LAVIE
Pierre-louis
M1 RISE

RAPPORT:
ANALYSEUR DE TRACE

I)Commandes,choix d'implémentations,fonctionnalités

a)Structures et choix d'implémentation

Je détaille ci-dessous les différentes structures utilisées afin d'obtenir les stats demandés ainsi que ma compréhension de ce que sont ces statistiques (uniquement celles non triviales).
Un paquet traité est un paquet soit émis soit retransmis .
Le délai moyen de bout en bout correspond à l'écart entre le départ et la réception d'un paquet.
Le temps d'attente dans une file correspond à l'écart entre l'arrivée d'un paquet dans un nœud (code = 0|1) et son départ d'une file d'attente (code = 2)
Le temps de transmission sur un lien correspond à l'écart entre un code 2 et un code 1 | 3

Soit N : Nombre de routeurs

Evenement.h : une structure simple qui contient l'ensemble des informations utiles d'un événement . Je ne garde pas ici bif et tos puisqu'ils ne sont pas utiles

listeEvenement.h : La "vie" d'un paquet correspond à une suite finie d'événement commençant par un code 0 et terminant par 3||4 . Cette structure est une liste chainée d'événement de même pid qui permet de contenir tous les événements d'un paquet .

ListeFlux.h : Une liste chainée de flux sans doublons , permets de compter le nombre de flux

nœud.h : Une structure permettant de retenir le nombre de paquets reçus/émis/traités/perdus au niveau d'un routeur.

StatsLien.h : Une structure contenant le nombre de paquets reçus/émis/traités/perdus ,le délai moyen de bout en bout , le temps attente moyen et le temps transmission au niveau d'un lien.

MatriceStats.h : Une structure contenant un tableau de N nœud , N * N statsLien et d'autres informations sur les statistiques de la trace

listeListeEvenement : Une liste de listes d'événements .Ces listes sont regroupées car elles sont à la même case du tableau de liste de listes d'événements.

analyseur.h : La structure principale . Elle contient un tableau de liste de liste d'événements dont la taille est donnée en argument (par défaut 1024) , un tableau de liste de flux , une matrice de stats .

ArgAnalyseur.h : Une structure contenant les arguments de l'exécutable.

Explication :
Afin d'obtenir les statistiques sur la trace je cherche à stocker des listes d'événements qui correspondent au vécu de mon paquet . Une fois que cette liste est terminé (c à d qu'on rencontre un code 3 ou 4) je peux effectuer des statistiques sur cette liste d'événements et donc libérer immédiatement la mémoire de cette liste . Cependant je ne peux pas me contenter de faire des simples listes car sinon les parcours sont trop longs . Les listes ne sont pas adaptées si on fait des parcours complets trop réguliers.
Voilà pourquoi je dispose d'un tableau de liste de flux et d'un tableau de liste de liste d'événements . Lorsque je traite un nouvel événement je peux le placer à la case pid/fid % taille tableau . Ensuite il s'agit dans le cas de la liste de liste d'événements de le mettre dans la bonne liste . Je contourne ainsi le problème des listes puisque, à supposer que les fids/pids sont répartis équitablement , les listes de mon tableau ont environ la même taille et j'accède rapidement à la bonne liste.

b)Commandes

Un appel de l'exécutable permet d'obtenir toutes les stats globales du fichier . J'ai choisi d'implémenter également le traçage d'un paquet par pid .
Les différentes options sont :
-f 'nom du ficher'
-s 'taille de la fenêtre de liste de liste'
-p 'pid du paquet à tracer'
-v (Active verbose)

exemples :
./analyse.out -f trace2650.txt
Affiche les stats globales ainsi que la localisation des pertes .

./analyse.out -f trace2650.txt -v
Affiche les stats globales ,la localisation des pertes ,les statistiques de chaque nœud , les statistiques de touts les liens qui ont au moins émis/reçus/perdus un paquet.

./analyse.out -s TAILLE_FENETRE -f 'trace2650.txt'
Affiche les stats globales ainsi que la localisation des pertes en utilisant un tableau de taille TAILLE_FENETRE

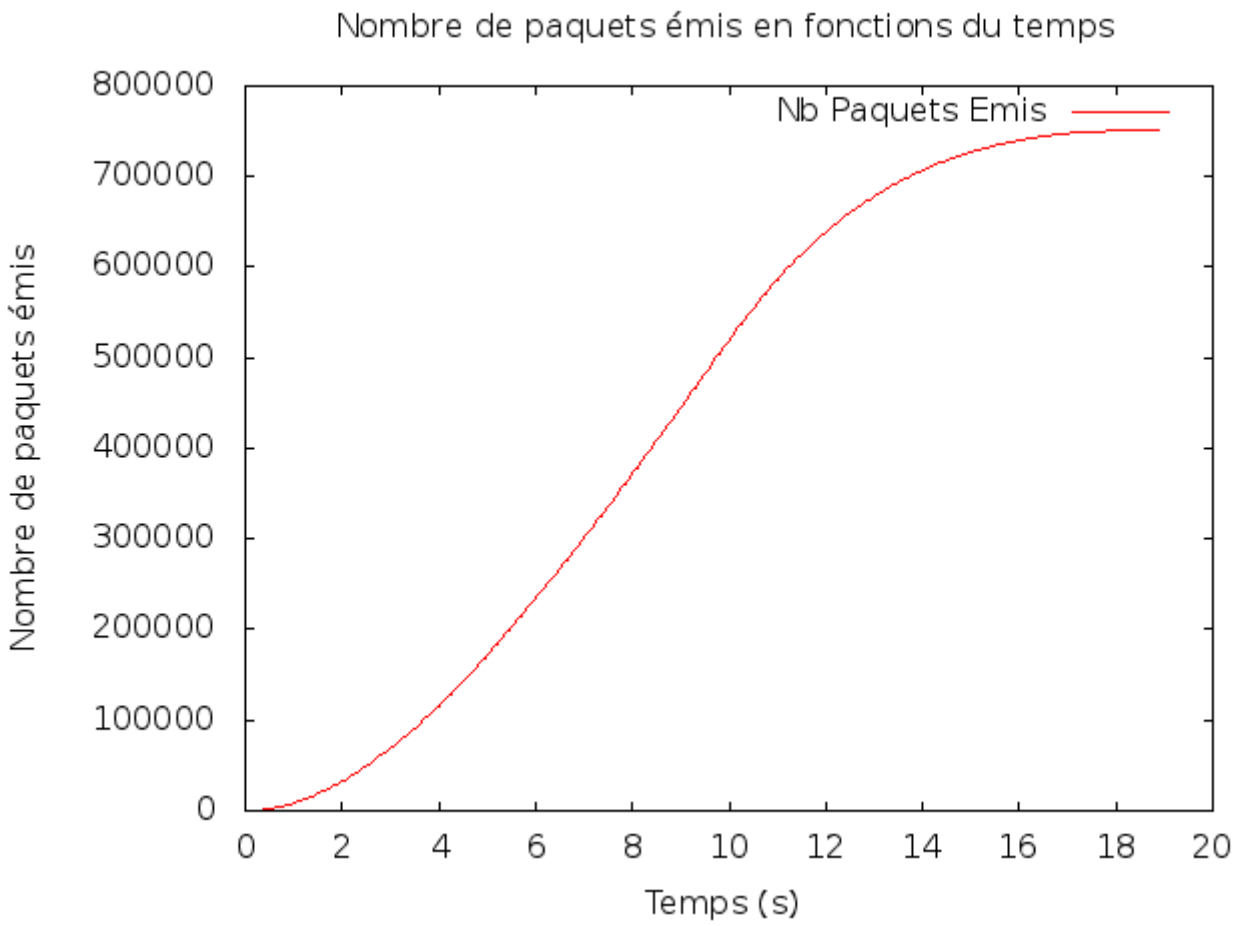
./analyse.out -p PID -f 'trace2650.txt'
Affiche tous les événements du paquet PID

c)Fonctionnalités

Toutes les stats concernant l'ensemble de la simulation sont obtenues sauf une : la taille des listes d'attente .Le fait de choisir d'évaluer une liste d'événements et non pas événement par événement m'empêche de calculer la taille maximum des listes d'attente c'est bien dommage car vu la nature du réseau je pense qu'elles ne sont pas uniformes.
Le traçage d'un paquet est implémenté.

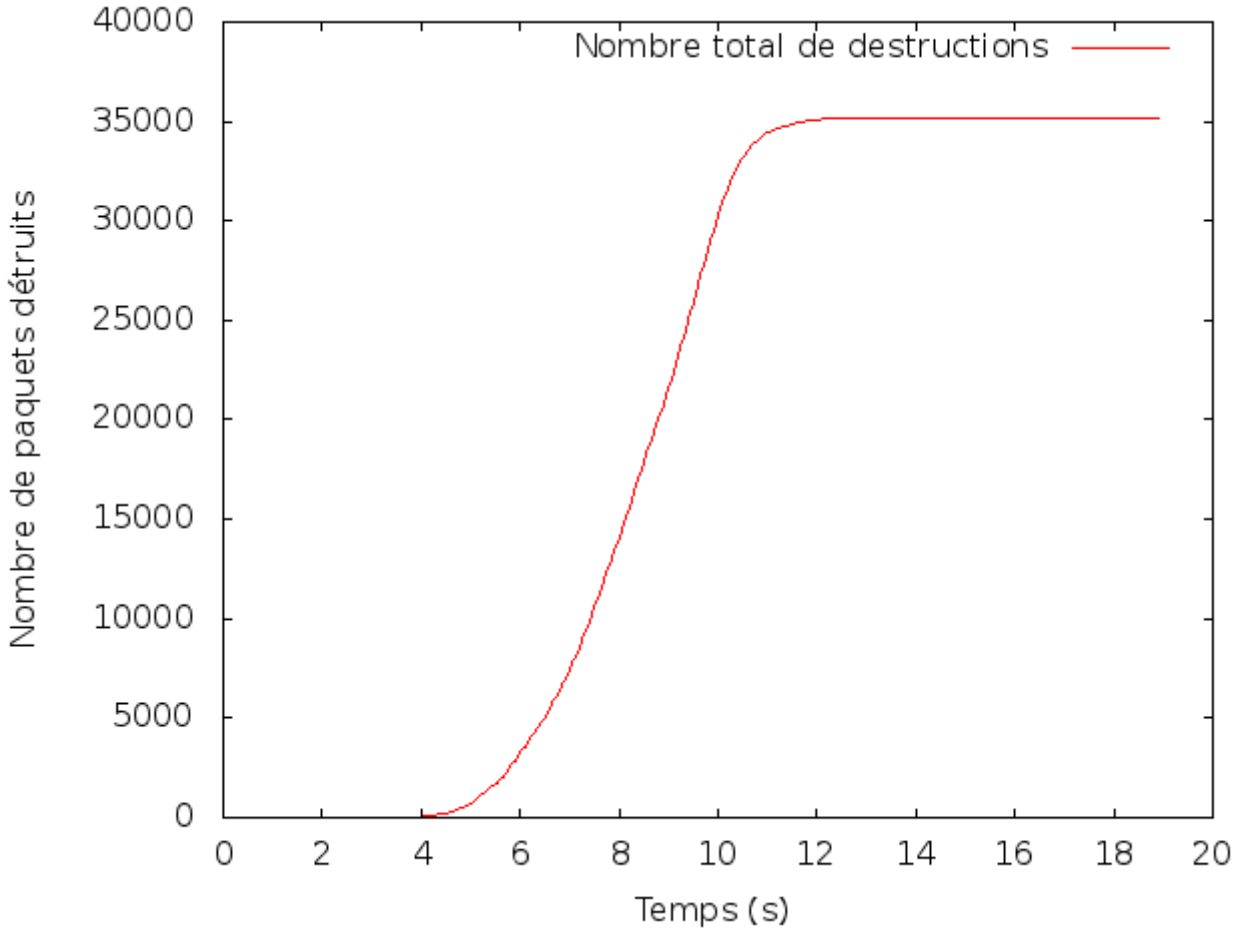
II)Ensemble des résultats obtenus

Pour comprendre le phénomène de congestion du réseau j'ai tout d'abord regardé des stats globales du réseau .



Evolution du nombre de paquets émis en fonction du temps

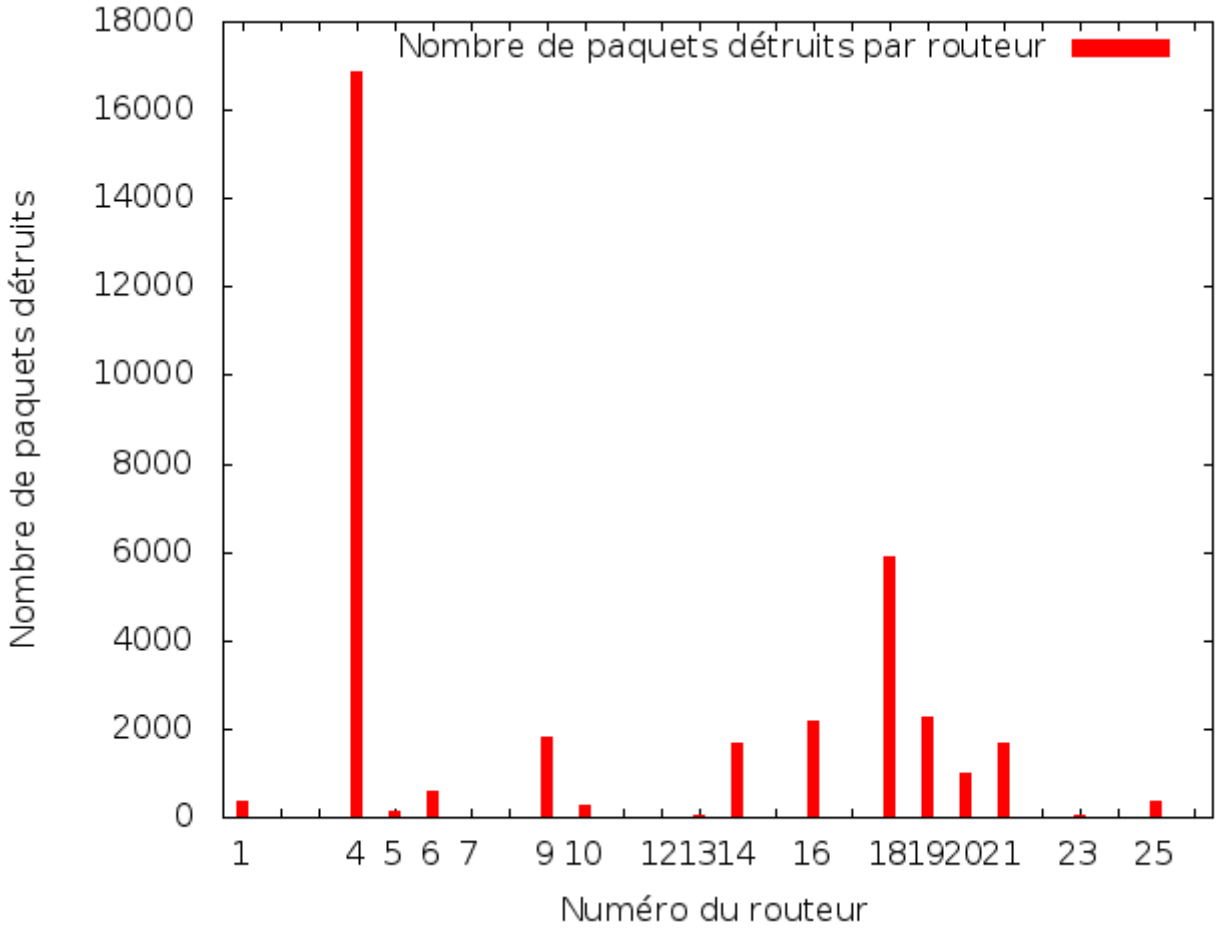
On peut voir sur cette courbe que le nombre de paquets émis est plutôt stable de 2 à 12 secondes .Puis il diminue lentement.



Evolution du nombre de paquets détruits en fonction du temps

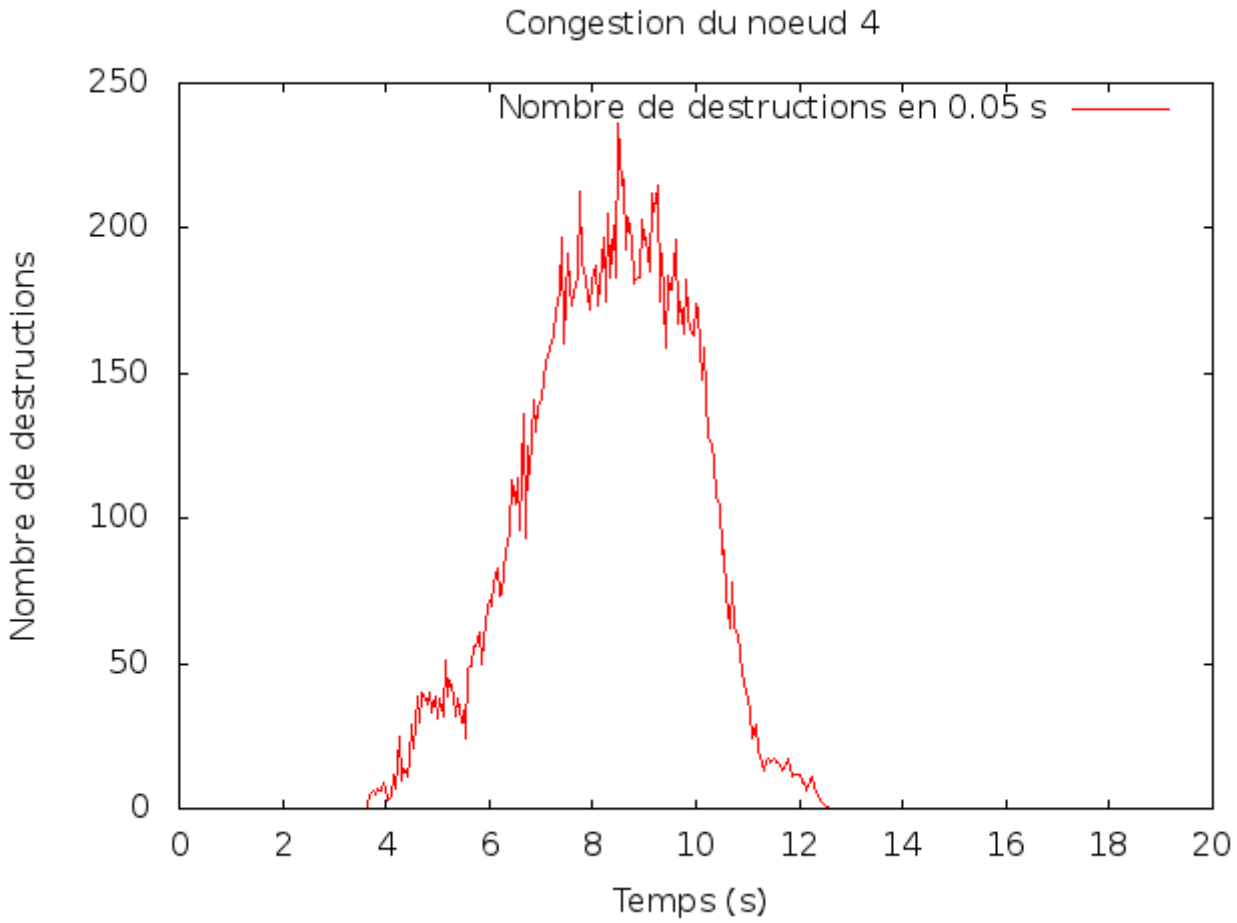
On voit ici que la congestion commence à partir de 4 secondes et dure jusqu'à environ 10 secondes . On peut donc observer un phénomène de congestion sur le réseau cependant ces courbes restent trop vagues car elle signale la congestion mais ne précise pas sa localisation .

En regardant la localisation des pertes on peut constater que la répartition des pertes n'est pas du tout équitable .Le nœud 4 représente 48 % des pertes , le nœud 18 17%des pertes .

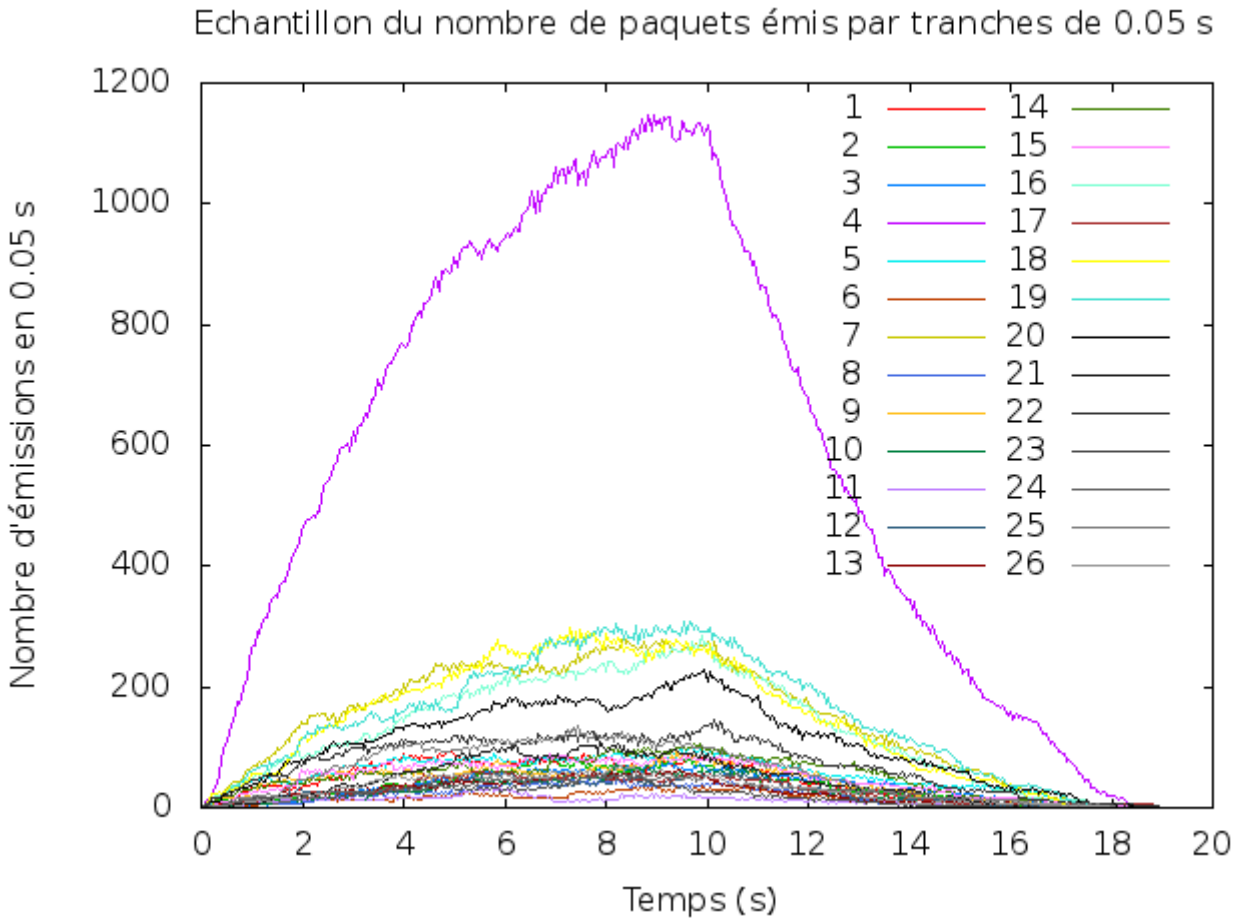


Localisation des pertes sur le réseau

Maintenant que je sais quel nœud est responsable (en majorité) de la congestion je peux faire un échantillonnage de son nombre de paquets perdus en fonction du temps .Je choisis un pas de 0.05 s



On voit donc ici que ce nœud est rapidement saturé . A partir de la seconde 4 il commence à détruire des paquets régulièrement . De 6 à 11 secondes il grimpe à plus de 175 paquets détruits par tranche de 0.05 secondes .
On peut donc se poser la question du rôle de chaque nœud dans le réseau . Est-ce que le nœud 4 est en congestion car il est s'agit d'un nœud carrefour ? Est-il seulement différent des autres nœuds par sa congestion ?



Echantillonage du nombre d'émissions en fonction du temps
par nœud

On peut voir ici que le nœud 4 n'est pas uniquement un nœud central mais il émet également beaucoup plus de paquets que tous les autres nœuds . On remarque également que les nœuds qui ont perdu beaucoup de paquets sont également les nœuds qui émettent le plus de paquets .
Dernière observation les nœuds ne sont indépendants en effet ils suivent tous à motif semblable.

III)Conclusion

Ce qu'on sait :
-il y a une congestion importante située sur le nœud 4 qui joue un rôle important dans le réseau .
-le réseau n'est pas uniforme il existe des nœuds qui émettent beaucoup plus que d'autres .
-il existe une forme de synchronisation des émissions dans le réseau

Ce qu'on aurait dû savoir :
-la taille maximum des files d'attente est-elle uniforme ? Probablement pas je pense qu'elle doit être proportionnelle à l'importance des nœuds

Ce qu'on ne peut pas savoir :
-est-ce que cet événement est représentatif d'un phénomène récurrent sur le réseau?
-Qu'est-ce qui provoque un tel afflux vers le nœud 4 ?