Détection des événements de congestion de TCP

Pascal Anelli[1], **Fanilo Harivelo**[1], **Emmanuel Lochin**[2]

[1] *LIM - Université de la Réunion*
15 Avenue René Cassin, 97715 Saint Denis Messag 9 - France
{fanilo.harivelo, pascal.anelli}@univ-reunion.fr
[2]*Université de Toulouse - ISAE - DMIA*
10 av. Edouard Belin - BP 54032 - 31055 TOULOUSE Cedex 4
emmanuel.lochin@isae.fr

RÉSUMÉ. *Ces dernières années ont vu un intérêt croissant pour l'étude de la mesure des flots TCP. Plusieurs méthodes ont été proposées afin d'estimer de façon précise et rapide le taux de perte des paquets. Toutefois, et à notre meilleure connaissance, l'estimation et l'identification des événements de congestion TCP ne sont actuellement pas adressées. Suite à la standardisation de TFRC (*TCP Friendly Rate Control*), nous assistons à un intérêt croissant pour les protocoles de transport utilisant un contrôle de congestion de type* rate-based*. Aussi, ce type de contrôle de congestion repose sur la même base métrologique que celle de TCP. Dans ce contexte, la détermination précise des événements de congestion (CE) est un élément clé. En effet, ces derniers donnent une information essentielle pour le calcul d'un débit d'émission qui soit équivalent à celui de TCP dans les mêmes conditions. L'objet de cet article est de mieux identifier les CE de TCP afin de fournir une référence précise à ces nouveaux protocoles. Nous vérifions dans cette étude que TCP n'identifie pas de manière efficace les CE du réseau et proposons une méthode capable de mieux les déterminer. Cette détection est effectuée passivement à partir de la capture temps réel des paquets d'un flot TCP.*

ABSTRACT. *The problem of path loss rate estimation with TCP protocol has been actively studied by the research community. Several schemes have been proposed in order to accurately estimate the loss rate of a given path. These proposals are used for monitoring or metrology purposed. However, estimating and identifying the TCP congestion event is currently not addressed. While nowadays rate-based protocols received a particular attention from the research community, it seems important to better identify these congestion events in order to give an accurate reference to these protocols which aim at behaving in a fair manner with TCP. In this study, we propose a method able to identify a congestion event at the border of an autonomous system.*

MOTS-CLÉS : *TCP, Evénements de congestion, Algorithme de mesures*

KEYWORDS: *TCP, Congestion events, Measurments algorithm*

1. Introduction

Une des caractéristiques importantes du protocole TCP est sa capacité à adapter son débit d'émission en fonction de l'état de sa route dans le réseau. La source TCP fonctionne en boucle fermée sur un réseau qui est vu comme une boite noire avec l'objectif d'écouler le plus rapidement possible les données. Pour se faire, la source TCP ajuste son débit d'émission en fonction des retours du réseau. L'interprétation des retours est binaire : soit l'absence ou la présence de congestion ; la réception d'un paquet acquitté signifiant un retour positif. La perte d'un segment s'interprète comme un débordement d'une file d'attente et donc d'une congestion. Ce retour négatif indique à la source qu'elle doit diminuer son débit d'émission afin d'éviter l'effondrement du réseau (*congestion collapse*) et préserver son flot d'un taux de perte important. A ces considérations de performances s'ajoute une dynamique de débit qui tend vers un partage équitable des ressources entre les flots concurrents.

La période du contrôle, c'est à dire le délai entre l'action de la source et le retour, est imposée par le temps d'aller retour (RTT : *Round Trip Time*). En automatique, on qualifie cette caractéristique de système à retard. En effet, suite à l'émission d'un segment TCP, la source n'aura connaissance d'une éventuelle congestion dans le réseau qu'après un RTT plus tard. C'est donc sur une période équivalente au RTT que les congestions sont détectées par la source. Du point de vue de TCP, une congestion se produit à la première perte de paquet et dure l'équivalent d'un RTT (le temps nécessaire pour avoir un retour). Le nombre de pertes dans un RTT n'est pas pris en compte. Cependant le nombre total de pertes va avoir un impact sur la durée de la reprise qui dépend également des méthodes de reprise utilisées par TCP.

La notion d'événement de congestion (CE : *Congestion Event*) indique une fenêtre de données avec un ou plusieurs paquets perdus ou marqués ECN [FLO 03]. A noter que le terme "événement de perte" (LE : *Loss Event*) s'utilise pareillement et désigne la même notion ([FLO 07]). En d'autres termes, le retour négatif pour TCP n'est pas précisément les pertes de paquets mais l'événement de congestion qui peut aussi être notifié avec ECN (*Explicit Congestion Notification*) [RAM 01]. De façon générale, un CE constitue un retour négatif pour une source utilisant un contrôle de congestion en boucle fermée.

Une source TCP contrôle son débit d'émission au moyen d'une fenêtre de données. Lorsque la source détecte un CE, elle ramène la taille de sa fenêtre de contrôle de congestion à la quantité de données en cours de transit au moment de l'apparition de l'événement et la diminue de moitié. Puisque TCP est un protocole fiable, il effectue également la retransmission des données perdues. De manière simplifiée, on peut conclure qu'un CE se traduit par une diminution de la fenêtre de données en interne et par une retransmission visible de l'extérieure de la source TCP. Le CE pour la source TCP se termine lorsqu'elle reçoit l'acquittement de la donnée émise au moment de l'apparition de l'événement. Dans [FLO 04], cette donnée est appelée *"recover"*.

Cette notion de CE apparaît encore plus clairement dans la spécification de TFRC. TFRC est un contrôle de congestion s'appuyant sur une équation modélisant le com-

portement moyen de TCP dans le but d'être compatible avec celui-ci (principe dit *TCP-friendly* présenté dans [HAN 03]). TFRC est notamment mis en œuvre dans la version CCID3 (*Congestion Control ID 3*) [FLO 06] de DCCP (*Datagram Congestion Control Protocol*) [KOH 06]. TFRC a formalisé la notion de CE utilisée naturellement dans TCP. L'équation de TFRC ne prend pas en compte la probabilité de perte en paramètre (très difficile à déterminer avec exactitude) mais un *loss event* équivalent au CE de TCP.

La détection de la congestion par TCP n'est pas exactement la congestion du réseau (une congestion du réseau correspond à un épisode de débordement de file d'attente de routeur [BAL 05]). Sur une échelle temporelle, TCP détecte la perte un RTT après son occurrence ou au délai d'expiration du temporisateur de retransmission (RTO : *Retransmission Time-Out*). Ceci dépend de l'importance du flot. En effet, un flot comportant peu de données (dit flot court) aura plus de difficulté à détecter une perte par *fast retransmit* car le nombre de segments à émettre après la perte peut être inférieur à 3 et donc empêcher la reprise. Dans ce cas, la détection de la congestion s'effectue un RTO après la perte. La congestion détectée par TCP fournit l'indication qu'il y a eu une congestion dans le réseau mais ne précise pas à l'instant de sa détection si elle est encore d'actualité et quelle a été la durée de la congestion du réseau. Comme la congestion peut persister à différentes échelles de temps [RAM 01], le début et la fin de la congestion du réseau ne peut être connu par un flot TCP. Cependant un CE du point de vue de TCP ne peut pas être plus long que la durée d'un RTT. La détection du CE représente une estimation de la période de congestion.

Les travaux présentés dans cet article visent à développer une méthode d'identification des CE par une mesure passive [1] en temps réel applicable à TCP. La détection des CE peut trouver une application dans la métrologie bien sur mais aussi dans le développement de fonction de notification de congestion externe à TCP ou pour des politiques de conditionnement de trafic TCP. Cet article est organisé de la façon suivante : la section 2 donne les hypothèses et détaille la conception de l'algorithme de détection des CE appelé *Implicit Congestion Notification (ICN)* ; la section 3 présente la campagne de validation ; enfin la section 4 donne la conclusion et les orientations futures de ces travaux.

2. Conception d'un détecteur d'événements de congestion

L'estimation du taux de pertes de paquets TCP est une information métrologique essentielle du transfert TCP. De précédentes contributions ont permis d'établir des méthodes de mesures passives du taux de pertes de bout en bout telles [BEN 02], [ALL 03], [MEL 02]. De manière générale, le principe de base de ces algorithmes consiste en l'écoute passive des numéros de séquence afin de d'identifier les ruptures de séquences d'un flot TCP. L'analyse effectuée des traces de ces études est alors différente. Dans notre cas, nous proposons d'étendre ces travaux de détection des pertes

1. sans action sur le trafic

à la notion de CE. Pour cela on partira des algorithmes d'estimation de taux de pertes utilisées dans ces algorithmes. Il est clair que le résultat obtenu ne sera plus un taux de perte mais plutôt un *Loss Event Ratio* défini comme étant le ratio du nombre de CE par rapport au nombre de paquets transmis. Dans la section suivante, nous présentons notre algorithme de détection de CE en donnant ses hypothèses de conception et la méthodologie d'interprétation utilisée.

2.1. *Hypothèses de conception*

L'hypothèse porte sur la localisation de la détection des CE effectuée depuis la source ou depuis un nœud immédiatement en aval de celle-ci (.i.e. avant le routeur de bordure). Dans ce cas, celui-ci doit être sur une route symétrique (les acquittements et les segments de données sont reçus par le nœud) et le RTT du flot mesuré dans le nœud est similaire à celui de la connexion TCP. Enfin, la contrainte de fonctionnement porte sur des mesures passives faites en direct sur le flot TCP et non sur l'analyse *a posteriori* des traces du flot TCP. Cette contrainte temporelle impose donc que la détection utilise uniquement les traces du côté émetteur.

Le principe développé par la méthode ICN consiste à déterminer les CE qui affectent un flot TCP par l'observation de son activité. L'observation des actions de TCP est faite à partir de la capture des segments du flot TCP. Comme TCP est un protocole de transport fiable, les données perdues sont retransmises. Un CE se déduit de l'observation des retransmissions dans le flot TCP. Cependant, toutes les retransmissions n'indiquent pas une perte et toutes les pertes ne signalent pas un CE. Quand le temporisateur de retransmission expire, TCP entame une procédure de reprise de type *go back* N, il se produit alors des retransmissions de segments déjà reçus. Il est aussi connu que TCP peut déduire à tort des pertes de paquets suite à un déséquencement opéré par le réseau (*network-reordering*) ou suite à une augmentation importante du RTT. Dans ce dernier cas, l'acquittement arrive trop tard pour désarmer le temporisateur de retransmission. Celui-ci ayant déjà expiré et retransmis inutilement les données. Ce genre de situation est qualifiée de mauvaise expiration du temporisateur (*spurious timeout*). Ces fausses indications de pertes sont traduites au niveau du contrôle de congestion de TCP par une réduction du débit d'émission et au niveau du contrôle d'erreur par des retransmissions inutiles. Les travaux [SAR 05, BLA 04] visent à rendre TCP résistant aux retransmissions inutiles. Enfin, toutes les pertes ne sont pas à prendre en compte de par la définition même d'un CE. Dans le cas de pertes multiples au sein d'une même fenêtre d'émission, seule la première perte sert à l'identification d'un CE. Les autres pertes de la fenêtre sont à négliger. Pour cela il faut pouvoir détecter la première perte et la taille de la fenêtre d'émission.

En résumé, la mesure des CE pour être la plus exacte possible doit :

– prendre en compte la taille de la fenêtre afin de distinguer les pertes déclenchant un CE des pertes faisant partie de l'événement lui-même ;

– identifier les retransmissions inutiles qui ne doivent pas être interprétées comme des pertes ;

– traiter les retransmissions multiples des mêmes données. Ce genre de situation se produit dans les congestions sévères. Dans ce cas, les CE se succèdent, on qualifiera par la suite cette situation de re-congestion.

2.2. *De l'interprétation d'un événement de congestion*

Comme présenté en introduction, un événement de congestion consiste en une ou plusieurs pertes se produisant sur une période de temps correspondante au RTT. Sachant qu'une fenêtre de données est émise dans un RTT, connaître la fenêtre revient à identifier les données émises pendant un RTT. Lorsqu'il y a une retransmission, le bas de la fenêtre pointe sur le segment retransmis. Le haut de la fenêtre correspond au numéro de séquence des données émises le plus important. Les retransmissions entre ces deux points sont interprétées comme faisant partie du même CE. Un CE commence lorsqu'une perte est identifiée et se termine quand le haut de la fenêtre de données au moment de la notification de congestion est acquitté.

La détection des retransmissions ne suffit pas pour identifier une perte. Il faut pouvoir s'assurer que la retransmission ne s'est pas produite suite à une erreur de TCP. Ce cas doit être pris en compte pour ne pas surestimer les CE dans les routes perturbées par les déséquencements ou les fortes variations de RTT. Le principe d'identification des retransmissions repose donc sur un délai d'attente T avant validation nommé par la suite délai de validation. Dans l'article [REW 06], les auteurs classent la retransmission comme inutile si elle est acquittée dans un délai inférieur à une fraction du RTT_{min} ayant pour valeur 0.75. La valeur de la fraction a un rôle important dans la détection des CE. Plus le délai de validation est proche du RTT et moins il y a de chance de valider un CE, l'acquittement pourra être reçu et la retransmission sera considérée comme inutile. L'algorithme ICN dans cette configuration peut sous-estimer le nombre de CE. A l'inverse, avoir un délai de validation très faible peut conduire à interpréter des retransmissions inutiles comme des pertes. En somme, la détection d'un CE s'effectue en deux temps. Tout d'abord, l'identification d'une retransmission puis la validation d'une retransmission pour une perte. Il est à noter qu'il existe une latence entre la congestion dans le réseau et sa détection au niveau d'ICN. En effet, la détection d'un CE apparaît à l'instant de la validation qui est en retard par rapport à l'instant de la retransmission et qui est elle-même se produit après un délai (de l'ordre de un à quatre RTT selon la méthode de détection de la perte). Le retard de notification du au délai de validation est la contre partie pour palier au manque de fiabilité de la détection des pertes par TCP.

Le RFC 2581 [ALL 99] complète la définition du CE en précisant que la perte d'une retransmission doit être prise comme une seconde indication de congestion. Les bornes de la fenêtre des données émises doivent s'ajuster au moment de la seconde indication de congestion. Ce cas demande cependant à prendre quelques précautions dans la validation du second CE. En effet, avec la version TCP New Reno [FLO 04],

il existe plusieurs variantes dans la gestion du temporisateur de retransmission suite à un acquittement partiel. La variante recommandée dite *impatient* réarme uniquement le temporisateur sur le premier acquittement partiel. L'objectif est de limiter la durée du *fast recovery* lorsqu'il y a des pertes multiples. En effet le *fast recovery* en l'absence de SACK (*Selective Acknowledgment*) [MAT 96] ne peut corriger qu'une perte par RTT. Lorsque le temporisateur expire, le segment est retransmis et la suite de la reprise s'effectue au moyen du *slow-start*. Du point de vue du réseau, cette retransmission ne signifie pas qu'il y a eu une perte dans le réseau. Il faut donc identifier cette retransmission afin de la classer comme une fausse indication de congestion. Lorsque ces événements se produisent, la route du flot TCP est en congestion ou sort de la congestion. Le délai de validation T doit pouvoir prendre en compte cette situation afin d'identifier avec exactitude les retransmissions inutiles.

2.3. *L'algorithme Implicit Congestion Notification (ICN)*

A partir de l'observation des segments de données et des acquittements, les CE de chaque connexion sont déduits au moyen d'une machine d'état qui identifie la phase du contrôle de congestion et qui classe les retransmissions comme utiles ou inutiles. Le contrôle de congestion de TCP réagit selon des notifications (*feedback*) binaires du réseau : absence ou présence de la congestion. La machine d'état utilisée par ICN (1) compte donc deux états induites par les notifications :

– l'état normal qui caractérise une connexion TCP où les transmissions sont sans pertes. Suivant l'algorithme dit de Karn [KAR 87], des mesures de RTT peuvent être faites en séquence dans cet état. A savoir que pour un segment émis, le délai est chronométré jusqu'à réception de l'acquittement correspondant. Une fois la mesure faite, le processus recommence pour le prochain segment transmis. La mesure du RTT sert au dimensionnement du délai de validation. Pour les connexions qui démarrent sur un réseau congestionné, les pertes de segments empêchent de prendre des mesures du RTT. La valeur d'initialisation est arbitraire et reprend celle du temporisateur de retransmission à savoir 3 secondes [PAX 00] ;

– l'état de congestion qui commence à la perte du premier segment d'une fenêtre de données. A chaque fois que l'algorithme ICN rentre dans cet état, un CE est comptabilisé pour la connexion TCP.

Deux états temporaires sont ajoutés entre les états des deux phases. Ces états visent à identifier les retransmissions inutiles à l'aide du délai de validation comme indiqué précédemment. Enfin, lorsque le haut de fenêtre (noté *recover* sur la figure 1) est acquitté, ICN repasse dans un état normal.

Dans le cas d'utilisation du drapeau ECN, les congestions ne sont plus déduites uniquement des pertes. Pour ce type de connexion, la méthode ICN doit être étendue pour analyser la signalisation de congestion dans les acquittements TCP. Autrement, ICN sous-estimerait le taux de CE affectant la connexion TCP. L'étude [MED 05] précise que ECN est utilisé par 2,1 % des hôtes en 2004. En conséquence, nous avons

choisi de ne pas présenter ce cas que nous réservons pour une étude future. A noter qu'ICN ne traite en rien du contrôle d'erreur qui reste du ressort de TCP. Si des améliorations dans la décision de retransmission sont introduites dans TCP, la méthode proposée étant indépendante du contrôle d'erreur, elle reste donc valide. En conclusion, ICN se veut général aux versions de TCP qui reposent sur un contrôle de congestion dont la notification négative est une perte.

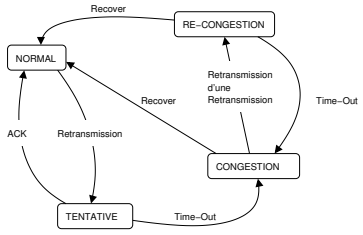


Figure 1. *Machine à état d'ICN*

3. Validation

Dans cette section, nous vérifions le principe de la méthode de détection des CE pour TCP et le délai de validation d'une perte. Notre objectif est d'estimer la fiabilité d'ICN. En d'autres termes, nous chercherons à déterminer quelle est l'erreur de notification d'un CE par rapport aux CE réels.

La validation du concept d'ICN est réalisée à l'aide de l'outil de simulation ns-2. La topologie utilisée est celle dite du papillon où n sources différentes sont connectées à n destinations différentes via un goulot d'étranglement entre deux routeurs. La bande passante et les temps de propagation de chaque lien sont respectivement de 1 Mbit/s et de 5ms. La bande passante du goulot d'étranglement peut être amenée à changer en fonction des études.Chaque source TCP src_i émet 400 paquets de taille égale à 1250 octets vers une destination dst_i. Le flot de paquets du couple (src_1, dst_1) forme la trace des segments TCP qui sont traités par ICN. Ce flot sera qualifié de flot de référence. Les $n-1$ autres couples source-destination génèrent un trafic transversal supplémentaire dans certains scénarios. Les n flots partagent les mêmes conditions de trafic (même profil de trafic et même RTT). Toutefois, ils démarrent en séquence dans un ordre aléatoire. La version de TCP retenue par défaut est celle dite New Reno [FLO 04]. L'objectif de ce modèle est de produire de la congestion pour le flot qui est analysé. L'intensité de la congestion est contrôlée par le nombre de flots en concurrence. Il est donc inutile d'agir sur le RTT des autres flots pour contrôler l'intensité de la congestion. La mesure de la précision de la détection des CE est faite à l'aide de l'erreur relative notée ϵ et qui se définie comme :

$$\epsilon = \frac{|N_{real} - N_{estimated}|}{N_{real}} \quad \text{avec } N \text{ le nombre de CE} \qquad [1]$$

La valeur $N_{estimated}$ s'applique au comptage fait par ICN ou par TCP. La mesure de référence N_{real} comptabilise les CE qui se sont réellement produits. Un CE réel se déduit selon les principes du CE d'ICN mais avec la connaissance certaine des pertes dans le réseau.

3.1. *Quel délai de validation ?*

Un élément important dans la prise de décision d'ICN réside dans son délai de validation T. Trois valeurs sont retenues pour son dimensionnement :

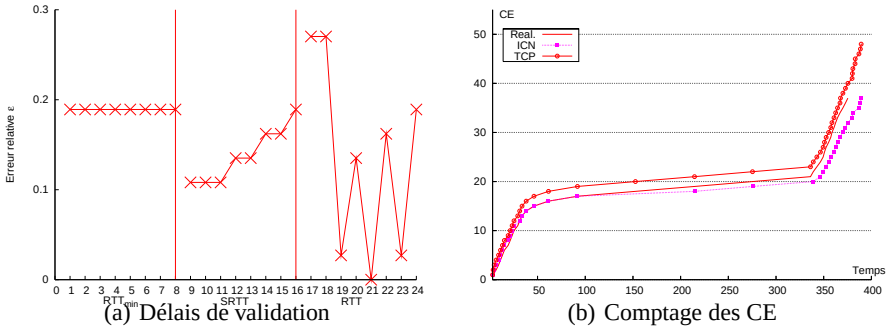
– RTT_{min}, celle-ci tend à attribuer une valeur faible ;

– $SRTT$ la moyenne exponentielle des mesures de RTT ;

– RTT issu de la dernière mesure, celle-ci tend à attribuer une valeur importante.

Ces valeurs sont ensuite pondérées par une fraction afin d'avoir une marge de sécurité. Cette marge vise à empêcher que l'expiration du temporisateur de validation du CE entre dans une période ou l'arrivée de l'ACK est probable. C'est à dire si l'acquittement de la retransmission déclencheur du CE est reçue pendant le délai de validation, le CE sera invalidé car la retransmission aura été classée comme abusive (à tort). La marge de sécurité traite donc de l'incertitude sur le RTT de la retransmission. Une marge de sécurité trop faible peut empêcher de détecter des CE et à l'inverse, avoir une marge de sécurité trop importante peut valider des CE sur des retransmissions inutiles. Les fractions testées sont $1, 0.75, 0.5, 0.25$. Dans le cas de re-congestion, le délai de validation sera testé avec ou sans pondération. La figure 2(a) présente l'erreur relative pour les 24 combinaisons possibles. Le scénario retenu porte sur 96 flots TCP. Ce qui assure une contention sévère des ressources. Les 8 premiers cas utilisent RTT_{min}, les cas 9 à 16 le $SRTT$ et les 8 derniers le RTT. Les valeurs de fractions sont évaluées dans l'ordre décroissant. Les cas numérotés par un nombre impair utilisent un délai de validation sans pondération pour les CE en situation de re-congestion. Dans ce scénario, le RTT_{min} est trop faible pour détecter les fausses notifications de CE. Avec $SRTT$, plus la pondération est faible, plus le nombre de faux CE augmente. Avec RTT, on remarque le rôle de l'absence de pondération pour valider les CE en situation de re-congestion. Ceci confirme notre analyse. Avec une pondération, le délai de validation est trop faible et des retransmissions inutiles déclenchent des faux CE. Pour le choix de la pondération, un minimum se dessine lorsque la fraction est de 0.5. Cette valeur donne un délai de détection assez rapide tout en préservant une marge de sécurité qui est équivalente au délai de validation. Par la suite, le délai de validation sera égal à $0.5 * RTT$ dans une situation normale et à RTT dans une situation de re-congestion.

La figure 2(b) montre l'évolution du comptage des CE pour le flot de référence dans le cas retenu (numéroté 21 sur la figure 2(a)). Au final, ICN a détecté 37 CE, TCP 48, alors qu'il y en avait 37 en réalité.



(a) Délais de validation (b) Comptage des CE

Figure 2.

3.2. *Analyse de la fiabilité*

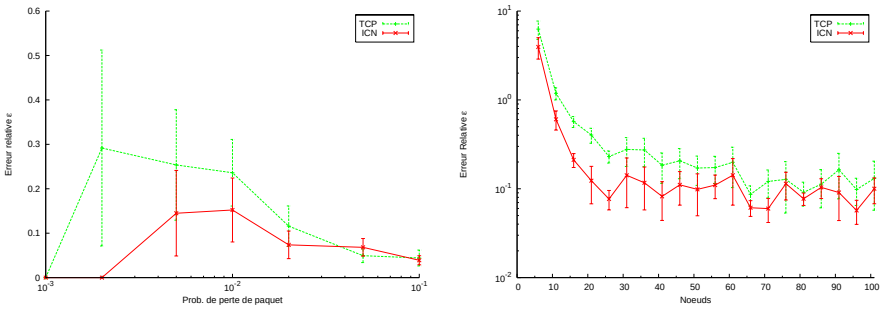
Le première étude évalue l'erreur relative de ICN en fonction du taux de pertes de paquets. A cette fin, un modèle de simulation est constitué avec un seul flot TCP pour lequel des paquets sont perdus avec un taux variant de 10^{-3} à 10^{-1} selon un modèle de perte uniforme. Les autres caractéristiques du modèle de simulation reprennent celles données en début de cette section.

La figure 3(a) montre les résultats de cette étude pour la version de TCP New Reno. Afin d'apprécier la précision des résultats, 30 réplications différentes sont faites pour chaque taux de pertes. L'intervalle de confiance à 95 % de l'erreur relative est aussi représenté. A propos du calcul de l'erreur relative sur l'estimation des CE détectés par TCP, une précaution est à prendre quant à la détermination du nombre de CE de TCP. Ce comptage est déduit du nombre de réductions de fenêtre et du nombre de retransmissions quand la fenêtre de congestion a déjà la taille minimum. Dans ce dernier cas, la perte d'un segment retransmis ne peut diminuer la taille de la fenêtre puisque la fenêtre de congestion a déjà la taille minimum. Cette précaution est intégrée dans le comptage les situations de re-congestion. La figure 3(a) soulève plusieurs remarques :

– lorsque le taux de pertes est important, le nombre de CE est important, l'erreur relative a tendance à diminuer. A l'inverse sur des taux faibles, une erreur d'estimation entraîne une erreur relative importante du fait du faible nombre de CE. Le démarrage de TCP est souvent source de retransmissions inutiles car une congestion en phase de *slow start* est plus enclin à produire des erreurs multiples. La connexion TCP n'a pas encore acquis l'auto-synchronisation ;

– en moyenne, l'erreur relative d'ICN reste inférieure à celle de TCP.

La seconde étude vise à introduire une contention croissante avec le flot de référence. La contention se traduit par des périodes de congestion qui entraînent des pertes causées par débordement de la file d'attente. Dans ce modèle, le RTT varie en fonction du niveau de remplissage de la file et les pertes se produisent quand le RTT atteint un sommet. La réalisation de ce modèle s'effectue en ajoutant un nombre de flots transversaux au flot de référence. La figure 3(b) montre l'évolution de l'erreur relative pour TCP New Reno et ICN lorsque le nombre de flots total varie de 6 à 101. En moyenne, l'erreur relative est importante lorsqu'il y a peu de flots à cause de démarrage de la connexion TCP comme expliqué précédemment. Plus la contention augmente, ICN converge vers TCP. La raison principale c'est qu'il y a de plus en plus de retransmissions inutiles sur expiration de temporisateur pour marquer la fin de la phase de *fast-recovery* de TCP New Reno. Ces retransmissions sont difficiles à détecter car l'ICN n'a plus de mesures de RTT dans l'état de congestion. Alors que le RTT a fortement augmenté. Le délai de validation n'est pas assez fort. Ce point reste encore à améliorer. Le problème ne provient pas d'un défaut de paramétrage d'ICN mais d'un défaut de mesure récente du RTT. Il est connu depuis [KAR 87] que les mesures de RTT en l'absence d'estampille temporelle ne peuvent être prise en compte lorsqu'un segment a été retransmis.



(a) Taux de pertes (b) Contention croissante

Figure 3.

3.3. *Retransmissions inutiles*

Bien que le modèle de simulation apporte quelques indications sur la méthode ICN. Il est difficile de reproduire la complexité et la diversité des situations qu'un flot TCP peut rencontrer. Le déséquencement introduit par le réseau est l'une de ces situations. Dans cette dernière, étude les pertes et les retards sont contrôlés afin de constituer un scénario avec un déséquencement du réseau produisant une retransmission inutile de TCP. La figure 4 montre l'émission et la réception d'un flot TCP New Reno dans lequel le segment 20 subit un retard suite un déséquencement du réseau et le segment 55 subit une double perte. Le segment 20 active un *fast retransmit* et

chaque acquittement partiel déclenche une retransmission inutile. Dans cette situation particulière, ICN détecte la première retransmission qu'il classe comme inutile. Ce qui est intéressant dans ce cas, c'est que ICN est résistant à la fausse détection d'un CE sur les segments retransmis sur acquittement partiel [FLO 04]. Cependant ICN détecte les 2 CE qui se produisent plus tard. Au final, TCP aura souffert d'une fausse indication de CE en plus alors qu'ICN aura pu rester fidèle à la réalité. Cet exemple n'est certes pas exhaustif mais il montre le comportement d'ICN dans un cas connu.

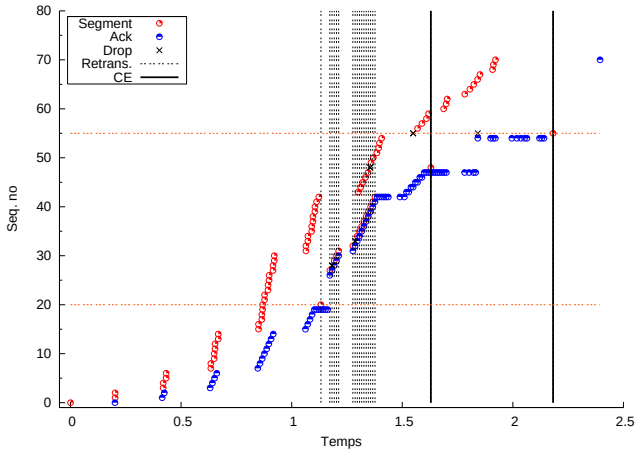


Figure 4. *CE identifié*

4. Conclusion

Dans cet article, nous avons présenté un algorithme qui détecte de façon plus fiable les événements de congestion de TCP. Cette détection est effectuée de manière passive en bordure du réseau. Cependant, cet algorithme doit encore être testé dans des conditions réelles pour apprécier ses caractéristiques de mise en oeuvre et ses capacités d'usage. Nous chercherons dans cette mise en oeuvre à quantifier le coût du délai inhérent à cette détection. Enfin, ICN gagnerait à être complété par un algorithme de discrimination de perte (*Loss Discrimination Algorithm*) afin de distinguer les pertes relatives à la congestion des pertes de corruption.

5. Bibliographie

[ALL 99] ALLMAN M., PAXSON V., STEVENS W., « TCP Congestion Control », Request For Comments n° 2581, 1999, IETF.

[ALL 03] ALLMAN M., EDDY W., OSTERMANN S., « Estimating Loss Rates With TCP », *ACM SIGMETRICS Performance Evaluation Review*, vol. 31, n° 3, 2003, p. 12-24.

[BAL 05] BALI S., JIN Y., FROST V., DUNCAN T., « Characterizing user-perceived impairment events using end-to-end measurements », *International Journal of Communication Systems*, vol. 18, n° 10, 2005, p. 935–960, John Wiley and Sons Ltd.

[BEN 02] BENKO P., VERES A., « A Passive Method for Estimating End-to-End TCP Packet Loss », *Proc. of IEEE GLOBECOM*, novembre 2002.

[BLA 04] BLANTON E., ALLMAN M., « Using TCP Duplicate Selective Acknowledgement (DSACKs) and Stream Control Transmission Protocol (SCTP) Duplicate Transmission Sequence Numbers (TSNs) to Detect Spurious Retransmissions », Request For Comments n° 3708, février 2004, IETF.

[FLO 03] FLOYD S., « HighSpeed TCP for Large Congestion Windows », Request For Comments n° 3649, décembre 2003, IETF.

[FLO 04] FLOYD S., HENDERSON T., « The NewReno Modification to TCP's Fast Recovery Algorithm », Request For Comments n° 3782, avril 2004, IETF.

[FLO 06] FLOYD S., KOHLER E., PADHYE J., « Profile for DCCP Congestion Control ID 3 : TRFC Congestion Control », Request For Comments n° 4342, mars 2006, IETF.

[FLO 07] FLOYD S., « Metrics for the Evaluation of Congestion Control Mechanisms », Internet Draft n° draft-irtf-tmrg-metrics-11.txt, octobre 2007, IETF.

[HAN 03] HANDLEY M., FLOYD S., PAHDYE J., WIDMER J., « TCP-Friendly Rate Control (TFRC) : Protocol Specification », Request For Comments n° 3448, janvier 2003, IETF.

[KAR 87] KARN P., PARTRIDGE C., « Improving round-trip time estimates in reliable transport protocols », *ACM Computer Communications Review*, vol. 17, n° 5, 1987, p. 2–7.

[KOH 06] KOHLER E., HANDLEY M., FLOYD S., « Datagram Congestion Control Protocol (DCCP) », Request For Comments n° 4340, mars 2006, IETF.

[MAT 96] MATHIS M., MAHDAVI J., FLOYD S., ROMANOW A., « TCP Selective Acknowledgment Options », Request For Comments n° 2018, octobre 1996, IETF.

[MED 05] MEDINA A., ALLMAN M., FLOYD S., « Measuring the Evolution of Transport Protocols in the Internet », *Computer Communication Review*, vol. 35, n° 2, 2005.

[MEL 02] MELLIA M., CARPANI A., CIGNO R., « Measuring IP and TCP behavior on Edge Nodes », *Proc. of IEEE GLOBECOM*, novembre 2002.

[PAX 00] PAXSON V., ALLMAN M., « Computing TCP's Retransmission Timer », Request For Comments n° 2988, novembre 2000, IETF.

[RAM 01] RAMAKRISHNAN K., FLOYD S., BLACK D., « The Addition of Explicit Congestion Notification (ECN) to IP », Request For Comments n° 3168, septembre 2001, IETF.

[REW 06] REWASKAR S., KAUR J., SMITH F., « A Passive State-Machine Approach for Accurate Analysis of TCP Out-of-Sequence Segments », *ACM Computer Communications Review*, vol. 36, n° 3, 2006, p. 51-64.

[SAR 05] SAROLAHTI P., KOJO M., « Forward RTO-Recovery (F-RTO) : An Algorithm for Detecting Spurious Retransmission Timeouts with TCP and the Stream Control Transmission Protocol (SCTP) », Request For Comments n° 4138, août 2005, IETF.