Analyse de données de simulation

Approche du TP :

L'objective de ce TP est de collecter un ensemble de résultats immédiats en se basant sur une trace de simulation qui est un échéancier contenant une liste d'événements survenus pendant l'exécution. On a donc recours au langage C comme utilitaire d'analyse de la trace. L'analyseur en C devrait être capable d'effectuer des calcules statistiques tout au long du 'parsing' de la trace afin d'en tirer le maximum d'informations intéressantes en un minimum de temps.

Analyse des données globales de la trace :

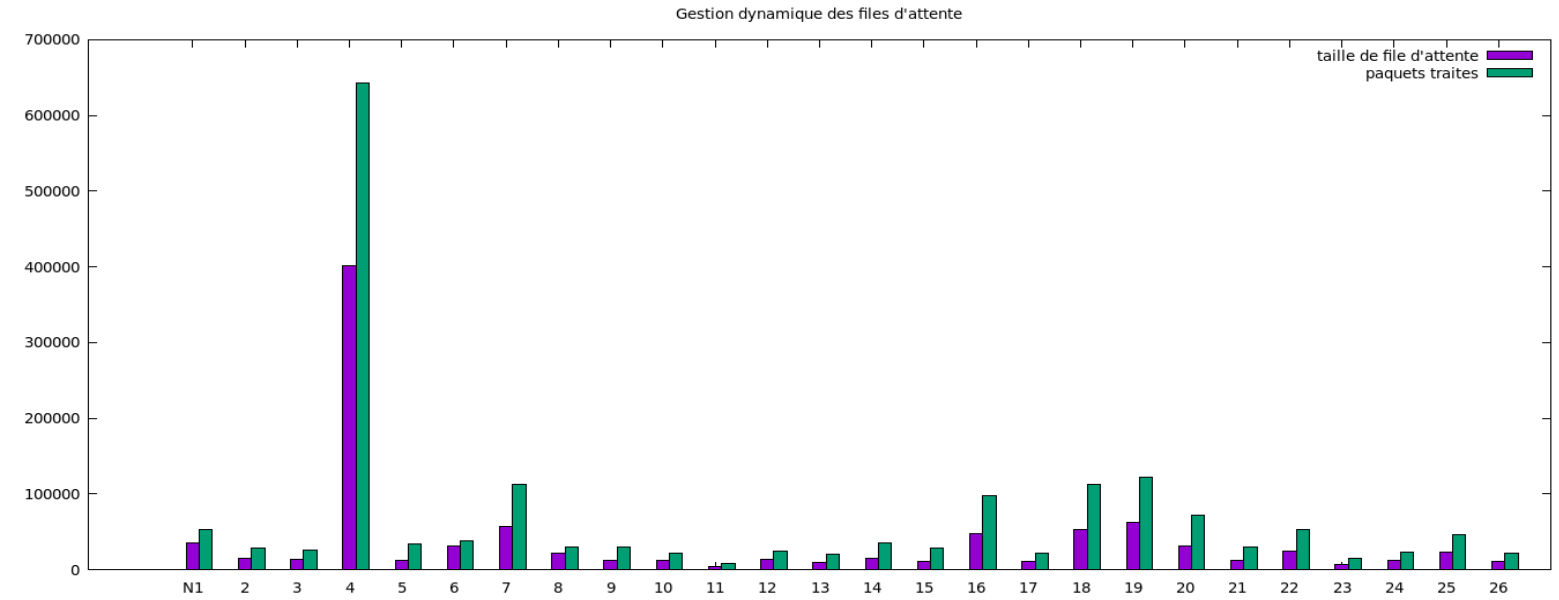
Cet le mode le plus simple à obtenir fournissant des résultats standards de la trace. L'utilitaire en C fait le « parsing» de l'intégralité du fichier et en déduire des données globales. Le fichier source responsable d'affichage des données globales est: «global_data.c». La commande pour l'exécution est :

./global_data <file_to_trace>.

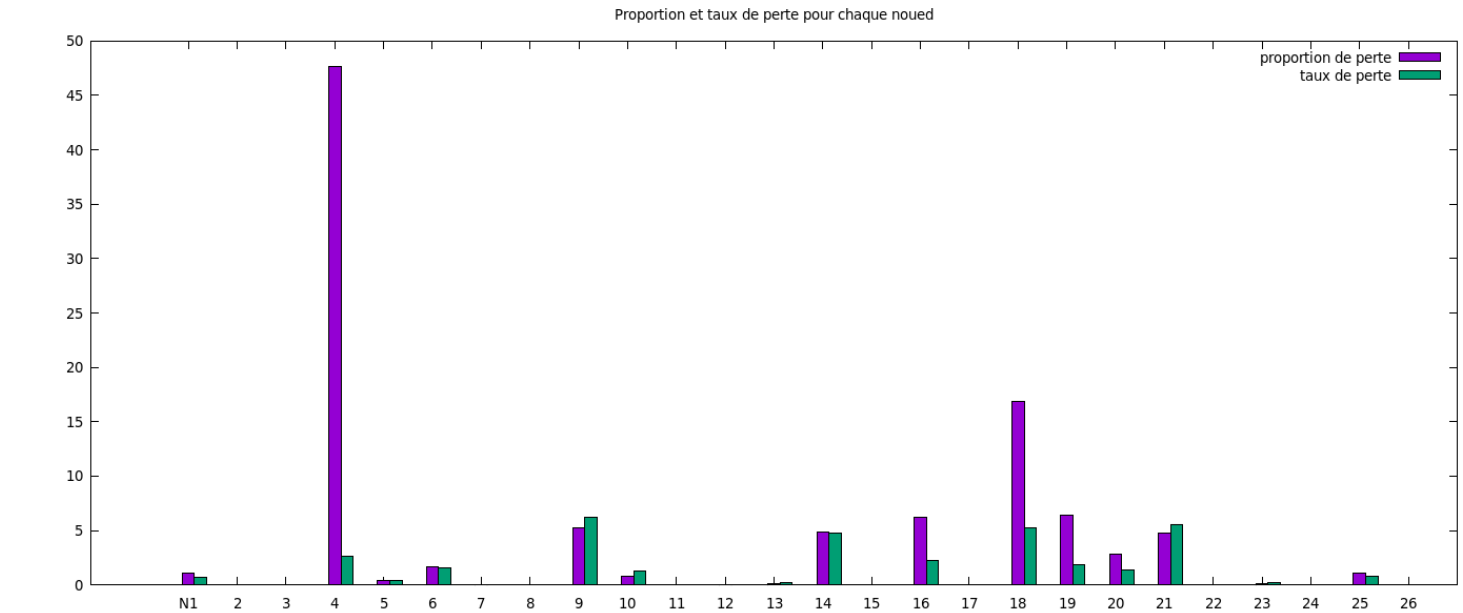
Le parsing du fichier de la trace fournie qui est de taille 136Mo avec 3.502.252 événements prend moyennement 3 à 4 secondes et le fichier n'est parcouru qu'une seule fois. Chaque nœud a ses propres informations définies dans la structure *node_infos* dans le fichier include « Ip_Network». Cette une structure basé sur les arbres bicolores (RB_TREE) qui est un cas particulier d'arbre binaire de recherche dans lequel chaque nœud a un attribut supplémentaire: sa couleur, qui est soit **rouge** soit **noire (Red/Black).**
J'ai utilisé cette structure pour rendre le programme performant au niveau d'insertion,la recherche et enfin la suppression. En effet durant le parsing du fichier, à chaque ligne lu, on doit mettre à jour les infos du nœud en cours, implique aller rechercher ce nœud dans l'arbre bicolore(RB_Node) en utilisant sa clé particulière fournie durant son insertion , et cette recherche se fait 3.502.252 fois (nombre de lignes) et donc le besoin d'un arbre de recherche. Notons bien que les opérations faites par les arbres bicolores se font en O($\log(n)$).

```
abdellah@debian:~/Bureau/EPTP2/Abdellah$ ./global_data trace2650.txt
Le nombre de paquets totale émis: 750979
Le nombre de paquets totale recus 715815
Le nombre de paquets totale perdus: 35164

Noued   Nbr-paquets_perdues     Prop_paquets_perdus     Taux de perte    Taille file d'attente  paquets traités
N1              365             1.037993                0.684572                35964           53318
N2              0               0.000000                0.000000                15336           28432
N3              0               0.000000                0.000000                13093           26470
N4              16776           47.707883               2.607860                401799          643286
N5              134             0.381072                0.395630                12976           33870
N6              575             1.635195                1.519556                30791           37840
N7              14              0.039813                0.012346                57606           113398
N8              0               0.000000                0.000000                22394           29909
N9              1832            5.209874                6.193374                12090           29580
N10             290             0.824707                1.293142                12172           22426
N11             0               0.000000                0.000000                4584            8767
N12             4               0.011375                0.016523                14133           24209
N13             34              0.096690                0.169627                10039           20044
N14             1691            4.808895                4.763514                14595           35499
N15             0               0.000000                0.000000                10459           28704
N16             2173            6.179616                2.209568                47919           98345
N17             0               0.000000                0.000000                10420           21322
N18             5925            16.849619               5.257694                53444           112692
N19             2263            6.435559                1.854568                62873           122023
N20             1000            2.843818                1.381444                31168           72388
N21             1687            4.797520                5.579811                12077           30234
N22             0               0.000000                0.000000                25160           53182
N23             29              0.082471                0.199862                7116            14510
N24             0               0.000000                0.000000                12197           22498
N25             372             1.057900                0.804864                23844           46219
N26             0               0.000000                0.000000                10734           21961

Temps pour l'affichage des données globales: 3.246972 s
```

L’exécutable,après un durée d’exécution de 3,24s, indique que 750979 ont été émis dans le réseau, dont 35164 ont été détruites. On remarque aussi que la taille de fille d’attente n’est pas uniforme pour tous les routeurs. En effet, la gestion active des files d’attentes est nécessaire pour contrôler les tailles moyennes globales de file d’attente,de telle sorte que les paquets arrivants puissent être traités sans abandon de paquet. Pour cela, il faut varier la taille du file d’attente d’un nœud en se basant sur le nombre de paquets qu’il traite. Pour comprendre ce phénomène on regarde la figure suivante :



On remarque bien qu’on adapte la taille du file d’attente au nombre du paquet traités durant l’état flux à flux afin d’assurer une file d’attente équitable **(Fair Queueing).** Ainsi, le taux de perte dans chaque routeur sera presque le même dans tous le réseaux IP. Ceci se voit bien dans la figure suivante, notons bien qu’il faut pas confondre proportion de perte avec le taux de perte. En effet, la proportion de perte pour le nœud 4 est de 48% mais çà veut dire pas à priori qu’il présente un défaut dans notre réseau,puisqu’il a aussi un taux de perte locale faible de 2,6% comme tous les autre nœuds. Ceci est du au faite que le nœud 4 traitent beaucoup de paquets(643286) et donc le nombre de paquets qu’il va perdre sera bien grande par rapport aux autre nœuds, ce qui justifie le 48% de proportion de perte. On peut donc conclure que le nœud 4 est un nœud centrale dans notre réseau implique que la bonne gestion de ce nœud pourrait influencer grandiosement sur notre réseau.



Tracer d’un paquet ou d’un flux :

C’est le mode qui concerne un seul paquet ou un seul flux. Le fichier source responsable pour le tracer des paquets ou flux est «tracer.c». On utilise aussi les arbres bicolores pour stocker tous les informations de chaque paquet et chaque flux comme on a fait pour les routeurs. La commande pour afficher la trace d’un paquet ou d’un flux est :

```
./tracer <trace_file> <matrice_adj> < -p | -f >
```

On note bien qu’on construit l’arbre qui contient les infos de tous les paquets ou bien les flux, après on pourrait choisir le(s) pid ou fid pour en afficher les infos.
Le tracage d’un paquet donne des résultats sur sa source,sa destination,sa taille,les routeurs qu’il a traversé,l’attente(Quand et ou),transmission(quand et ou),les temps d’attente,et enfin les temps d’arrivée.

Un exemple d’affichage pour les paquets est le suivant :

```
abdellah@debian:~/Bureau/EPTP2/Abdellah$ ./tracer trace2650.txt res26.txt -p
Parsing de la matrice res26.txt..........
Construction de l'arbre bicolore contenant les infos de tous les paquets dans le fichier trace2650.txt
Temps pour la construction de l'arbre des paquets: 4.268994 s
Entrer le pid du paquet a tracer ou 0 pour quitter:  8998
Source: N10
destination: N19
La taille est: 4272
Le paquet 8998 est dans la file d'attente du routeur N4 a l'instant: 0.865452 et traverse le routeur N4 a l'instant:0.866520
Le paquet 8998 est dans la file d'attente du routeur N19 a l'instant: 0.866520 et traverse le routeur N19 a l'instant:0.866825
Le temps d'attente du paquet 8998 est:  +0.000000s(N4) +0.000000s(N19) = 0.000000s
Le temps de transmission du paquet 8998 est:  +0.001068s(lien N10-N4) +0.000305s(lien N4-N19) = 0.001373s
Le paquet arrive a 0.866825
Entrer le pid du paquet a tracer ou 0 pour quitter:  4975
Source: N2
destination: N4
La taille est: 1579
Le paquet 4975 est dans la file d'attente du routeur N6 a l'instant: 0.499558 et traverse le routeur N6 a l'instant:0.499822
Le paquet 4975 est dans la file d'attente du routeur N4 a l'instant: 0.499822 et traverse le routeur N4 a l'instant:0.500085
Le temps d'attente du paquet 4975 est:  +0.000000s(N6) +0.000000s(N4) = 0.000000s
Le temps de transmission du paquet 4975 est:  +0.000263s(lien N2-N6) +0.000263s(lien N6-N4) = 0.000526s
Le paquet arrive a 0.500085
Entrer le pid du paquet a tracer ou 0 pour quitter:
```

« Tracer de deux paquets de pid=8998 et pid=4975 »

Pour les flux, c’est la même démarche avec ‘-f’ comme option dans la commande. Le traçage d’un flux donne des résultats comme le nombre totale de paquets émis,reçus,perdus ainsi son taux de perte et durée de vie. Un exemple d’affichage est le suivant :

```
abdellah@debian:~/Bureau/EPTP2/Abdellah$ ./tracer trace2650.txt res26.txt -f
Construction de l'arbre bicolore contenant les infos de tous les flux dans le fichier trace2650.txt
Temps pour la construction de l'arbre des flux : 3.483985 s
Entrer le fid du flux a afficher infos ou -1 pour quitter:  2
Total des paquets : 425
Nombre de paquets recu : 385
Nombre de paquets détruits : 40
Taux de perte 10.389610 pourcent
La durée de vie du flux est 7.811773s(dead) - 0.003683s(born) = 7.808090s
Entrer le fid du flux a afficher infos ou -1 pour quitter:  1800
Total des paquets : 286
Nombre de paquets recu : 254
Nombre de paquets détruits : 32
Taux de perte 12.598425 pourcent
La durée de vie du flux est 15.070643s(dead) - 6.043870s(born) = 9.026773s
Entrer le fid du flux a afficher infos ou -1 pour quitter:
```

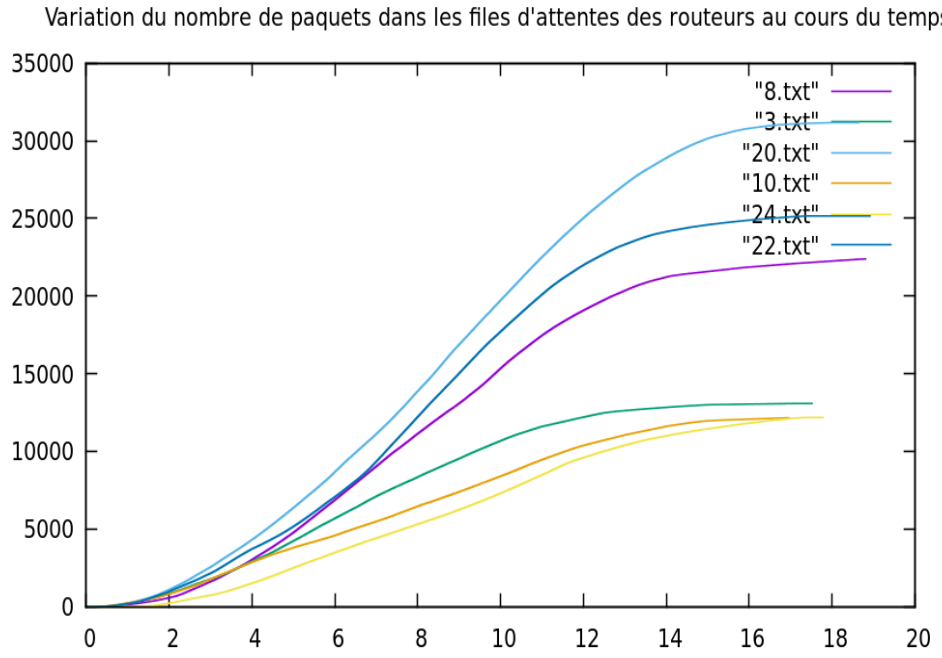
« Tracer de deux flux de fid=2 et fid=1800 »

Remarquons que la durée du vie du flux de fid:1800 à un durée de vie grande (9s) par rapport à celle du flux de fid 2 (7,8) même si il a émit un nombre de paquets (286) inférieur à celle du flux fid=2 (425 paquets). Ceci est à cause des périodes de warmup et cooldown qu'on détaillera après.

Échantillonnage :

C'est le mode le plus important durant l'analyse des données. Il permet d'évaluer les performances du réseaux et donne une évolution précises du réseau Ip. Et donc repérer les congestions.

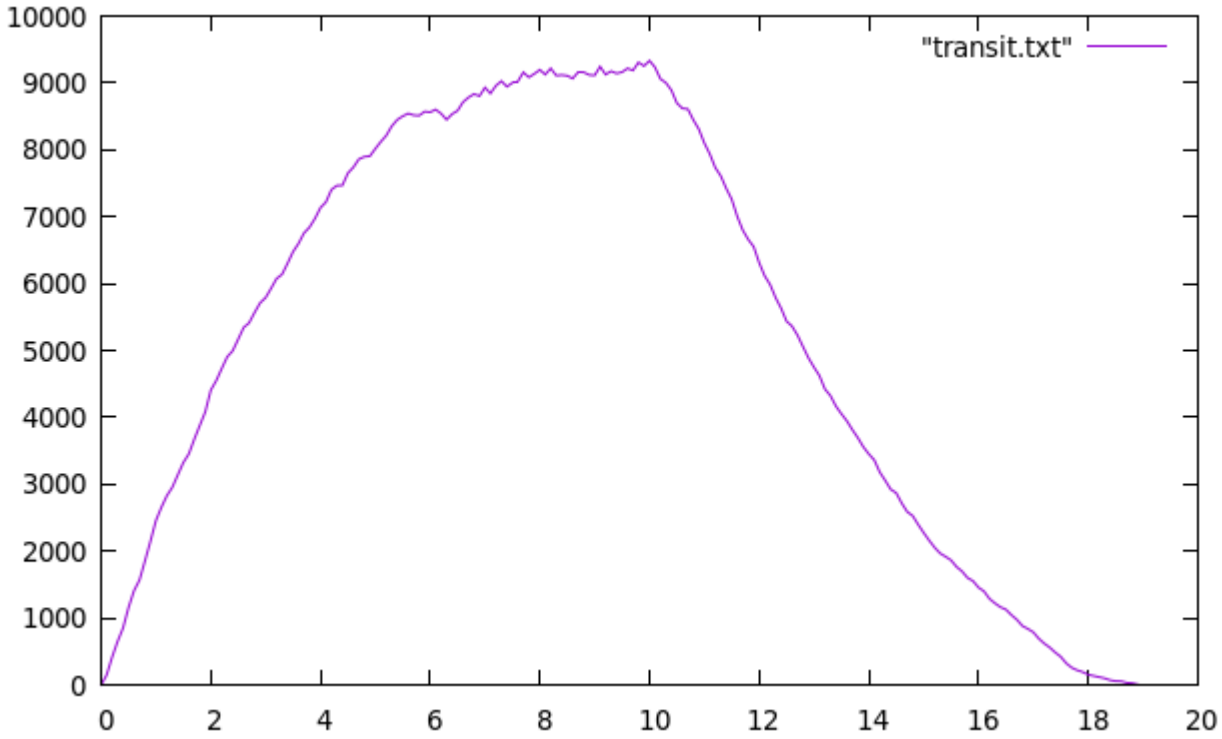
On commence tout d'abord par étudier le comportement des files d'attentes en représentant graphiquement le nombre de paquets en attente au cours du temps.



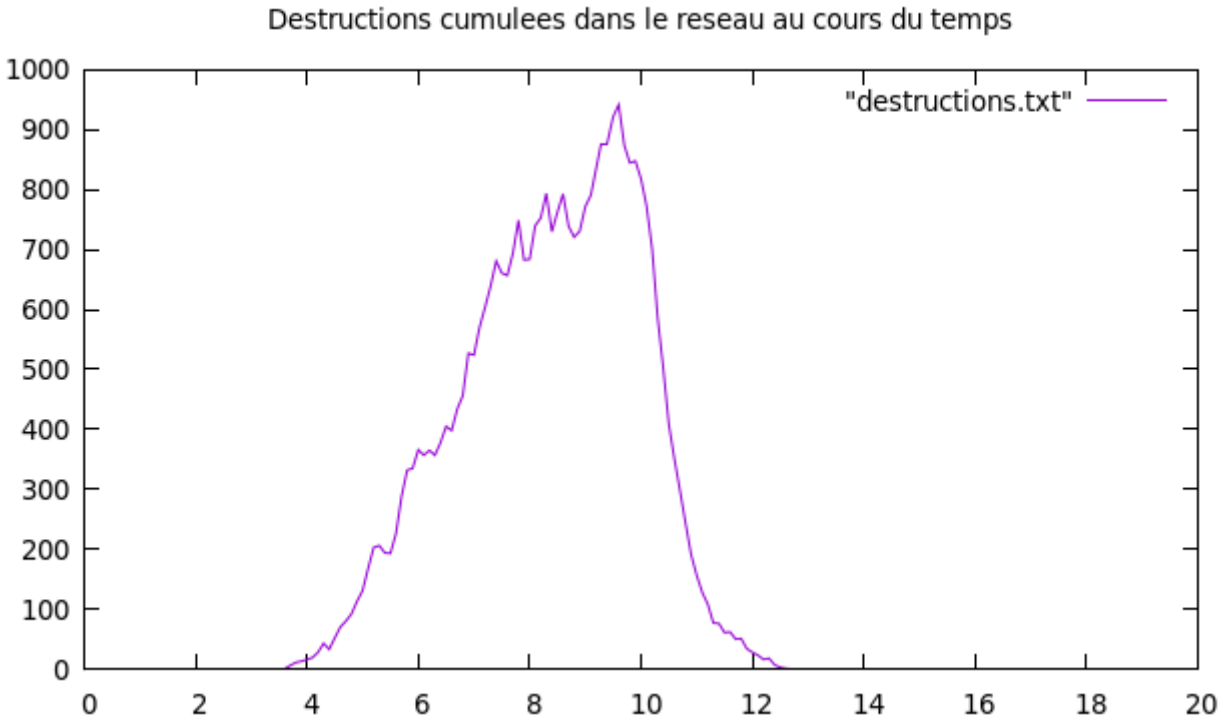
« Le nombre de paquets dans les files d'attentes des Noeuds 8,3,20,10,24,22 »

Tout d'abord, on peut déduire la taille des files d'attentes graphiquement en constatant que la courbe ne croit pas à un certain moment, implique que la file est saturée. On voit bien que tous les routeurs ont le même comportement en ce qui concerne le remplissage de leurs files d'attentes, ceci montre bien qu'il y'a une synchronisation de tous les émetteurs, car ils émettent tous de la même façon, et donc le même algorithme TCP. On constate ausssi qu'Il y'a des problèmes de congestion due `a des rafales de paquets à partir de l'instant 12 s.Les trames entrantes dans les buffers des routeurs sont rejetées dans ce cas.

Pour bien vérifier la congestion au niveau de notre réseau, on trace le nombre de paquets en transit dans le réseau au cours du temps. On choisit comme temps d'échantillonnage : 0,1s.

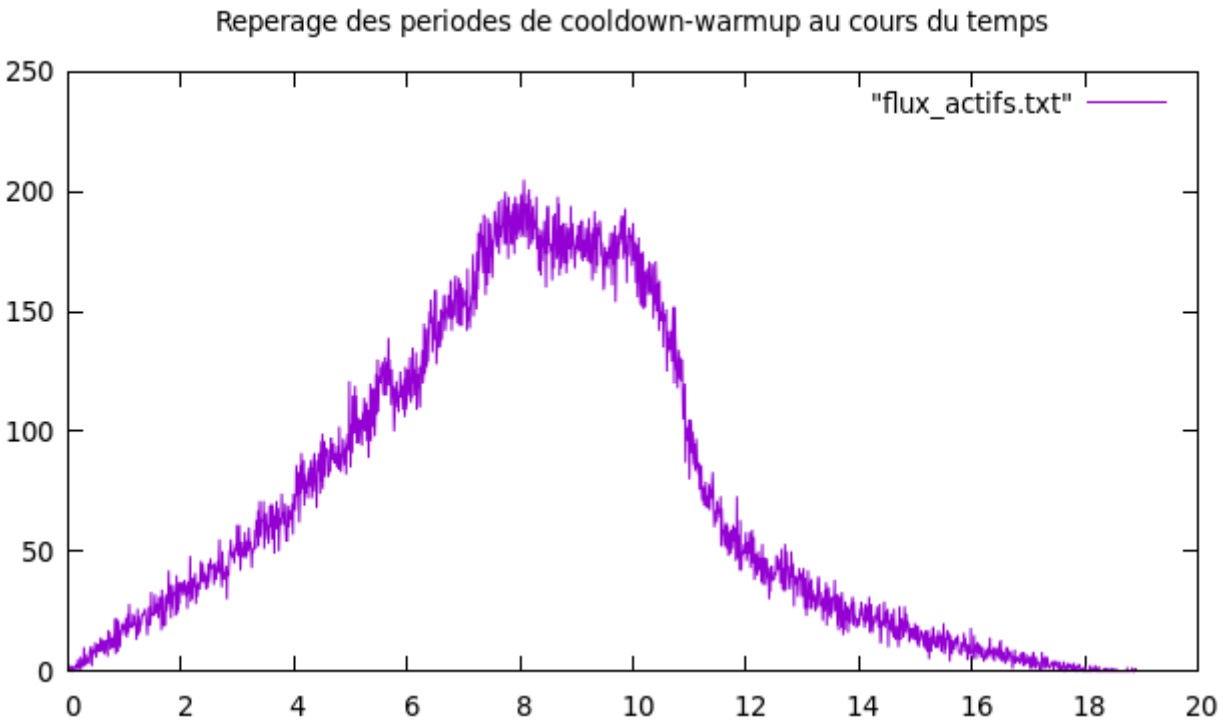


Le trafic croit d'une façon lente au début (Slow Start),jusqu'à l'instant t=10s, après, il commence à décroître. Ce qui prouve bien comme montré avant qu'il y'a une congestion causé par une grande perte de paquets quelque part dans le réseau. Afin d'étudier la cause, on a recours encore a l'échantillonnage du nombre de paquets détruits dans le réseau IP.



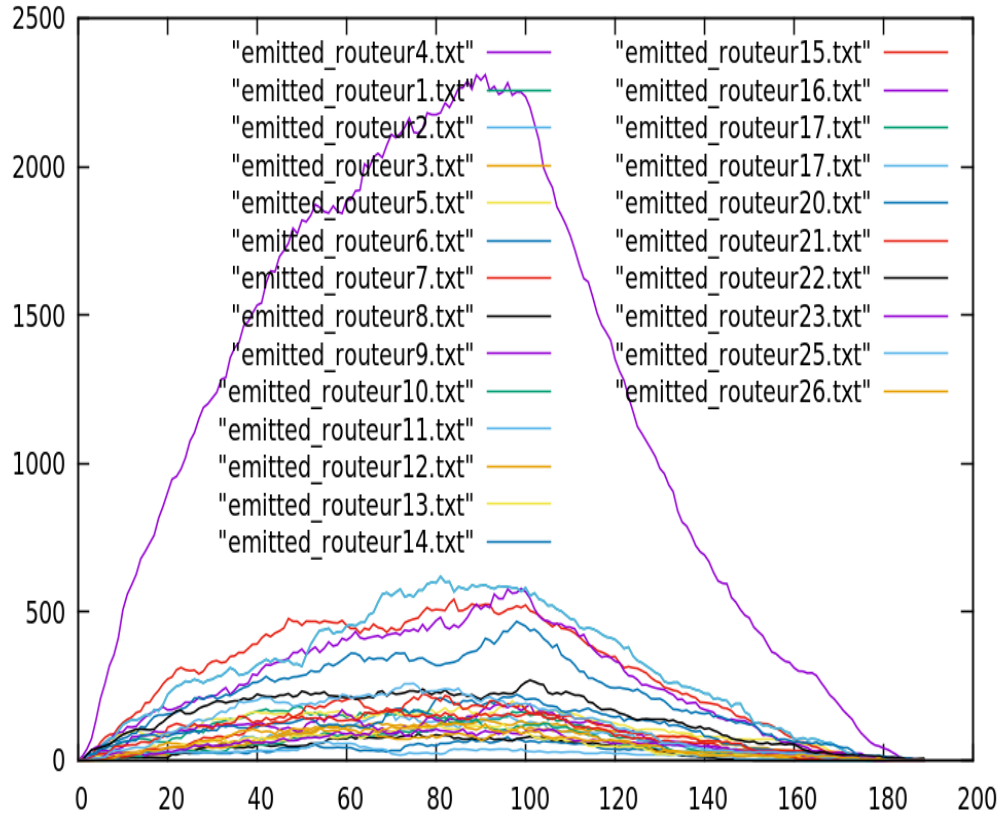
Parfaitement, le nombre de paquets cumulatives détruits dans le réseau passe de 100 paquets détruits chaque 0,1s (1000 paquets/s) à l'instant t=5 , a 950 paquets détruits chaque 0,1s (9500 paquets/s) à l'instant t=10s.

Un autre aspect important est de tracer le nombre de flux actifs au cours du temps. On prend le temps d'échantillonnage:0,1s. Ceci permet de repérer les périodes de Warmup et Cooldown.



On remarque que la phase de Warmup dure jusqu'à t=8s et la phase de cooldown commence a t=10s. C'est deux périodes permet au réseau Ip de se stabiliser, puisque entre les deux, il y'a la période critique ou les flux commence à se chevaucher, vu qu'il y'a presque 180 flux/0,1s (implique 1800flux/s). On pourrait donc dire que le **WARMUP** est la période pour laquelle tous les flux courent vers, pour commencer à envoyer les paquets , et le cooldown joue le rôle d'amortisseur après que les flux entre dans une phase de latence(Dure 2s dans notre réseau) ce qui augmente leur durée de vie .

Qui dit congestion,dit le plus de données injectées dans le réseau que le réseau n'accepte. Afin de voir le nœud responsable a l'injection de plus de nombre de paquets au cours du temps, on échantillonne le nombre de paquets transmis par tous les nœuds dans des temps d'échantillons de 0,1s comme d'habitude.



« Nombre de paquets injéctés dans le réseau par tous les noeuds »

On voit bien que le nœud numéro 4 n'est pas seulement un noeud centrale,mais émet également plus de paquets dans le réseau. Ainsi, à priori il est le responsable de la congestion au niveau de notre réseau. Les autres nœuds mettent du temps a détecter la congestion. On peut donc théoriser que le nœud 4 est effectivement un nœud centrale dans la topologie, par ses grand nombre de paquets qui transitent dans le réseau.