LAVIE
Pierre-louis
M1 RISE

RAPPORT NS :
TP 2

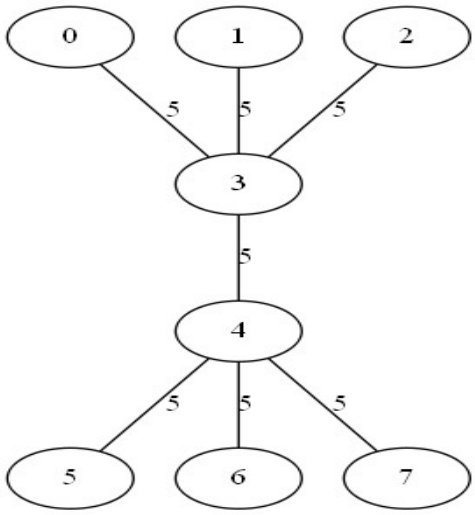
Exercice1

Les 93 nœuds sont générés . Il existe un flux UDP de type CBR entre chaque couple qui n'est pas issu du même groupe . Les files d'attentes sont bien visibles.

Exercice 2 Congestion TCP

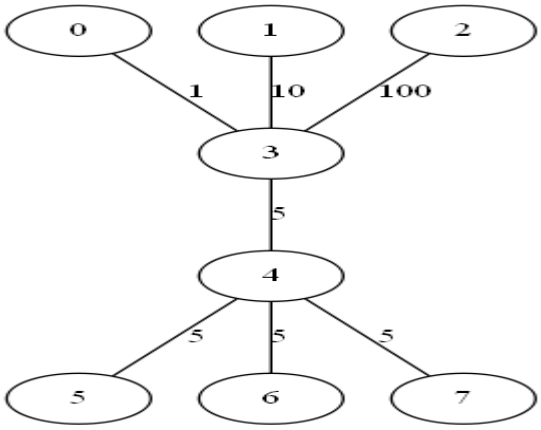
Topologie , paramètres

La topologie du réseau est la suivante :



(Les valeurs des arcs correspondent aux RTT)

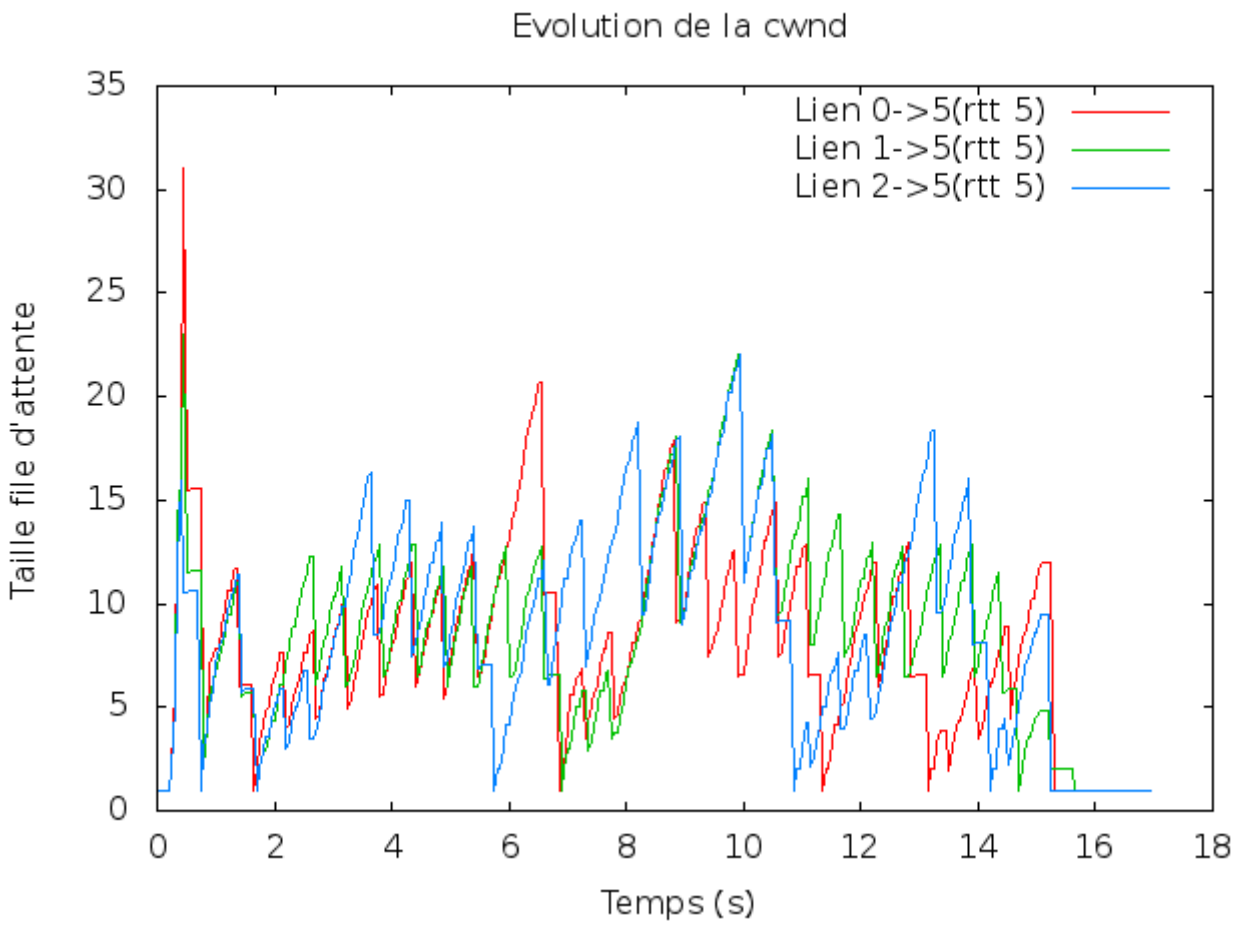
Les agents TCP sont de type Reno entre chaque couple (0,1,2) et (5,6,7).Il y a création d'un phénomène de congestion sur le nœud 3 car le débit de tout les liens est de 10 Mb .L'évolution de la taille de la file d'attente sur le nœud 3 est visible .
Les files sont de types DropTail,les paquets sont de taille 1500 (MTU).J'ai étudié l'évolution de la taille de la cwnd en fonction du rtt en comparant les résultats de la topologie précédente et celle-ci.



(Les valeurs sur les arcs correspondent aux RTT)

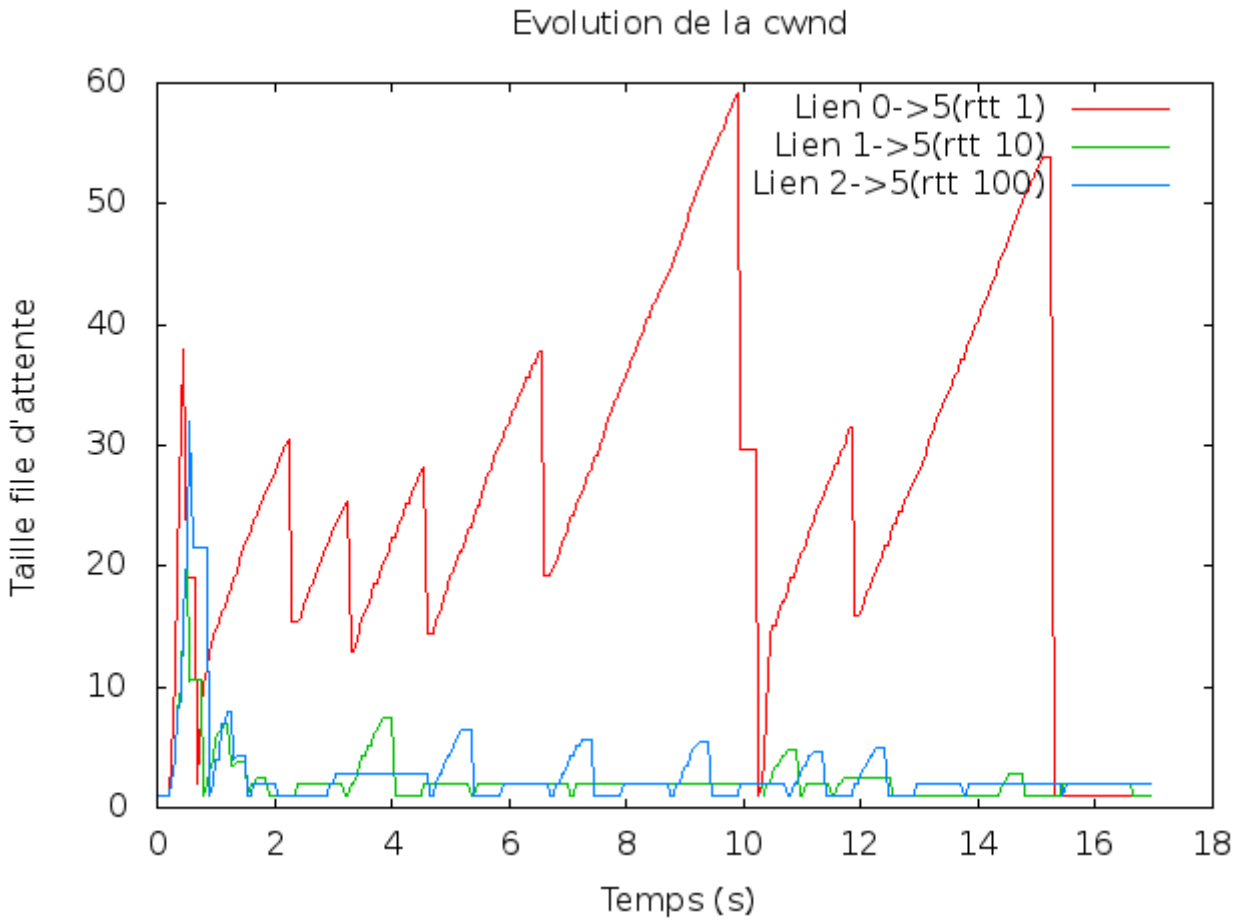
Pour les comparer je génère 2 graphes en prenant pour x le temps et en y la taille de la cwnd pour les agents TCP (0 → 5, 1 → 5 et 2 → 5).Les valeurs x et y sont obtenues à l'aide de la fonction proc traceGrapheCwnd {} qui inscrit dans un fichier le temps ainsi que les valeurs des cwnd.

Graphes



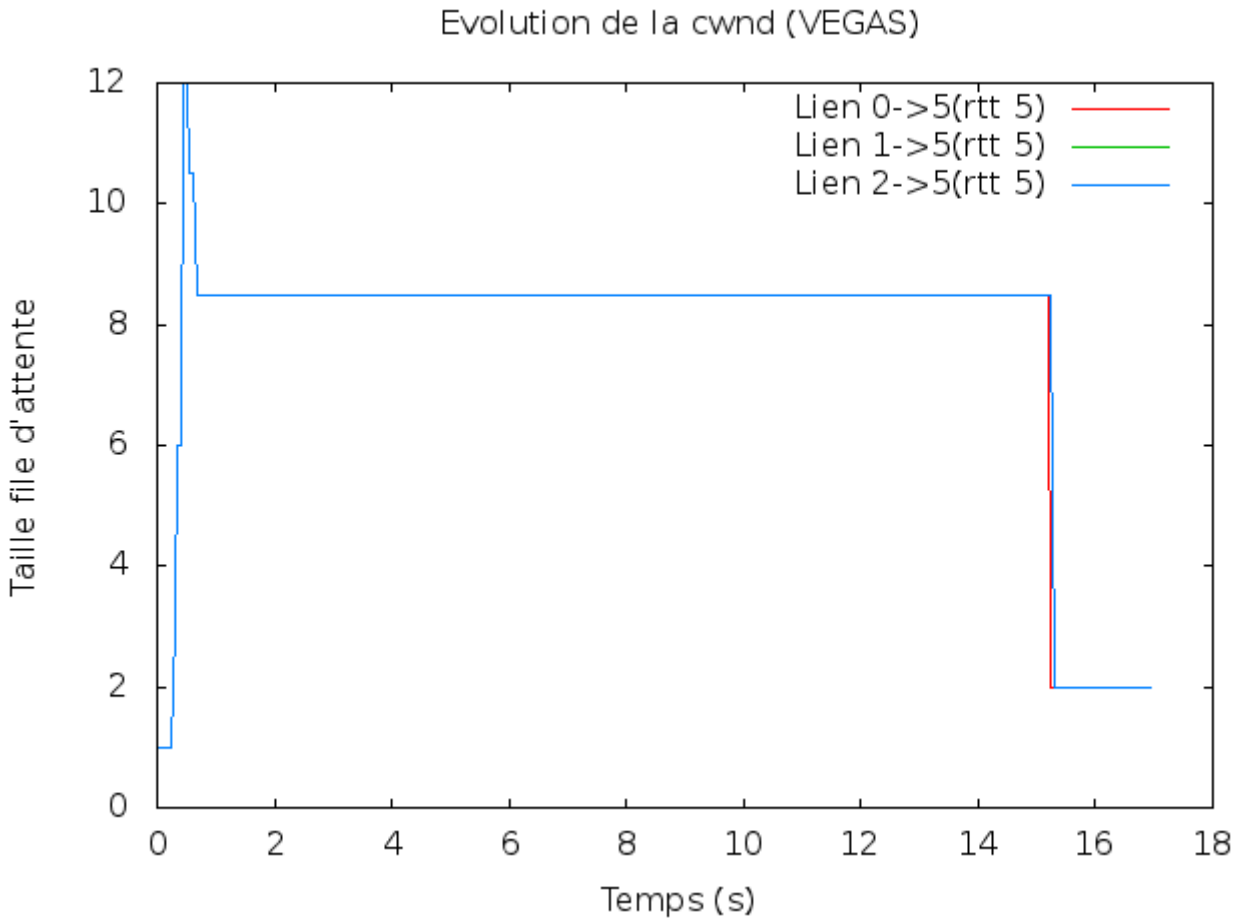
Évolution de la cwnd (RTT EGAUX)

Lorsque les RTT sont égaux les cwnd ont tendances a être égales et l'on observe le comportement en dents de scie classique de TCP . Les agents augmente tous leurs cwnd jusqu'à perte .Lors d'une perte cette dernière est divisé par deux.On constate que cela reste relativement équilibré . J'utilise ici un temps de simulation très court pour obtenir un graphe lisible .Dans la mesure ou la congestion est atteinte très vite je pense que la période de warm-up est négligable .



Évolution de la cwnd (RTT INEGAUX)

Lorsque les RTT sont très différents (1 , 10 ,100) alors la répartition est très injuste . Le lien avec le rtt le plus petit va prendre toute la bande passante .



Évolution de la cwnd (RTT EGAUX)

Ce graphe est obtenu en utilisant TCP Vegas . Vegas à pour but d'éviter les pertes par congestion et on remarque ici un comportement bien différent . Au lieu d'augmenter constamment la cwnd et donc aboutir forcément à des pertes , Vegas va ajuster sa cwnd en fonction des RTT .
On obtient donc une équité parfaite quasiment aucune pertes . (Vegas paraît très intéressant dans ce cas puisqu'il ne pose aucun problème d'équité mais il est très peu utile en pratique puisque mis en concurrence avec Reno par exemple il va accepter de diminuer sa cwnd et va donc se faire "dévorer" par les autres algorithmes).

Robustesse

Pour résoudre le problème de robustesse de notre réseau il faut supprimer le goulet d'étranglement .
J'ai donc choisi de relier toutes les feuilles aux 2 nœuds du goulet et non pas à une des extrémités .
Ainsi quelque soit le nœud qui disparaît le graphe restant est connexe .