UNIVERSITÉ Grenoble Alpes

Master Informatique — Option Réseau

Qualité de Service

Martin Heusse, Pascal Sicard

1 La qualité de service dans les réseaux de données

Dans bien des cas, le service *best effort* des réseaux IP est insuffisant par rapport à leur utilisation. Un exemple simple est celui de la téléphonie sur IP : pour obtenir un fonctionnement satisfaisant, les flots générés doivent être protégés pour ne pas souffrir de la présence d'autres trafics.

Dans le cas *best effort*, les paquets sont routés à leur réception et éventuellement mis en attente avant leur retransmission, tous dans la même file d'attente. Pour obtenir des traitements différenciés, on utilise plusieurs files d'attente, dans lesquelles on vient chercher les paquets à des taux différents. Le choix de la file d'attente dans laquelle ira un paquet se fait en fonction de ses entêtes ou de son interface d'arrivée.

Il existe plusieurs approches pour décider comment mettre en correspondance les besoins des applications avec le service alloué aux différentes files d'attente. On peut envisager les réservations explicites tout au long du trajet du flot, c'est l'approche *IntServ*. On parle sinon d'approche *DiffServ*, quand c'est uniquement l'entête des paquets qui décide de leur priorité.

Dans le cadre de ce TP, on se limitera à la mise en œuvre locale de techniques de qualité de service, qui sont les mécanismes à la base des architectures de qualité de service au niveau d'un réseau complet.

RAPPEL : Dans le cas d'utilisation de routeurs *cisco 800* plusieurs points à respecter : Ce type de routeur intègre :

- Un mini commutateur 100 mégabits/s (prises **FE0** à **FE7**). L'interface Ethernet sur le routeur connectée au mini switch s'appelle **Vlan 1**. Les débits des prises du switch sont par défaut à 100 Mégabits/s.
- Deux interfaces Ethernet (**FE8 (Fast Ethernet)** à 100 mégabits/s et **GE0 (GigaEthernet)** à 1 gigabits/s.

ATTENTION ne pas utiliser l'interface *vlan 1* sur le routeur 1 pour la liaison avec le routeur 2 (le **fairqueuing** ne fonctionne pas sur cette interface).

2 Mise en place du réseau

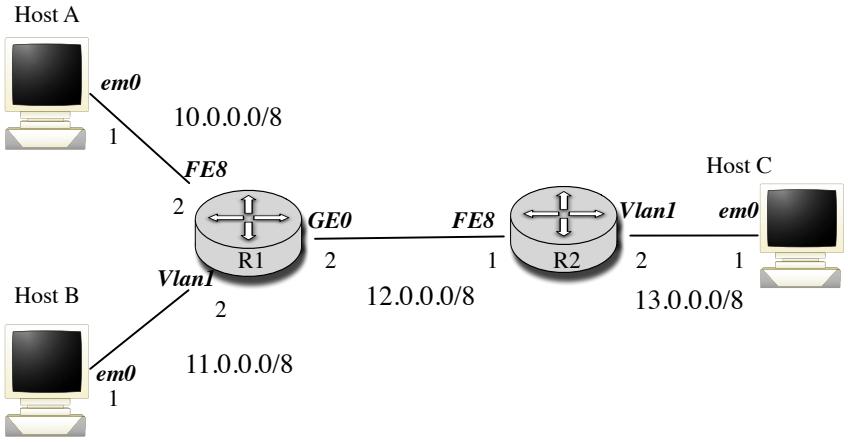

FIGURE 1 – Le réseau de travail

On considère le réseau apparaissant en figure 1.
Un plan d'adressage est donné pour vous faciliter la tâche. Il suppose que vous ayez 2 routeurs 800. Il permet d'avoir sur chacun des réseaux un débits de 100 mégabits/s sans avoir à régler ces débits.
R1 et R2 sont les deux routeurs Cisco, le nom de l'interface et la valeur du dernier octet de son adresse sont donnés (par exemple l'adresse de l'interface vlan1 du routeur R1 est 11.0.0.2/8).

Si vous réalisez un autre montage, il faut se débrouiller pour avoir le même débit sur chaque réseau. Au besoin, cela est facilement réglable, soit sur les routeurs par la commande `speed` sous `configure interface XXX`, soit sur les machines par la commande `ifconfig nominterface media 100baseTX`. **Attention** à faire avant de brancher le câble.

- Installez ce réseau.
- Configurez les interfaces des machines et des routeurs.
- **Vérifier les débits de chaque réseau par `ifconfig` ou `show interfaces` sur les routeurs**.
- Vérifier les connexions directes par des ping.
- Remplissez les tables de routage à l'aide de route par défaut.
- Vérifiez que toutes les machines peuvent communiquer entre elles.

Faisable en 15 minutes quand on est débrouillé, en 2 heures si on s'y prend mal.

3 Premières manipulations

Rappel : Par défaut dans un routeur, il n'y a qu'une file d'attente de sortie par interface ethernet, avec un service de type *fifo*. Il y a également un file d'attente en entrée, mais qui ne sert qu'en cas de surcharge du sous-système d'aiguillage : les paquets sont commutés immédiatement et **l'attente se fait toujours devant l'interface de sortie** (voir figures 2).

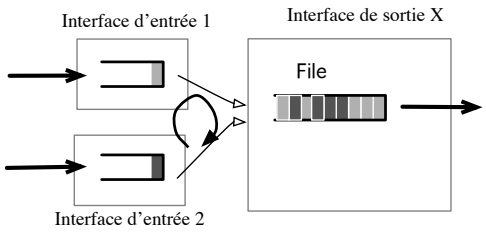

FIGURE 2 – Traitement des files d'attente

Dans les expérimentations suivantes, il est conseillé d'observer l'évolution des débits (en réception dans le cas d'UDP) à l'aide de graphes issus de ***Wireshark*** (menu STATISTICS/ IO GRAPH). On peut afficher plusieurs courbes en utilisant le bon filtre.
Exemples de filtre :

- `icmp || dns`
- `ip.addr == 192.0.0.2`
- `tcp.port <= 90`
- `ip.src != 192.0.0.1 or ip.dst == 193.0.0.1`

Si on veut des débits, il faut passer en `octets/ticks` au lieu de `paquets/ticks`.

3.1 Flux UDP contre TCP

On considère la concurrence entre un trafic UDP de A vers C avec un trafic TCP de B vers C. (Générés par exemple par `udpmt` et `tcpmt`.)

Observez les débits donnés par `udpmt` et `tcpmt` en réception et en émission, ainsi que leur variation ? Pourquoi UDP prend t il toute la bande passante ? (Bien détailler ce pourquoi...) Quel mécanisme de TCP est mise en jeu ici ? Que se passe t il dans le routeur ? Les débits sont ils stables ?

3.2 Flux UDP contre UDP

On observe maintenant la concurrence entre 2 trafics UDP. Utilisez `udpmt` / `udptarget`. Ces outils permettent de suivre l'évolution de la bande passante dévolue à chacun des

flots à une échelle de temps relativement fine. Attention il faut lancer deux `udptarget` sur deux ports différents avant les `udpmt`.

Qu'observe-t-on ? Vous seriez-vous attendu à cela ? Les débits sont ils les mêmes en réception ? Sont ils stables ?

Pouvez-vous expliquer le phénomène observé ? que se passe t il au niveau du routeur ?

3.3 Flux TCP contre TCP

On observe maintenant la concurrence entre 2 trafics TCP. Même remarque, Attention il faut lancer deux `tcptarget` sur deux ports différents avant les `tcpmt`.

Qu'observe t -on au niveau des débits ? Quel mécanisme de TCP est mise en jeu ici ? Observez les courbes d'évolution des flux depuis A et B donnés par Wireshark pour vous aider à comprendre.

Remarque : `sh interface ethernet x/x` vous permet d'observer la taille courante et maximale des files d'attente d'une interface. Les tailles maximales (en entrée et en sortie) peuvent être changée par la commande `hold-queue` des interfaces [1]. Vous pouvez aussi capturer les paquets sur C pour observer l'évolution des flux depuis A et B.
Pour observer la charge du routeur, il existe la commande `show processes`, qui est l'équivalent du `top` de unix.

3.4 *Fair Queuing*

Il est très simple (en termes de configuration) de corriger les problèmes relevés ci-dessus. Les routeurs Cisco permettent d'utiliser pour une interface de sortie des files d'attente différentes **en fonction des interfaces d'entrée** d'où sont issus chaque paquet (voir la figure 3). Chaque file d'attente est ensuite servie de manière équitable par rapport aux autres.
On active cette politique par la commande `fair-queue` au niveau configuration d'interface (sur l'interface de sortie du trafic, c'est là que les paquets sont mis en attente).

1. Les compteurs de `show interface`, peuvent être remis à zéro par `clear counters` sous `enable` (ou au niveau interface).

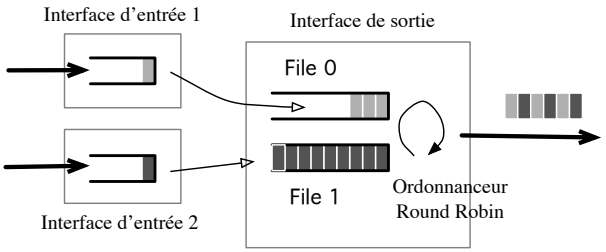

FIGURE 3 – Fair-Queuing sur files d'attente définies à partir des interfaces d'entrées

Reprenez les expériences précédentes. Comparez les résultats. Expliquez les différences.

L'équité s'entend-elle en terme de nombre de paquets transmis ou en nombre d'octets transmis ? Comment vérifier cela ?

4 Limitation de débits sur des flux de paquets particuliers

Dans bien des cas, le problème de la protection d'un flot de paquet particulier (identifié par le protocole de transport, les adresses IP ou les ports utilisés) se pose. Nous allons protéger le trafic venant de l'hôte A (TCP par exemple), en limitant le débit de celui provenant de l'hôte B (UDP par exemple).

Supprimer tout d'abord le `fair-queuing`.

Indications :

1. Identification d'un flux Pour arriver à prendre en charge de façon particulière un flot donné, il faut d'abord l'identifier. On définit une `access-list` correspondant à ce flot (utiliser une liste de numéro supérieur à 100, c'est alors une liste étendue qui permet de préciser l'adresse source.)
   Exemples d'acces list :

   - Filtre les paquets TCP à destination du réseau 129.88.38.0/24, vers le port 80 ; quel que soit l'émetteur :
     ```
     Routeur (config)# access-list 101 permit tcp 0.0.0.0
              255.255.255.255 129.88.38.0 0.0.0.255 eq 80
     ```

     Remarque : Ici les bits désignés par les masques sont les bits sans importance (à l'opposé de ce qui se passe pour les masques de sous-réseau) ; 0.0.0.255 est équivalent à un netmask 255.255.255.0 ; 0.0.0.0 255.255.255.255 désigne donc effectivement une adresse quelconque. On peut aussi utiliser le mot clé `any` à la

place de `0.0.0.0 255.255.255.255`.
Filtre équivalent :

```
Routeur (config)# access-list 101 permit tcp any
                    129.88.38.0 0.0.0.255 eq 80
```

- Filtre les paquets UDP :
  `Routeur (config)# access-list 101 permit udp any any`

- Filtre les paquets TCP :
  `Routeur (config)# access-list 102 permit tcp any any`

2. Définition d'une classe de paquet permettant ensuite d'appliquer un traitement de QoS (appelé class-map)

```
Routeur (config)# class-map match-all prio-sur-acces-list101
Routeur (config-cmap)# match access-group 101
```

   A savoir, il existe d'autres types de filtres (voir les paramètres de la commande `match`). Par exemple pour tous les paquets arrivant sur une interface :
   `match input-interface nom-interface`

   On peut vérifier la définition des classes par : `show class-map`

3. Limitation de débit
   On peut maintenant définir un traitement particulier à ces classes de paquets.
   Par exemple une limite du débit :

```
Routeur (config)# policy-map limitation-debit
Routeur (config-pmap)# class prio-sur-acces-list101
Routeur (config-pmap-c)# shape average 16000
exit
Routeur (config-pmap)# class prio-sur-acces-list102
Routeur (config-pmap-c)# shape average 256000
```

   Ici le débit des paquets de l'acces list 101 est limité à 16 koctets/s, ceux de l'acces list 102 à 256 koctets/s.
   On peut vérifier la définition des traitements : `show policy-map`

4. Application des traitements définis précédemment sur une interface de sortie :

```
Routeur (config)#  interface GigabitEthernet 0
Routeur (config-i)# service-policy output limitation-debit
```

Protéger un flux TCP en limitant le débit d'un trafic UDP. Donnez le résultat de vos expérimentations (évolutions des débits observés).

Si maintenant c'est en flux TCP qui est limité (sans concurrence), comment TCP adapte-t-il son débit d'émission au goulot d'étranglement ?

Quelle est la limite de cette méthode ?

5 Traitement de QoS par pondération

On peut au lieu de faire une limitation de débit "dure", pondérer le traitement des différentes classes de paquets, par exemple :

```
Routeur (config)# policy-map ponderation-debit
Routeur (config-pmap)# class prio-sur-acces-list101
Routeur (config-pmap-c)# priority percent 10
Routeur (config-pmap)# class prio-sur-acces-list102
Routeur (config-pmap-c)# priority percent 89
```

Puis comme dans la manip précédente appliquer ce traitement à une interface de sortie :

```
Routeur (config)#  interface GigaBitEthernet 0
Routeur (config-i)# service-policy output ponderation-debit
```

Mettre en évidence cette pondération au moyen de deux flux TCP vers deux ports différents. [Ne pas oublier d'effacer les commandes de la manipulation précédente]

Même chose avec deux flux UDP vers deux ports différents. [Ne pas oublier d'effacer les commandes de la manipulation précédente]

Donnez le résultat de vos expérimentations (évolutions des débits observés).

Est ce que la bande passante totale est allouée lorsqu'un seul flux est présent ?

Essayez avec des flux comportant des paquets de tailles différentes. Est ce que le traitement des files par le routeur change ?

6 Marquage des paquets

Il y a des situations où les paquets doivent être protégés plus loin dans le réseau. C'est le cas si C émet deux trafics vers A, dont 1 devra être protégés si la bande passante disponible en aval de R1 est insuffisante—du fait d'un trafic entre B et A par exemple : voir figure 4.

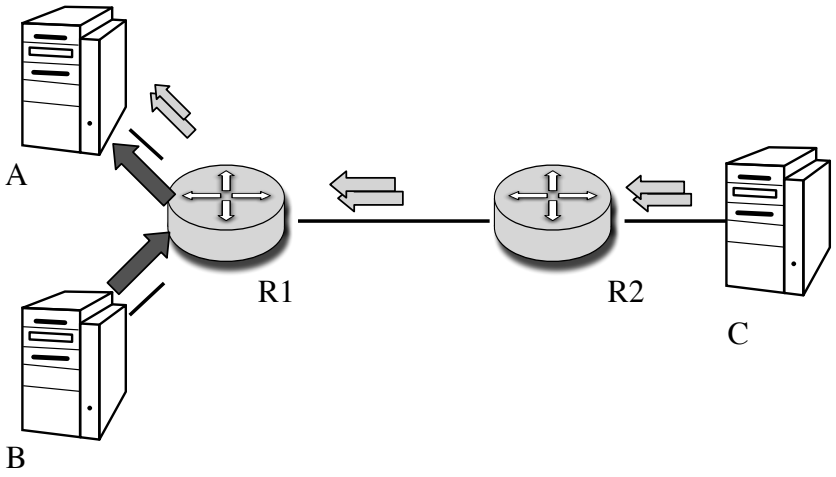

FIGURE 4 –

6.1 Observation en cas de congestion sans différentiation des flux

Hop. Que constatez-vous sur le partage entre 3 flots UDP : deux de C vers A, un de B vers A ?

Que se passe-t-il si vous utilisez le *fair-queueing* sur FastEthernet 8 de R1 ? Expliquez les débits obtenus en réception.

Refaites les mesures avec trois flux TCP ? Expliquez les débits obtenus. Pourquoi le partage est cette fois équitable entre les trois flux ?

6.2 Marquage des paquets dans un routeur

Nous allons utiliser le champs DSCP (*Differentiated Services Code Point*) des paquets IP pour marquer **sur R2** ceux qui sont à protéger. Le champ DSCP (*Differentiated Services Code Point*) était auparavant appelé ToS (*Type Of Service*).

- Il faut définir une `access-list` qui correspond aux paquets en question, identifiés par exemple par le port destination.
  ```
  Routeur (config)# access-list 120 permit tcp any any eq 13001
  ```

- Ensuite, il vous faut créer une classe qui va être utilisée pour changer le champs DSCP des paquets :
  ```
  Routeur (config)# class-map match-all prio-sur-acces-list120
  Routeur (config-cmap)# match access-group 120
  ```

- Puis on définit le marquage des paquets de cette classe :
  ```
  Routeur (config)# policy-map marquageDSCP7
  Routeur (config-pmap)# class prio-sur-acces-list120
  ```

```
Routeur (config-pmap-c)# set ip dscp cs7
```

`cs7` met 7 dans le champ DSCP de l'entête IP des paquets.
Remarque : On peut spécifier directement la valeur du champ ou un traitement qualitatif qui est alors traduit en une valeur (voir paramètres du `set ip`).

- On n'oublie pas d'associer ce marquage à une interface de sortie :
```
Routeur (config)#  interface FastEthernet 8
Routeur (config-i)# service-policy output marquageDSCP7
```

Procédez au marquage sur R2 et capturez ces paquets marqués.

6.3 Utilisation du marquage pour privilégier les paquets suivants la classe de service

Pour le moment, le marquage n'a aucune influence sur le traitement au routeur R1. Pour modifier le comportement de R1 suivant le marquage des paquets, on peut créer une classe de paquet liées au marquage effectué sur R2.

- Créez une `access-list` pour identifier les paquets marqués :
Par exemple : `access-list 130 permit ip any any dscp cs7`

- Créez une classe associée à cet accès list :
```
Routeur (config)# class-map match-all prio-sur-acces-list130
Routeur (config-cmap)# match access-group 130
```

- Pondérer cette classe :
```
Routeur (config)# policy-map ponderation-classe-serv
Routeur (config-pmap)# class prio-sur-acces-list130
Routeur (config-pmap-c)# priority percent 90
```

- Associer cette politique de QoS à l'interface de sortie FE8 :
```
Routeur (config)#  interface FastEthernet 8
Routeur (config-i)# service-policy output ponderation-classe-serv
```

Vérifiez que lors d'une congestion, les paquets marqués sont privilégiés ! La congestion peut être provoqué par un flux UDP de A vers B.

Essayez à l'aide d'une limitation de débit au lieu de la pondération. Donnez les résultats de vos observations (débits obtenus dans chaque cas).

Dernière remarque : ces manipulations n'abordent qu'une partie des fonctionnalités offertes par les routeurs dans le domaine de la qualité de service.