

Examen de la valeur

INGENIERIE DES RESEAUX D'ENTREPRISE

Durée 2 heures

Seconde session 2002

Le 16 septembre 18h15 à 20h15

DOCUMENTS AUTORISES : TOUS

Une réponse non argumentée, même juste ne pourra donner lieu au maximum des points.

CORRIGE

1. Multicast IP (6 points)

1.1. Les adresses Multicast de groupe Ethernet peuvent se déduire des adresses Multicast IP. Il a été vu en cours comment on faisait l'extraction des 23 derniers bits de l'adresse IP pour former cette adresse Ethernet. (4 points)

1.1.a. Soit l'adresse IP Multicast privée 239.0.0.5, ou encore EF.0.0.5 en hexadécimal ou encore 11101111.00000000.00000000.00000101 en binaire. (2 points). Déterminez l'adresse Ethernet Multicast de groupe correspondante.

Réponse :

On extrait les 23 bits de fin de l'adresse IP Multicast soit "0000000.00000000.00000101", on les colle à 01:00:5E en hexadécimal suivi d'un bit à 0, ce qui donne l'adresse de Ethernet Multicast:

01:00:5E:00:00:05

1.1.b. Plusieurs adresses multicast IP peuvent produire la même adresse multicast Ethernet, dans l'exemple ci-dessus quel est le problème posé. (2 points)

Réponse :

Dans l'exemple, les adresses multicast IP (privées) qui sont de la forme 11101111.0xxxxxxx.xxxxxxxx.xxxxxxxx vont fournir la même adresse multicast Ethernet que les adresses multicast IP (privées elles aussi) 11101111.1xxxxxxx.xxxxxxxx.xxxxxxxx. Un administrateur ne pourra disposer que de la moitié de l'espace d'adresses multicast IP privées pour éviter des collisions d'adresses multicast Ethernet.

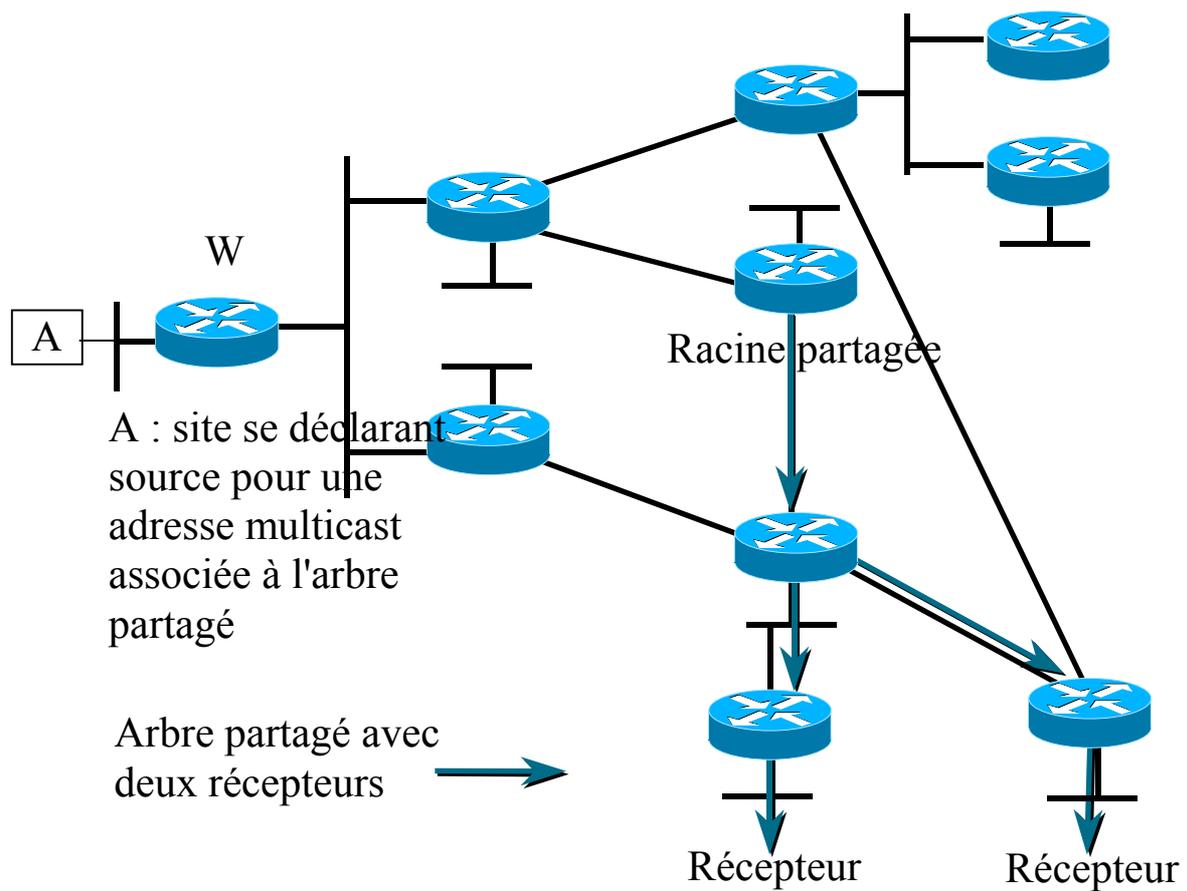
1.2 On s'intéresse au routage multicast.

1.2.a. Deux modes de routage multicast existent: le mode dense (DVRMP, MOSPF, PIM-DM), et le mode éparpillé (PIM-SM). Quelle est la principale différence entre ces deux types de techniques de routage multicast. (1 point).

Réponse :

Le mode dense correspond à une utilisation où les différents membres d'une diffusion multicast sont relativement regroupés. Dans ce cas, chaque source a son propre arbre de routage. Dans le cas du mode éparpillé, plusieurs diffusions multicast partagent le même arbre multicast.

1.2.b. Si on utilise PIM-SM, et que le routage courant donne l'arbre multicast partagé donné page suivante. Est-ce que, du point de vue du trafic multicast IP, le routeur W adjacent à A devient la nouvelle racine de l'arbre partagé. (1 point)



Réponse :

La racine de l'arbre partagé ne change pas. Il faudrait que l'algorithme de routage détecte que du trafic emprunte un autre chemin et que la racine est devenue inutile pour effectuer un élagage.

2. QoS (6 points)

On s'intéresse à une application de suivi de production de Gaz. Les puits sont géographiquement répartis sur un territoire de 1000 km². Les puits produisent de petites quantités de gaz, et font donc l'objet d'une surveillance automatisée. Le suivi de production par un opérateur se fait en un point déporté.

Les équipements informatiques de suivi pour chaque puit sont reliés à un réseau public haut débit à 2Mb/s via un réseau satellite offrant une liaison à 56kb/s. La configuration du routeur d'interconnexion entre les deux réseaux nous intéresse. Le MTU sur le chemin des données est de 200 octets.

Quelques éléments sont donnés sur l'application elle-même, en particulier, les équipements informatiques qui sont soumis à différents types de flux :

- Flux de données de production, 80 octets par minute avec un maximum de 80 octets par 20 secondes en cas d'anomalie pendant 5 minutes. Le délai de transfert requis est de 1mn pour chaque segment.
- Alarmes, c'est un flux exceptionnel de 100 octets par seconde mais qui est prioritaire sur tous les autres quand il se produit. On peut avoir des rafales de 160 octets par seconde pendant 5s. Le délai de transfert requis est de 10s pour chaque segment.

Les débits indiqués correspondent à des débits applicatifs.

2.1. On vous demande de spécifier les paramètres pour une réservation de QoS. Fournissez les valeurs des attributs suivants et expliquez très brièvement vos calculs (4 points):

TokenRate
TokenBucketSize
PeakBandwidth
Latency
DelayVariation
ServiceType
MaxSduSize
MinimumPolicedSize

Réponse :

Sachant que tous les flux traversent l'Internet, il faudra qu'ils participent au contrôle de congestion global. Ils sont donc tous véhiculés par TCP à priori. On aurait pu choisir UDP pour les données de suivi du processus d'extraction de gaz.

Compte tenu de nos hypothèses, pour obtenir la taille d'un datagramme IP, il faut ajouter 20 octets d'entête IP, et 20 octets d'entête TCP aux octets d'un segment de données applicatives.

Rappelons que :

- Le TokenRate doit être exprimé en octets de datagramme IP par seconde.
 - Données du procédé : 120 octets/mn (80+20+20) soit 2 octets/s
 - Alarmes : 140 octets/s (100+20+20 octets)
- Le TokenBucketSize doit être exprimé en octets de datagramme IP. Il est difficile de le dimensionner exactement. Compte tenu du sens des données, on va du réseau satellite vers le réseau haut débit, donc à priori, le routeur ne devrait pas avoir à stocker des datagrammes. Il faut au moins prendre un datagramme. Sachant que dans les deux cas, il y a des rafales, on peut avoir une vision conservatrice et chercher à encaisser la totalité d'une rafale car elles sont relativement peu volumineuses. C'est ce choix que nous faisons.
 - Pour les alarmes, la rafale dure 5s, on transmet 200 octets pour chaque datagramme, le seau doit pouvoir contenir 1000 octets
 - Pour les données procédé, la rafale dure 5mn, on transmet 120 octets à chaque datagramme toutes les 20s, le seau doit contenir $5 \times 3 \times 120$ octets, soit 1800 octets.
- La latence doit être exprimée en micro-secondes. Les latences sont clairement spécifiées dans l'exercice.
- Le PeakBandwidth doit être exprimé en octets de datagramme IP par seconde. Les rafales concernent tous les flux. On applique le même type de raisonnement que pour le TokenRate avec les débits max spécifiés.
- Le DelayVariation doit être exprimé en micro-secondes. Pour aucun trafic, il n'est exprimé de contraintes de gigue. On peut avoir deux approches : ne pas admettre de gigue ou au contraire être très permissif. Ne pas admettre de gigue est très difficile car le protocole utilisé est TCP, et qu'on utilise en plus une liaison satellite qui est sujette aux erreurs de transmission, donc aux retransmissions génératrices de gigue. Une erreur de transmission est très pénalisante, et oblige 2 délais aller et 1 délai retour (pourvu que la retransmission ne soit pas elle-même victime d'une erreur). Donc en dehors de spécification plus précise, on va considérer que DelayVariation prend $2 \times$ la latence fournie dans le contexte applicatif.
- Le MaxSduSize doit être exprimé en octets de datagramme IP.
- Le MinimumPolicedSize doit être exprimé en octets de datagramme IP.

On obtient alors le tableau suivant :

Types de Flux	alarmes	données procédé
TokenRate	140 octets/s	2 octets/s
TokenBucketSize	1000 octets	1800 octets
PeakBandwidth	200 octets/s	30 octets/s
Latency	10000000µs (10s)	60000000µs (1mn)
DelayVariation	20000000µs (20s)	120000000µs (120s ou 2mn)
ServiceType	GARANTEED	CONTROLLED LOAD
MaxSduSize	200 octets	120 octets
MinimumPolicedSize	140 octets	120 octets

2.2. On demande de donner la table de gestion des trafics traversant le routeur d'interconnexion.

La station de supervision a l'adresse IP 10.0.0.1, l'application de supervision utilise le port 1250 pour le suivi du procédé et 1251 pour la réception des alarmes. (2 points)

On rappelle la suite d'informations contenue dans une entrée de table que vous devez renseigner :

IP Source	Port Source	protocol	IP Destination	Port Destinat	Classe/Priorité	PHB	Ovrflow PHB
-----------	-------------	----------	----------------	---------------	-----------------	-----	-------------

Réponse :

IP Source	Port Source	protocole	IP Destination	Port Destinat	Classe/Priorité	PHB	Ovrflow PHB
*	*	TCP	10.0.0.1	1250	2	AF	BE
*	*	TCP	10.0.0.1	1251	1	EF	AF

3. TCP, Evolutions et Approches alternatives (9 points)

3.1. En quoi TCP n'est pas adapté au transfert de données qui ont des contraintes temps réel (contrainte de gigue constante, contrainte de latence bornée par exemple. (1 point)

Réponse:

Essentiellement :

- la gestion des erreurs de transmission introduit délai variable et donc de la gigue, le temps de transmission maximum n'est pas prédictible
- la gestion de la congestion : "slow start" puis "congestion avoidance" introduisent un débit variable donc de la gigue
- la gestion du contrôle de flux "sliding window" ralentit l'émetteur en fonction des capacités de réception (vitesse à consommer des tampons mémoire contenant des données issues du réseau) du récepteur et introduit de la gigue.
- on peut aussi indiquer que TCP ne supporte pas le multicast, mais ceci n'est pas une contrainte temporelle

3.2.

3.2.1. Rappelez quels événements sont utilisés par TCP pour décider que le réseau est en congestion ? (1 point)

Réponse :

Deux événements sont utilisés :

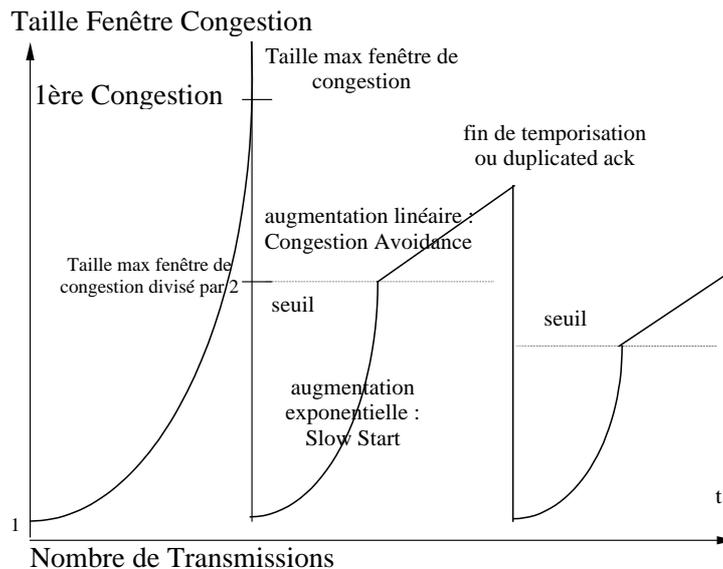
Une fin de temporisation sur un segment émis.

Trois acquittement dupliqués pour le même segment TCP (le récepteur indique qu'un segment lui manque dès qu'il a un trou dans sa numérotation des octets reçus).

3.2.2. Rappelez brièvement les principes des mécanismes baptisés "slow-start", et "congestion avoidance" du protocole TCP. Eventuellement, aidez-vous d'un dessin et indiquez sur ce dessin la partie correspondant au "slow-start" et la partie correspondant à "congestion avoidance". (1 point)

Réponse:

Le mécanisme de Slow Start correspond à la phase de relance de l'émission par l'émetteur qui vient de rencontrer une congestion (duplicated ack ou fin de temporisation d'un segment émis non acquitté). Schématiquement, pendant le Slow Start, à chaque fois que l'émetteur reçoit l'acquittement d'un segment, il en émet 2. Dès qu'il a atteint la valeur de la moitié de la fenêtre de congestion, il ne fait progresser son débit que de façon linéaire, on entre dans la phase congestion avoidance.



3.3. TCP a une approche réactive pour estimer la bande passante disponible pour une connexion. En fait, le débit soumis augmente tant qu'une congestion n'apparaît pas (et tant qu'on ne dépasse pas le crédit alloué par le récepteur). Différentes études proposent des modifications de TCP.

TCP se voit ajouter des extensions. On examine une extension qui est relative au débit d'émission. Des statistiques sont faites par connexion.

- On calcule un débit attendu instantané de la façon suivante :

$$\text{Attendu}(t) = \text{TailleFenêtreCongestion}(t) / \text{BaseRTT} \text{ où}$$

- BaseRTT : délai Aller/Retour associé à un segment quand le réseau n'est pas congestionné (avant que les routeurs ne saturent à cause du trafic de cette connexion), on entend par délai Aller/retour (ou encore RTT pour Round Trip Time) le délai qui s'écoule entre l'émission du segment et l'arrivée de son acquittement. BaseRTT est le plus petit RTT mesuré. Il est mesuré à chaque fois qu'on émet un segment, si lors du retour de l'acquittement du segment, le RTT associé à celui-ci est inférieur à BaseRTT, alors BaseRTT prend cette nouvelle valeur.
- TailleFenêtreCongestion(t) est la valeur courante de la fenêtre de congestion de l'émetteur.

- On mesure le débit d'émission réel instantané de la façon suivante :

$$\text{Mesuré}(t) = \text{TailleSegmentEmis}(t) / \text{RTTduSegment}$$

$$\text{Soit Diff}(t) = \text{Attendu}(t) - \text{Mesuré}(t)$$

3.3.1. Expliquer pourquoi Diff(t) ne peut être négatif. (1 point)

Réponse :

BaseRTT est le plus petit RTT par définition.

Le volume d'octets émis est inférieur ou égal à la taille de la fenêtre de congestion.

Par conséquent $\text{Attendu}(t) > \text{Mesuré}(t)$

3.3.2. Deux seuils sont définis : $A < B$

Si $\text{Diff}(t) < A$ alors la fenêtre de congestion augmente linéairement.

Si $A < \text{Diff}(t) < B$ alors rien n'est fait.

Si $B < \text{Diff}(t)$ alors la fenêtre de congestion diminue linéairement.

3.3.2.a A quoi correspondent les seuils A et B ? (2 points)

Réponse :

A et B sont des valeurs de référence fixées vis-à-vis de l'écart entre le débit attendu et le débit mesuré.

Si la différence est trop grande, le débit mesuré est trop loin du débit attendu, c'est que le réseau est congestionné, on doit réduire le débit de l'émetteur.

Si la différence est petite, le débit mesuré est proche du débit attendu, le réseau n'est pas congestionné, on peut augmenter le débit de l'émetteur.

3.3.2.b Expliquer en quoi, cette modification vous semble permettre d'augmenter l'efficacité de TCP vis-à-vis de l'occurrence des congestions. (1 point)

Réponse :

Ce mécanisme contribue à améliorer TCP :

Il n'y a plus besoin de provoquer la congestion du réseau pour estimer la bande passante disponible. Si A et B sont correctement paramétrés, l'émetteur est capable d'anticiper la congestion réelle du réseau et de réduire son débit. Ceci évite la retransmission des segments victimes des congestions. Le slow start est aussi déclenché moins souvent. Donc, c'est plus efficace que le fonctionnement actuel de TCP.

Par ailleurs, avec A et B, on encadre le débit de la connexion.

3.4. UDP et gestion de la congestion dans Internet.

3.4.1 Quel type de mécanisme faudrait-il ajouter dans le protocole UDP ou en dehors du protocole UDP pour qu'il ne contribue pas à la saturation d'Internet quand un utilisateur tente d'émettre à un fort débit indépendamment de la charge du réseau. (1 point)

Réponse:

Côté émetteur, le protocole UDP devrait pouvoir réduire le débit émis vers l'Internet quand il se produit un phénomène de congestion. En l'état, ceci n'est pas possible, ce mécanisme doit plutôt être ajouté au-dessus de UDP.

Hors réponse : C'est à l'aide des informations du protocole RTCP associé à RTP qu'on peut arriver à le faire.

3.4.2 Comment devrait alors se comporter une relation ou une "connexion" de cette nouvelle version d'UDP du point de vue du partage de la bande passante de l'Internet vis-à-vis des connexions TCP co-existant sur le réseau ? La réponse doit uniquement considérer le point de vue de la couche transport. Il faut donc imaginer que le comportement au niveau transport est indépendant d'un mécanisme de réservation de ressources tel que RSVP, cela doit fonctionner même sur un réseau de type best effort. (1 point)

Réponse:

Le partage de la bande passante entre deux connexions (une UDP et l'autre TCP) ou entre les différents types de flux devrait être équitable. Globalement, si un flux occupe plus de bande passante que l'autre, les utilisateurs de l'autre protocole sont spoliés.