-way-delay Measurements". Proceedings of Passive and Active Measurement Workshop (PAM 2001), Amsterdam, The Netherlands, 23-24 avril 2001.

# Appendice A. Fonctions de hachage

## A.1 Fonction de hachage IP Shift-XOR (IPSX)

La fonction de hachage IPSX est conçue pour agir sur les paquets IP version 4. Elle exploite la structure des paquets IP et en particulier la variabilité attendue présentée dans les différents champs du paquet IP afin de fournir une valeur de hachage avec peu de corrélation apparente avec les champs de paquet individuels. Les champs des champs d'en-tête IPv4 et TCP/UDP sont utilisés en entrée. La fonction de hachage IPSX utilise un petit nombre d'instructions simples.

Paramètres d'entrée : aucun
Paramètres incorporés : aucun
Résultat : le résultat de IPSX est un nombre de 16 bits
Fonctionnement : le fonctionnement peut être divisé en deux parties : choix d'entrée dont les formes sont l'entrée composite provenant de diverses portions du paquet IP, suivie par le calcul du hachage sur le composite.
Choix d'entrée : l'entrée brute est tirée des 20 premiers octets de l'en-tête de paquet IP et des huit premiers octets de la charge utile IP. Si des options IP ne sont pas utilisées, l'en-tête IP fait 20 octets, et donc les deux portions s'ajoutent et composent les 28 premiers octets du paquet IP. On utilise ensuite l'entrée brute comme quatre sous portions de 32 bits de ces 28 octets. On spécifie l'entrée par le décalage de bits à partir du début de l'en-tête ou charge utile IP.
f1 = bits 32 à 63 de l'en-tête IP, comprenant le champ Identification IP, les fanions, et le décalage de fragment.
f2 = bits 96 à 127 de l'en-tête IP, l'adresse IP de source.
f3 = bits 128 à 159 de l'en-tête IP, l'adresse IP de destination.
f4 = bits 32 à 63 de la charge utile IP. Pour un paquet TCP, f4 comprend le numéro de séquence TCP suivi par la longueur de message. Pour un paquet UDP, f4 comprend la somme de contrôle UDP.
Calcul de hachage : le hachage est calculé à partir de f1, f2, f3, et f4 par une combinaison d'opérations OUX (^), glissement à droite (>>), et glissement à gauche (<<). Les quantités intermédiaires h1, v1, et v2 sont des nombres de 32 bits.

1. v1 = f1 ^ f2;
2. v2 = f3 ^ f4;
3. h1 = v1 << 8;
4. h1 ^= v1 >> 4;
5. h1 ^= v1 >> 12;
6. h1 ^= v1 >> 16;
7. h1 ^= v2 << 6;
8. h1 ^= v2 << 10;
9. h1 ^= v2 << 14;

10. h1 ^= v2 >> 7;

Le résultat du hachage est les 16 bits de moindre poids de h1.

### A.2 Fonction de hachage BOB

La fonction de hachage BOB est une fonction conçue pour que chaque bit d'entrée affecte chaque bit de la valeur retournée et qui utilise des différences de 1 bit et de 2 bits pour réaliser l'effet dit d'avalanche [Jenk97]. Cette fonction a été à l'origine construite pour la recherche dans un tableau de hachage avec une mise en œuvre de logiciel rapide.

Paramètres d'entrée : les paramètres d'entrée d'une telle fonction sont :
- la longueur de la chaîne d'entrée (clé) à hacher, en octets. Les blocs d'entrée élémentaires du hachage BOB sont les seuls octets ; donc, aucun bourrage n'est nécessaire.
- une valeur initiale (un nombre arbitraire de 32 bits).

Paramètres incorporés : le hachage BOB utilise le paramètre incorporé suivant :
- le ratio d'or (un nombre arbitraire de 32 bits utilisé dans le calcul de la fonction de hachage pour éviter de transposer les tout à zéro en tout à zéro).

Note : la sous fonction mix (voir la macro mix (a,b,c) dans le code de référence ci-dessous) a un certain nombre de paramètres qui gouvernent les décalages dans les registres. Celui présenté n'est pas le seul choix possible.
La question est ouverte de savoir si on peut envisager des paramètres incorporés supplémentaires pour spécifier une configuration de fonction.

Résultat : le résultat de la fonction BOB est un nombre de 32 bits. Il devrait être spécifié par :
- un gabarit de 32 bits à appliquer au résultat,
- la gamme de sélection comme une liste d'intervalles sans chevauchement [valeur de début, valeur de fin] où valeur est dans [0 à 2^32]

Fonctionnement : la valeur de hachage est obtenue en calculant d'abord une initialisation d'un état interne (composé de trois nombres de 32 bits, appelés a, b, c dans le code de référence ci-dessous) puis, pour chaque octet d'entrée de la clé, l'état interne est combiné par addition et mixé en utilisant la sous fonction "mix". Finalement, l'état interne est mixé une dernière fois et le troisième nombre de l'état (c) est choisi comme valeur de retour.

```
typedef unsigned long int  ub4;                 /* quantités non signées de 4 octet */
typedef unsigned      char ub1;                 /* quantités non signées de 1 octet */

#define hashsize(n) ((ub4)1<<(n))
#define hashmask(n) (hashsize(n)-1)

/* --------------------------------------------------------
   mix -- mixe trois valeurs de 32 bits de façon reversible.

Pour chaque delta avec un ou deux bits établis, et les deltas des trois bits de poids fort ou des trois bits de moindre poids, si la valeur originale de a,b,c est presque toute de zéros ou est uniformément distribuée,
* Si mix() est fait vers l'avant ou vers l'arrière, au moins 32 bits dans a,b,c ont au moins une probabilité de 1/4 de changer.
* Si mix() est fait vers l'avant, chaque bit de c va changer entre 1/3 et 2/3 du temps (22/100 et 78/100 pour des deltas de 2 bits) mix() a été construit à partir de 36 instructions de latence à un seul cycle dans une structure qui pourrait supporter un parallélisme 2x, comme :
   a -= b;
   a -= c; x = (c>>13);
   b -= c; a ^= x;
   b -= a; x = (a<<8);
   c -= a; b ^= x;
   c -= b; x = (b>>13);
   ...
   Malheureusement, les Pentiums et Sparcs super scalaires ne peuvent pas tirer parti de ce parallélisme. Ils ont aussi transformé certaines de ces instructions de latence à un seul cycle en instructions de latence multi cycles
------------------------------------------------------------*/

#define mix(a,b,c)  \
{ \
  a -= b; a -= c; a ^= (c>>13); \
  b -= c; b -= a; b ^= (a<<8); \
```

```
    c -= a; c -= b; c ^= (b>>13); \
    a -= b; a -= c; a ^= (c>>12);  \
    b -= c; b -= a; b ^= (a<<16); \
    c -= a; c -= b; c ^= (b>>5); \
    a -= b; a -= c; a ^= (c>>3);  \
    b -= c; b -= a; b ^= (a<<10); \
    c -= a; c -= b; c ^= (b>>15); \
  }

/* --------------------------------------------------------
hash() : hache une clé de longueur variable en une valeur de 32 bits.
k : la clé (dispositif d'octets non aligné de longueur variable).
len : la longueur de la clé, en octets.
initval : peut être toute valeur de 4 octet. Retourne une valeur de 32 bits. Chaque bit de la clé affecte chaque bit de la valeur
de retour. Chaque delta de 1 bit et 2 bits réalise une avalanche. Environ 6*len+35 instructions.

Les meilleures tailles de tableau de hachage sont des puissances de 2. Il n'y a pas besoin d'un modulo d'un nombre premier
(mod est si lent !). Si on a besoin de moins de 32 bits, utiliser un gabarit binaire. Par exemple, si on a besoin de seulement
10 bits, faire h = (h & hashmask(10)), dans ce cas, le tableau de hachage devrait avoir hashsize(10) éléments.

Si on hache n chaînes (ub1 **)k, le faire comme suit : pour (i=0, h=0; i<n; ++i) h = hash( k[i], len[i], h);

Par Bob Jenkins, 1996. bob_jenkins@burtleburtle.net. On peut utiliser ce code de toutes les façons souhaitées, privée,
éducative, ou commerciale. Il est libre. Voir http://burtleburtle.net/bob/hash/evahash.html. À utiliser pour la recherche dans
un tableau de hachage, où toute collision sur 2^^32 est acceptable. NE PAS utiliser à des fins cryptographiques.
-------------------------------------------------------------- */


  ub4 bob_hash(k, longueur, initval)
  register ub1 *k;                                  /* la clé */
  register ub4  longueur;                           /* longueur de la clé */
  register ub4  initval;                            /* valeur arbitraire */
  {
    register ub4 a,b,c,len;

/* Établir l'état interne */
    len = longueur;
    a = b = 0x9e3779b9;                             /*le ratio d'or ; une valeur arbitraire
  */
    c = initval;                                    /* une autre valeur arbitraire */

/*---------------------------------- traite la plus grande partie de la clé */

    while (len >= 12)
    {
      a += (k[0] +((ub4)k[1]<<8) +((ub4)k[2]<<16)
  +((ub4)k[3]<<24));
      b += (k[4] +((ub4)k[5]<<8) +((ub4)k[6]<<16)
  +((ub4)k[7]<<24));
      c += (k[8] +((ub4)k[9]<<8)
  +((ub4)k[10]<<16)+((ub4)k[11]<<24));
      mix(a,b,c);
      k += 12; len -= 12;
    }

/*------------------------------ traite les 11 derniers octets */
    c += longueur;
    switch(len)                                     /* toutes les déclarations de cas passent */
    {
    cas 11 : c+=((ub4)k[10]<<24);
    cas 10 : c+=((ub4)k[9]<<16);
```

```
        cas 9 : c+=((ub4)k[8]<<8);          /* le premier octet de c est réservé pour la longueur */
        cas 8 : b+=((ub4)k[7]<<24);
        cas 7 : b+=((ub4)k[6]<<16);
        cas 6 : b+=((ub4)k[5]<<8);
        cas 5 : b+=k[4];
        cas 4 : a+=((ub4)k[3]<<24);
        cas 3 : a+=((ub4)k[2]<<16);
        cas 2 : a+=((ub4)k[1]<<8);
        cas 1 : a+=k[0];
/* cas 0 : rien ne reste à ajouter */
        }
        mix(a,b,c);
/*-------------------------------- rapporte le résultat */
        return c;
    }
```

## Adresse des auteurs

Tanja Zseby
Fraunhofer Institute for Open Communication Systems
Kaiserin-Augusta-Allee 31
10589 Berlin
Germany
téléphone : +49-30-34 63 7153
mél : tanja.zseby@fokus.fraunhofer.de

Maurizio Molina
DANTE
City House
126-130 Hills Road
Cambridge CB21PQ
United Kingdom
mél : maurizio.molina@dante.org.uk

Nick Duffield
AT&T Labs - Research
Room B-139
180 Park Ave
Florham Park, NJ 07932
USA
téléphone : +1 973-360-8726
mél : duffield@research.att.com

Saverio Niccolini
Network Laboratories, NEC Europe Ltd.
Kurfuerstenanlage 36
69115 Heidelberg
Germany
téléphone : +49-6221-9051118
mél : saverio.niccolini@netlab.nec.de

Frederic Raspall
EPSC-UPC
Dept. of Telematics
Av. del Canal Olimpic, s/n
Edifici C4
E-08860 Castelldefels, Barcelona
Spain
mél : fredi@entel.upc.es