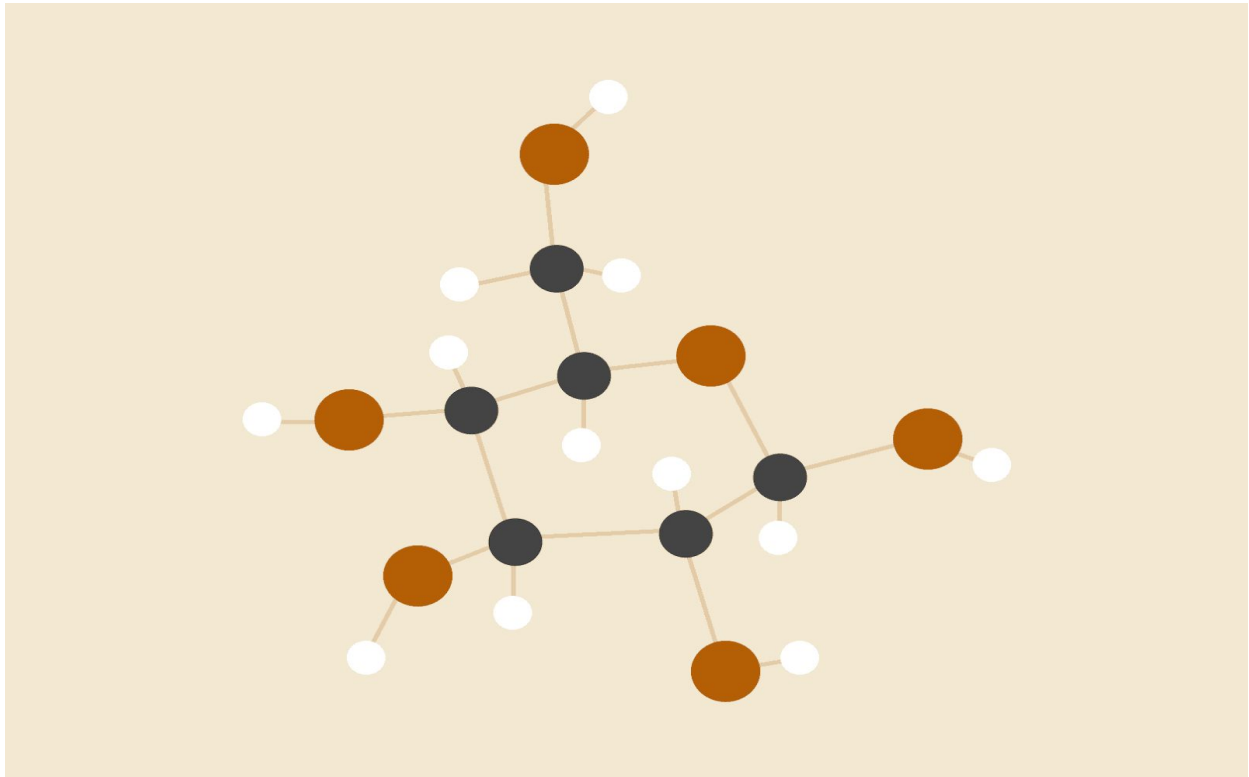


Université

de Strasbourg

Rapport Network Simulator 2

Partie 3



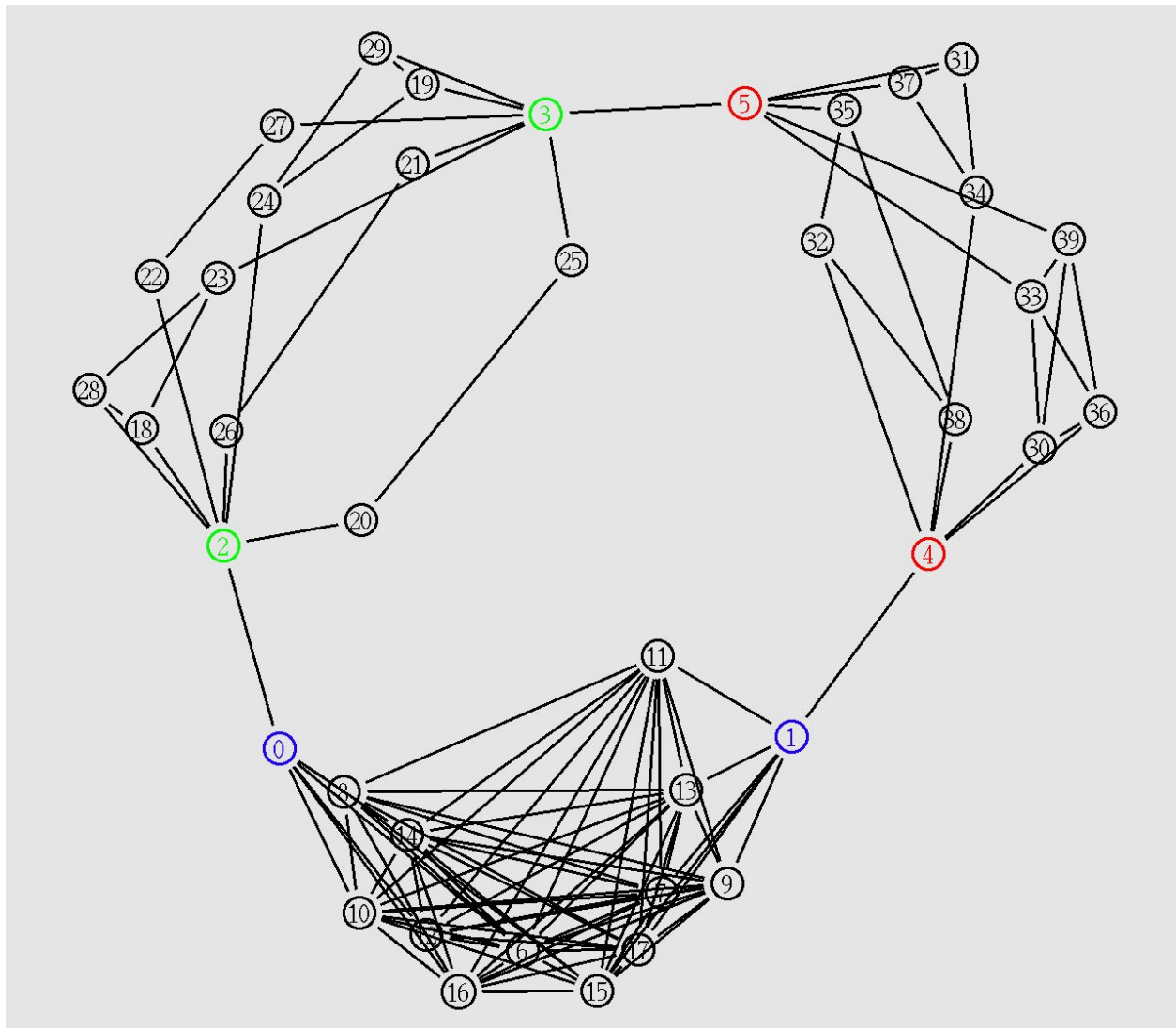
Seyyid-Ahmed LAHMER et Philippe CHEVALIER

01/01/2020

TOPOLOGIE RÉSEAU

Notre topologie représente trois domaines distincts. Les deux premiers ont 12 nœuds et le troisième a 10 nœuds, pour un total de 34 nœuds. Chacun de ces domaines a deux liens vers les deux autres, pour un total de trois liens critiques.

Voici cette topologie en image :



Les liens normaux ont une latence de 20 ms et un débit de 500Mb/s. Les liens critiques ont, quant à eux, un débit de 250Mb/s.

Nous admettons volontiers qu'il n'est pas réaliste d'avoir des liens critiques d'une capacité inférieure aux liens normaux, mais nous souhaitons nous assurer que les pertes ne se produisent qu'au niveau des points critiques.

MATRICE DE TRAFIC ET GÉNÉRATION DES DONNÉES

La matrice de trafic a été générée avec une distribution uniforme. En moyenne, chaque "case", ou plutôt chaque couple départ et arrivé émet 5 Mo (hormis ceux qui n'émettent rien).

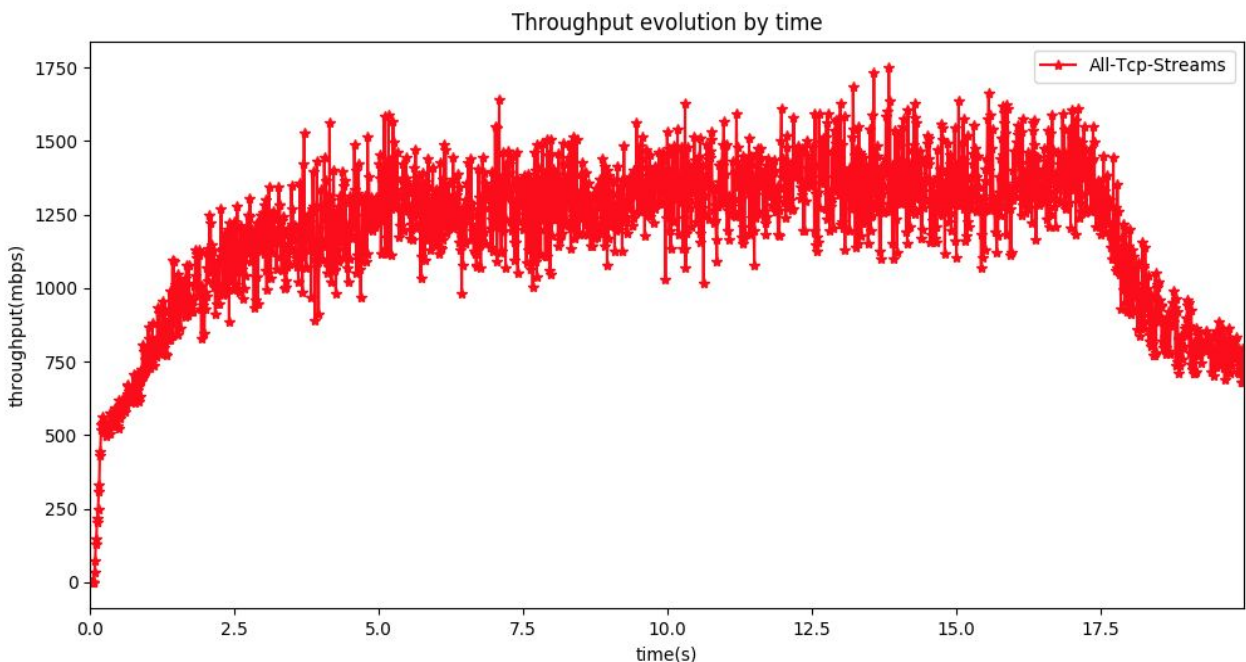
Notre réseau est conçu pour que 80% des données soient générées par des flux ON / OFF, et pour que les 20% restants soient générés par des flux TCP dont les tailles suivent une loi exponentielle avec un paramètre $k = \ln(4) / \ln(5)$, c'est-à-dire un Pareto 20 - 80.

La section suivante a pour objectif, entre autres, de prouver de manière empirique que ces paramètres sont respectés.

ANALYSE DE LA TOPOLOGIE RÉSEAU

Période de warm-up et cool-down

Voici une analyse du débit utile au cours du temps. Nous considérerons que la période de “warm-up” s’arrête avant la **6^e seconde** et que la période de “cool-down” débute à la **17^e seconde** :



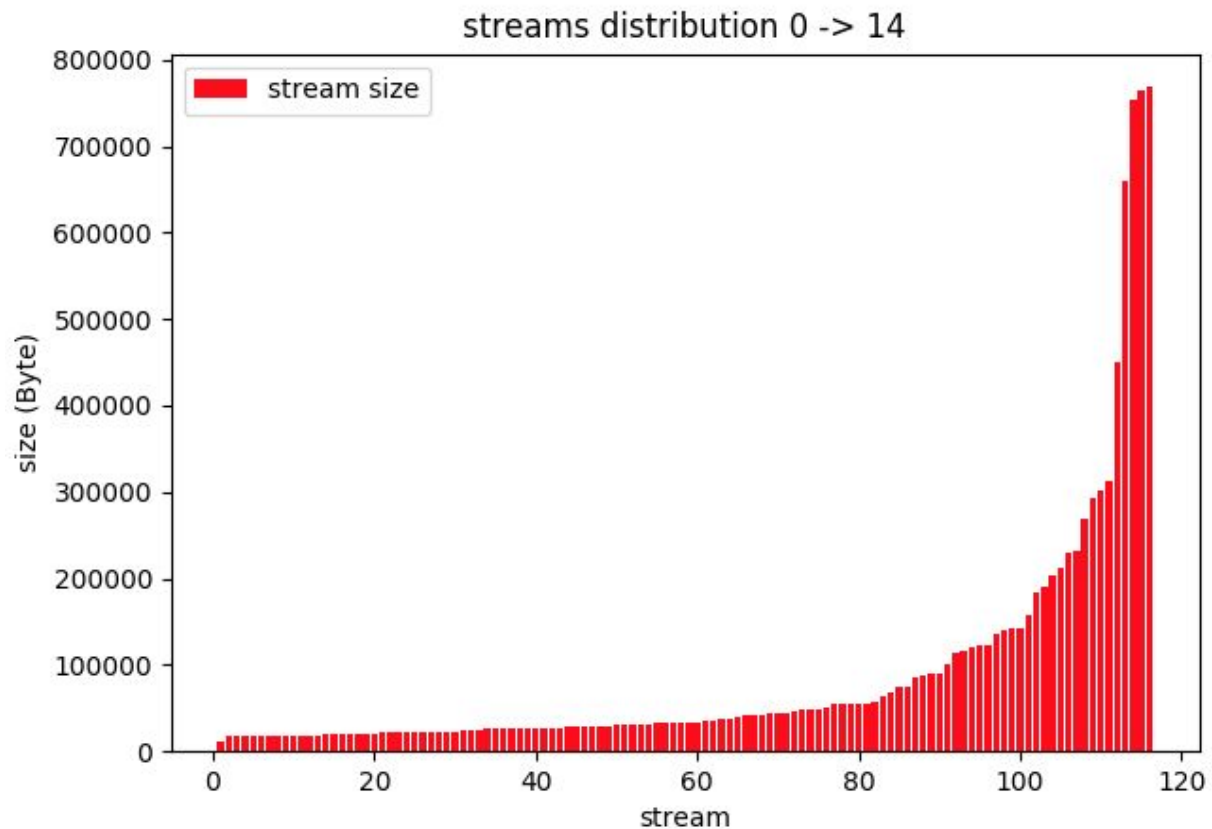
Pertes de paquets

En utilisant le visualiseur de simulation Network Animator (nam), nous constatons de nombreuses pertes de paquets tout au long de la simulation sur l’ensemble des liens critiques. Comme prévu, nous n’avons constaté aucune perte en dehors de ceux-ci.

Distribution de la taille des flux TCP

On vérifie maintenant que les flux TCP suivent bien une distribution de Pareto. On génère un graphe de la taille de tous les flux entre une source et une destination. Nous avons testé plusieurs sources / destinations, et nous avons pu confirmer qu'elles suivent une loi de puissance.

Voici le graphe d'un de ces tests, entre le noeud 0 et 14 :



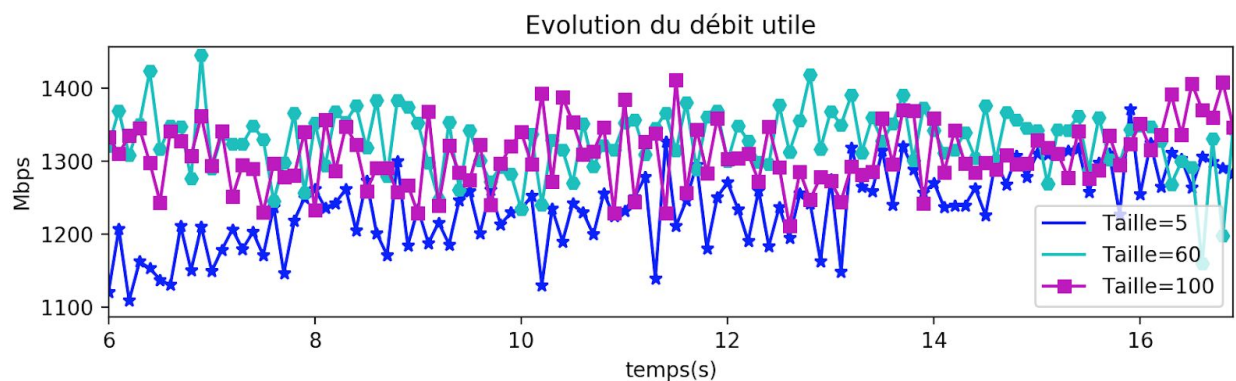
REMARQUE POUR LES SECTIONS SUIVANTES

Lors de tous les tests qui suivront, les tailles d'échantillons seront supérieures à 100. Selon la loi des grands nombres et le théorème central limite (et en supposant que les données soient indépendantes et identiquement distribuées) les moyennes des échantillons convergent vers une loi normale, nous avons donc pu calculer sereinement les écarts-types puis les intervalles de confiance. (Pour bien faire les choses, il aurait fallu tester la dépendance des données en utilisant des outils statistiques.)

INFLUENCES DE LA TAILLE DES FILES D'ATTENTE

Nous cherchons à savoir l'importance de la taille des files d'attente sur le réseau, et notamment sur le débit utile. Les graphes ont été générés avec Python (sauf si précisé autrement) et les moyennes, les écarts-types et les intervalles de confiance ont été calculés avec R.

Voici un graphe de la performance de trois tailles différentes :



Nous détaillerons ces mesures (+ trois autres tailles) dans les sections suivantes. On observe toutefois sans surprise que la petite taille de cinq est nettement désavantagée.

La taille de 5

Avec cette petite taille, la moyenne est de 1238,0 Mbps et l'écart-type est de 54,3 Mbps. L'intervalle de confiance à 95 % est donc [1227,8 1248,3].

La taille de 10

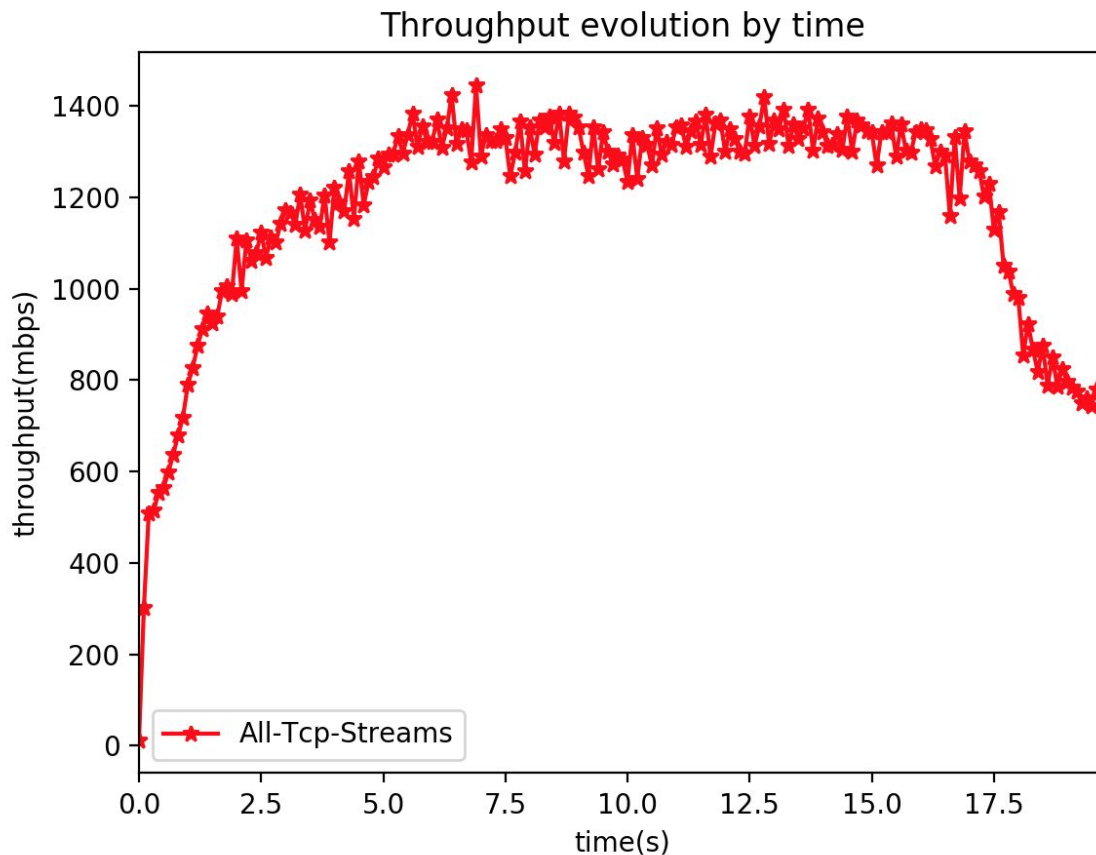
Avec cette configuration, le débit moyen utile est de 1290,0 Mbps ; l'écart-type est de 48,9 Mbps. L'intervalle de confiance à 95 % est donc [1280,8 1299,3].

La taille de 30

Cette fois-ci, le débit moyen utile est de 1321,6 Mbps ; l'écart-type est de 62,3 Mbps. L'intervalle de confiance à 95 % est donc [1309,8 1333,4].

La taille de 60

Voici un graphe du débit utile au cours du temps, avec une taille de files d'attente de soixante :



La moyenne est de 1326,9 Mbps ; l'écart-type est de 44,3 Mbps. L'intervalle de confiance à 95 % est donc [1318,5 1335,3].

La taille de 100

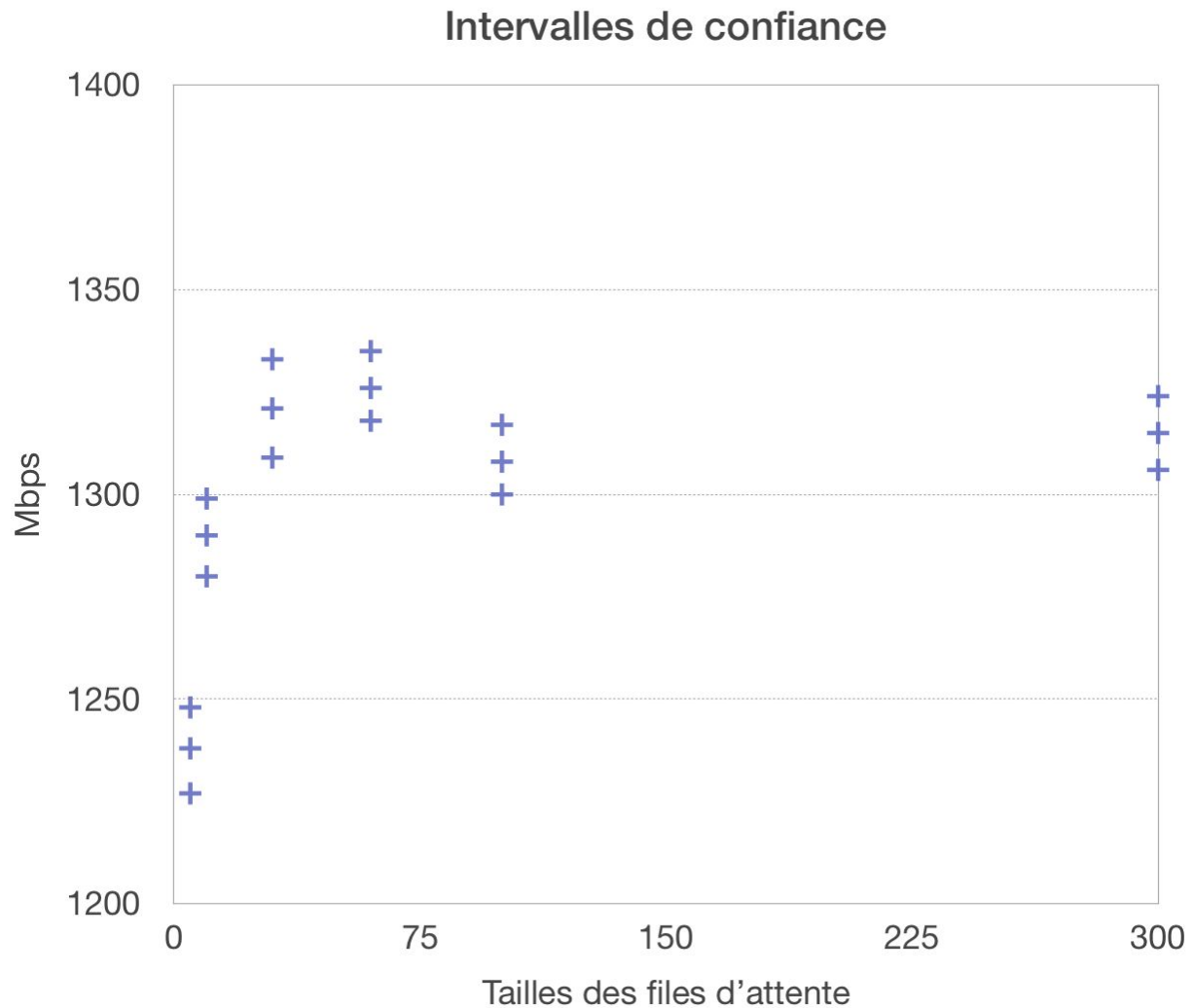
Avec cette configuration, le débit moyen utile est de 1308,8 Mbps ; l'écart-type est de 44,3 Mbps. L'intervalle de confiance à 95 % est donc [1300,4 1317,1].

La taille de 300

Avec cette grande taille, la moyenne est de 1315,3 Mbps et l'écart-type est de 47,4 Mbps. L'intervalle de confiance à 95 % est donc [1306,4 1324,3].

Analyse des performances des différentes tailles

Voici les intervalles de confiance résumés en un graphe (créé avec Numbers) :

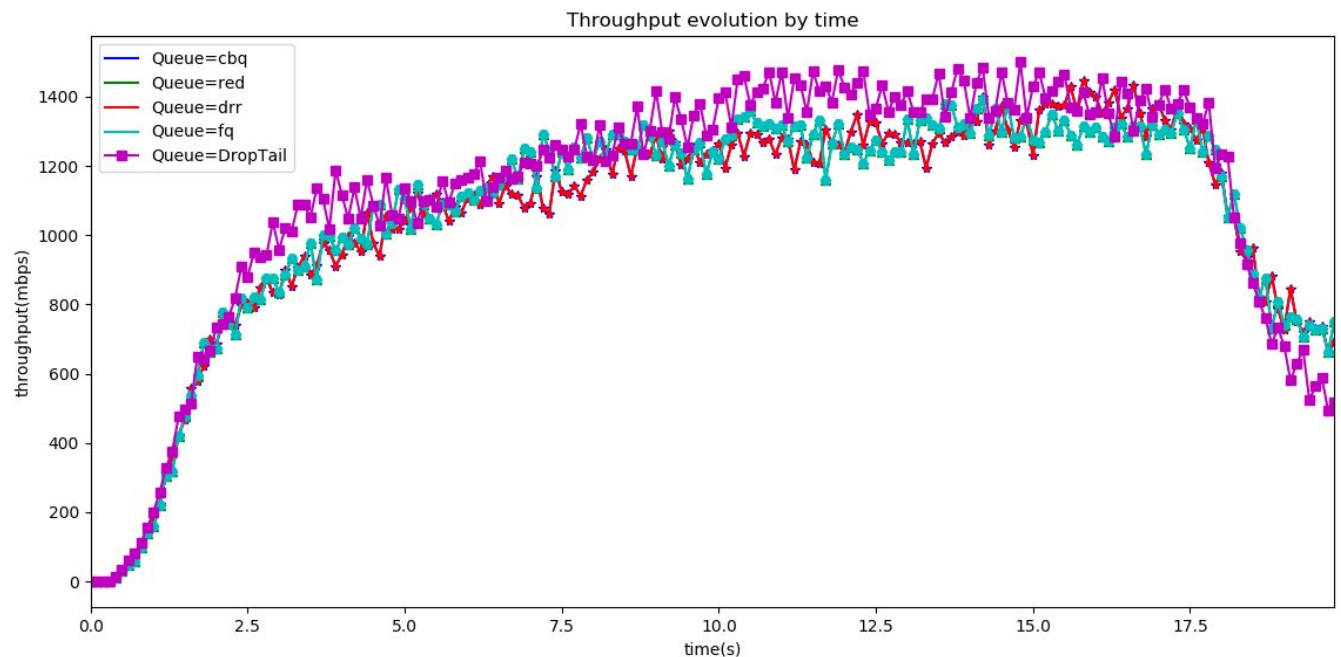


L'augmentation du débit utile est très nette de 5 à 10, puis de 10 à 30. Ensuite, pour les tailles subséquentes, l'augmentation est beaucoup moins claire, car les intervalles de confiance se chevauchent. On observe tout de même un pic à la taille 60 puis une légère régression sur les tailles supérieures.

INFLUENCES DES TYPES DE FILES D'ATTENTE

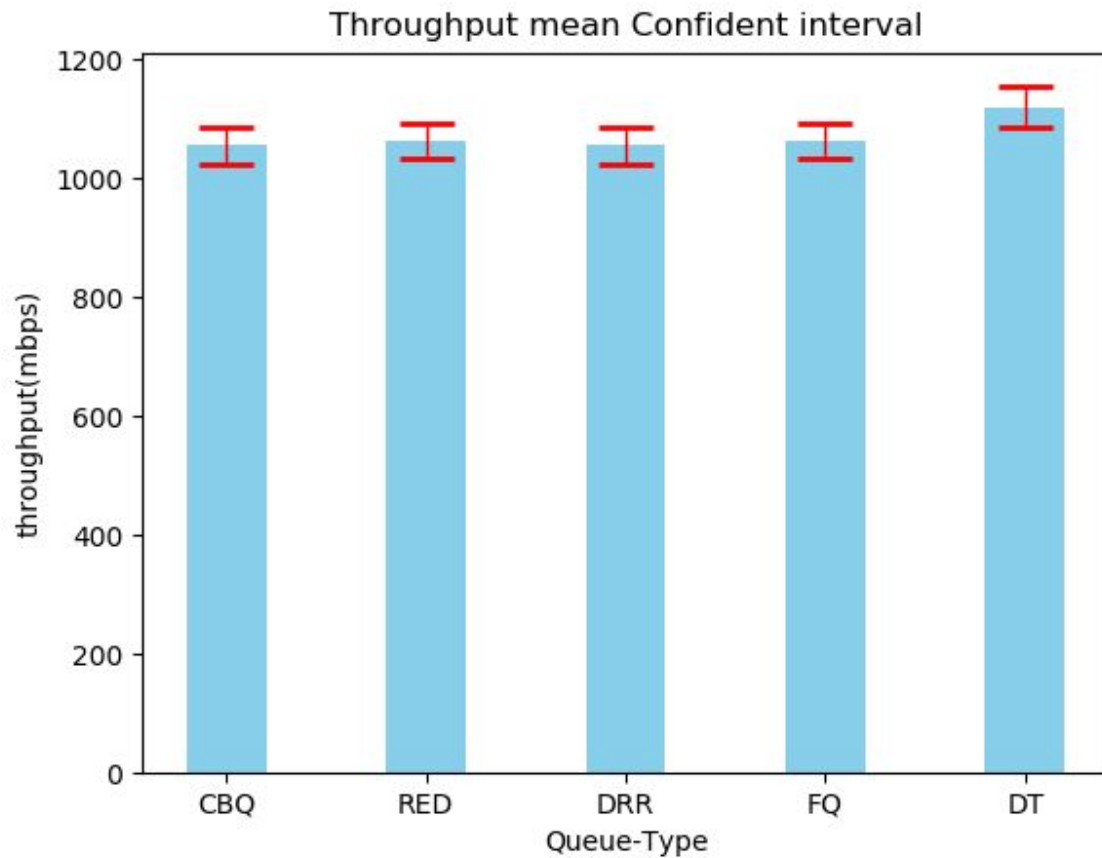
Nous souhaitons tester de manière empirique les performances de différentes gestions de files d'attente. Les graphes ont cette fois tous été générés avec Python et les chiffres (moyennes, écarts-types et intervalles de confiances) ont été calculés avec R. Pour amplifier les écarts, nous avons diminué la capacité des liens utiles à 200 Mbps.

Graphes des débits utiles au cours du temps



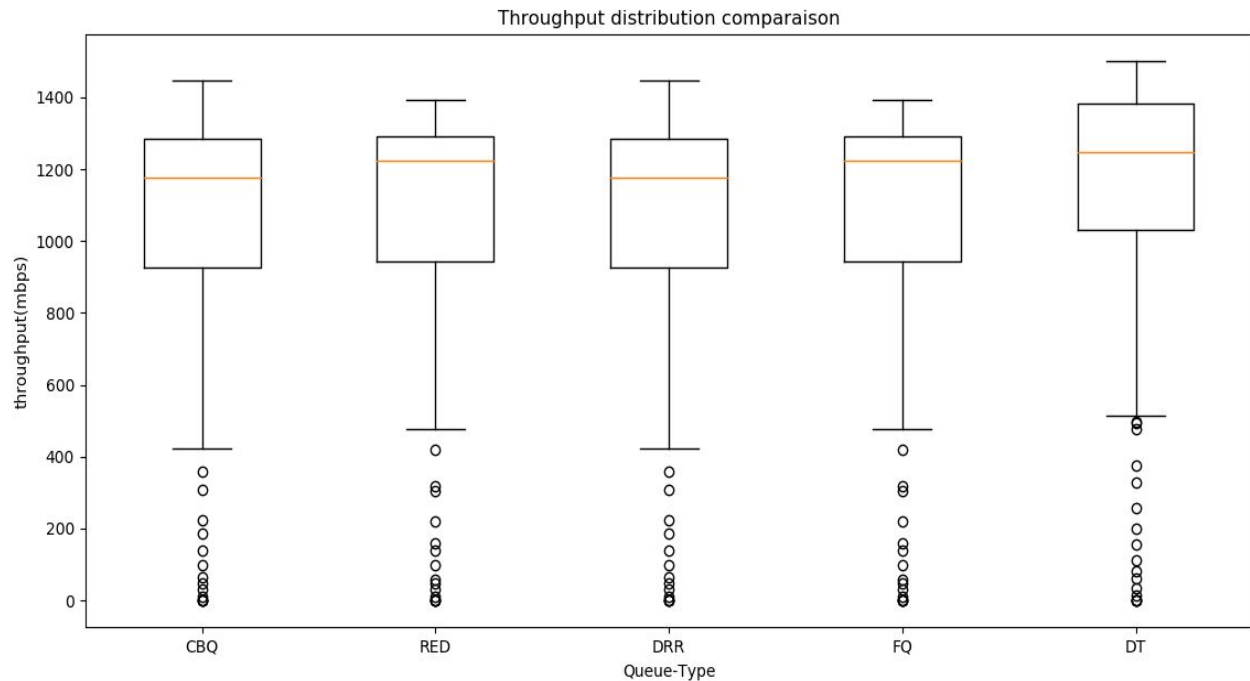
À vue d'œil, difficile d'être certain de quoi que ce soit, tant les écarts sont ténus. Toutefois, la plus simple des files d'attente, DropTail, semble fournir la meilleure performance sur notre réseau.

Graphe des moyennes et de leurs intervalles de confiance



Les moyennes confirment que DropTail obtient les meilleures performances, mais de peu, et même avec des confiances bien trop généreuses de 80 %, les intervalles se chevauchent quasiment tous. Nous fournissons plus bas les chiffres pour des intervalles de confiances à 99 %, pour une analyse plus fine.

Graphe des boîtes à moustaches : médiane, quartile et étendu !



Même remarque que précédemment : Droptail est le vainqueur, mais les écarts sont tenus, non pas uniquement sur les moyennes et médianes, mais aussi sur leurs étendues (et donc probablement aussi sur leurs écarts-types, mais nous vérifierons cela dans la section suivante.)

Comparaison en chiffres entre Fair Queuing et DropTail

Nous avons choisi un intervalle de confiance à 99 %, calculé avec R.

Fair Queuing (FQ)

Moyenne : 1269,4

Écart-type : 62,2

Intervalle de confiance : [1253,9 1285,0]

DropTail (DT)

Moyenne : 1349,5

Écart-type : 94,1

Intervalle de confiance : [1326,0 1373,1]

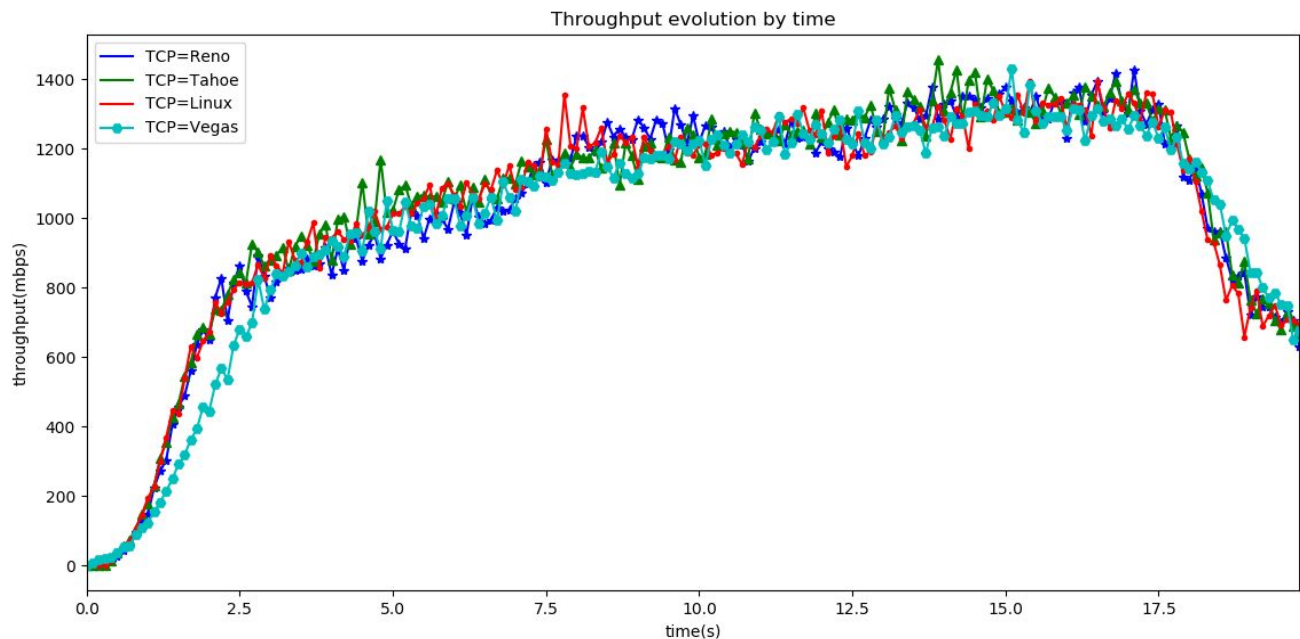
Analyse

Les chiffres ici semblent démontrer que DropTail est meilleur que Fair Queuing puisque leurs intervalles de confiance ne se chevauchent pas.

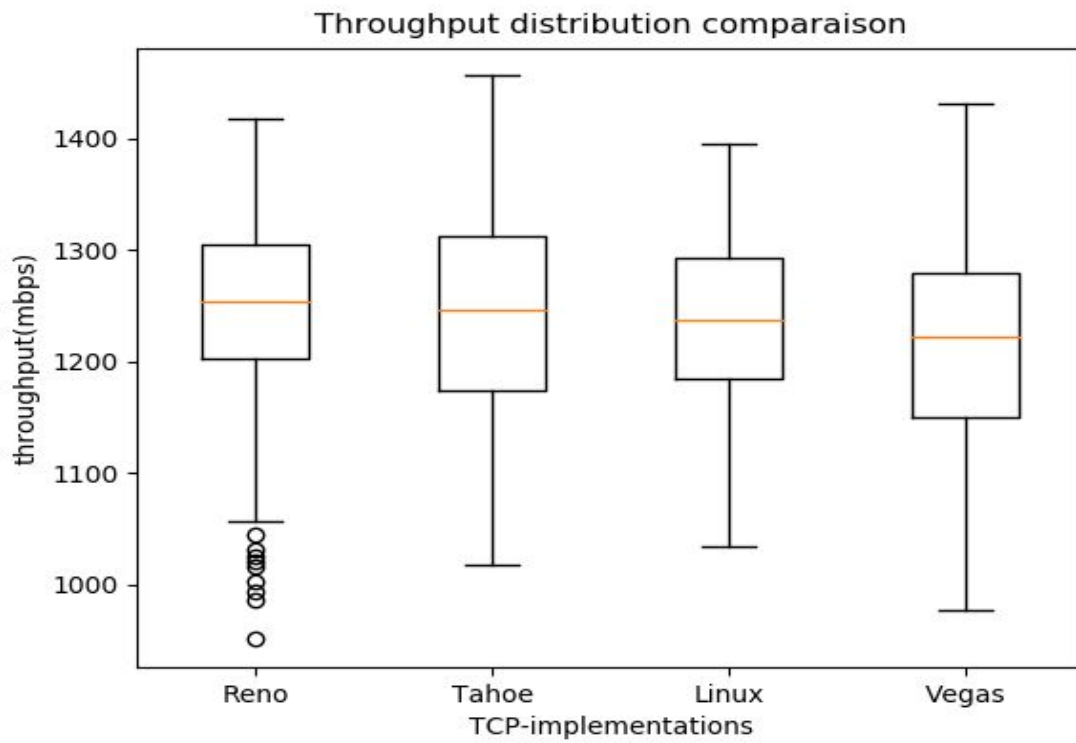
INFLUENCES DES ALGORITHMES TCP

Nous avons testé quatre versions TCP : Tahoe, Reno, Linux et Vegas. Nous nous attendions à avoir des différences marquées entre ces différentes versions, mais à vue d'oeil les écarts sont petits.

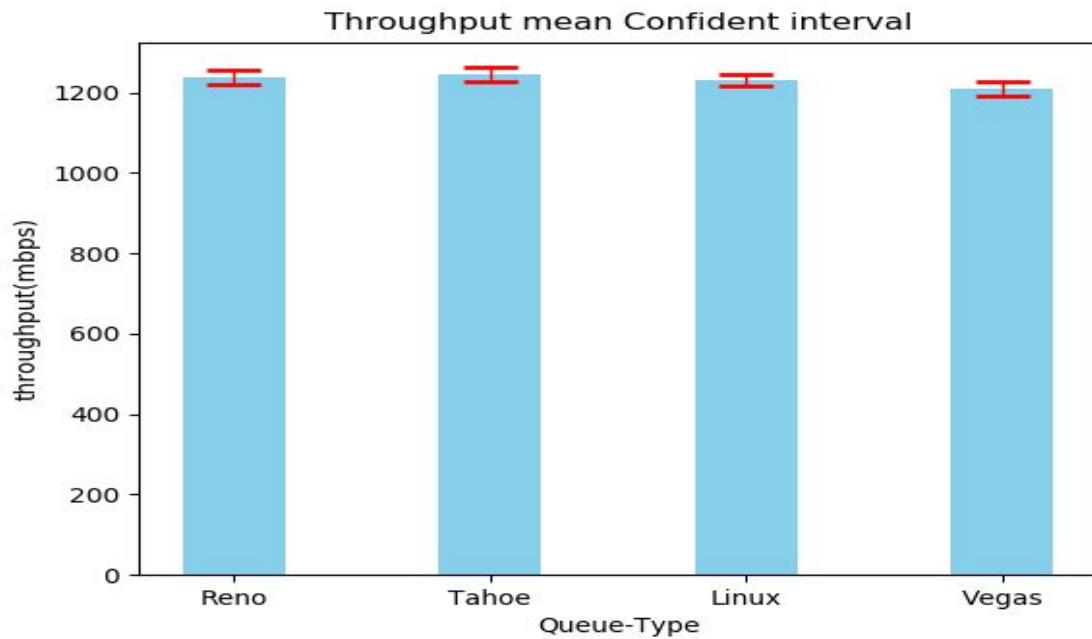
Graphe des débits utiles au cours du temps



Grphe des boîtes à moustaches : médiane, quartile et étendu !



Grphe des moyennes et de leurs intervalles de confiance



Les chiffres

Les résultats sont visuellement assez proches, nous passons donc aux chiffres. Voici donc les intervalles de confiance à 95 % des débits utiles en Mbps :

Reno : [1220,0 1257,0]
Tahoe : [1227,3 1262,5]
Linux : [1217,8 1246,9]
Vegas : [1192,9 1226,1]

La seule chose que l'on puisse déduire de ces chiffres est que, sur notre topologie réseau, TCP Vegas obtient de moins bonnes performances que TCP Tahoe, en moyenne. L'intersection de tous les autres intervalles n'étant pas vide, nous ne pouvons rien déduire de plus sans prendre des risques importants.

CONCLUSION

Les paramètres réseau “software” qui ont le plus impacté le débit utile sur notre topologie sont premièrement la taille des files : les petites tailles font nettement chuter les performances, mais au-delà d'un certain point il est contre-productif de les augmenter davantage. Ensuite, pour les types de files d'attente, les écarts étaient faibles, mais la plus simple, DropTail, est sortie vainqueur. Finalement, pour les versions TCP, il semblerait que sur notre topologie les différences soient proches du néant. La seule chose que nous ayons pu prouver est que TCP Tahoe est plus performant que TCP Vegas sur notre simulation.

CODE

https://github.com/stevlulz/super_tp3_ns2