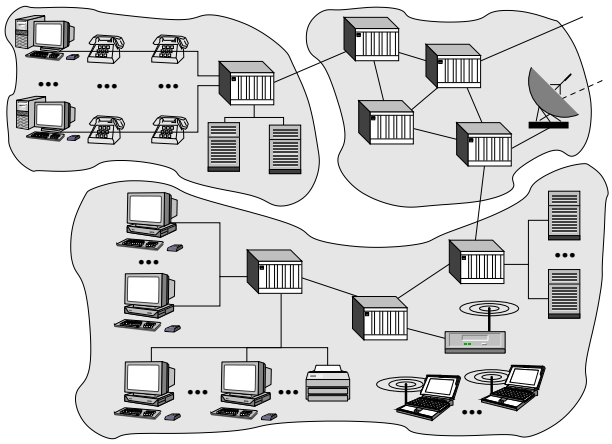
Master Informatique de l'Université Pierre & Marie Curie

U.E. **ARES** (MI011)

Architecture des Réseaux



Sujets TD, sujets TME et quelques annales.
version 5.4

Olivier Fourmaux
(olivier.fourmaux@upmc.fr)

UPMC PARIS UNIVERSITAS

U.E. ARES - Travaux Dirigés n°1

Applications

1 Exercices

1. Qu'est-ce qu'un protocole applicatif ? Citez des exemples.
2. Quels programmes accédant au réseau utilisez-vous couramment ? Savez-vous quels protocoles applicatifs ils utilisent ?
3. Sur quel modèle de communication s'appuient principalement les applications actuelles ? Comment identifier les rôles des participants pour les applications citées précédement ?
4. Décrivez les grandes catégories d'applications utilisant les réseaux. Pour chacune d'elles, indiquez les besoins en termes de débit, de tolérance à la variation de débit, de sensibilité aux pertes et de contraintes temporelles.

2 Terminal virtuel

1. Quel est l'intérêt des applications d'accès à distance dont le terminal virtuel fait partie ?
2. Quelles contraintes peuvent poser ce type d'applications ? Citez des exemples.
3. Pour une application reposant sur le protocole TELNET, quel type de service réseau va être utilisé ? Et quelles informations vont êtres échangées ?

3 Messagerie électronique

1. Alice accède à sa messagerie par le web. Elle envoie un message à Bob. Ce dernier rapatrie ses messages sur son ordinateur de bureau lorsque celui-ci est allumé. Décrivez les échanges d'informations ainsi que les protocoles mis en jeu.
2. Rappelez la structure du message échangé par les serveurs de courrier électronique.
3. Pouvez-vous décoder ce champ d'entête ?

   ```
   Subject: =?iso-8859-1?B?Qydlc3QgcGFzIGZhY2lsZSAhCg==?=
   ```

4. Vous envoyez un courrier électronique avec un message en texte et en HTML, accompagné de quelques pièces jointes : une image au format PNG, du son encodé en MP3, une vidéo MPEG et un fichier WAD pour Doom. Quelles lignes d'entête devrait-on observer dans le message ?

4 Web

1. Expliquez les étapes nécessaires à la récupération d'une page web.
2. Quelles optimisations prévues par HTTP 1.1 utilisent les serveurs web actuels pour réduire la latence des échanges ?
3. Une autre possibilité pour réduire le temps de réponse est l'utilisation de la mise en mémoire cache. Décrivez où intervient ce mécanisme et pour quels types d'objets il est intéressant.

5 Le protocole FTP – découverte du RFC 959

Pour les questions suivantes, un texte est fourni à la fin du présent document, à la suite des sujets de TD et TME. C'est un **RFC** (*Request For Comment*) produit par les groupes de travail de l'**IETF** afin d'assurer un processus de standardisation.

1. Que pouvez-vous dire sur la forme du RFC 959 ? Quelles sont les différentes sections abordées dans ce document ?
2. Précisez l'architecture de communication de FTP. Pourquoi dit-on que les informations de contrôle circulent "hors-bande" ?
3. Quelles sont les différentes commandes à la disposition du client ?
4. Pouvez-vous citer les différents types d'erreur que peut signaler FTP ? Comment les informations d'erreur sont-elles transmises ?

6 Le système DNS

1. Chaque serveur sur l'Internet possède au moins un serveur de nom local et un serveur de nom de référence (*authoritative*). Quel est le rôle joué par chacun d'eux au sein du système DNS ?
2. Le serveur Web et le serveur de courrier d'une institution peuvent-ils partager le même nom (par exemple `toto.org`) ?
3. En surfant sur le Web, vous cliquez sur un lien menant à une page qui vous intéresse. Votre machine ne connaît pas l'adresse IP correspondant à l'URL de la page demandée et celle-ci ne se trouve pas dans le cache de votre navigateur. Si n serveurs DNS sont visités de manière **itérative** avant d'obtenir l'adresse IP recherchée, en combien de temps peut-on escompter voir apparaître la page (le temps de transmission de l'objet est négligeable) ?

   Faire un chronogramme.
4. En reprenant le problème précédent, supposez que le fichier HTML à récupérer contienne dorénavant trois objets de taille réduite stockés sur le même serveur. En négligeant les temps de transmission de ces objets, combien de temps est-il nécessaire pour obtenir la page (i) avec le protocole HTTP en mode non persistant non parallèle ; (ii) avec le protocole HTTP en mode non persistant avec connexions parallèles ; (iii) avec le protocole HTTP en mode persistant sans pipelinage ; (iv) avec le protocole HTTP en mode persistant avec pipelinage ?

7 Administration SNMP

1. Pour un administrateur réseau, quel est l'intérêt d'utiliser des outils de gestion du réseau ? Citez plusieurs possibilités.
2. Représentez sur un schéma intégrant quelques éléments à administrer les mécanismes de bases de l'administration de réseau (éléments applicatifs, messages échangés...) :
3. Définissez les termes suivants : Station d'administration, Equipement administré, Agent d'administration, Base d'information de gestion (MIB), Structure des informations de gestion (SMI) et Protocole de gestion du réseau.
4. Représentez sur un schéma intégrant quelques éléments à administrer les mécanismes de bases de l'administration de réseau (éléments applicatifs, messages échangés...) :
5. Quels sont les PDU utilisés par SNMP ? Quels messages sont utilisés pour des requêtes/réponses ou des envois spontanés ? Quel est la différence entre ces deux types d'échanges ? Quels en sont les avantages et les inconvénients ?

6. Les PDU SNMP sont décris en ASN.1 et utilisent l'encodage BER pour obtenir les octets transmis. Rappeller ce système d'encodage et appliquez-le à la structure simple suivante : `{{nom, "Dupont"} {credit, 277}}` ?
7. A votre avis, pour quelles raisons utilise-t-on UDP plutôt que TCP pour le transport des PDU SNMP ?
8. Dans la suite, un administrateur souhaite gérer les routeurs du réseau de son entreprise grâce au protocole SNMP. Ce réseau fonctionne sous TCP/IP et interconnecte plusieurs réseaux locaux à l'aide de routeurs dont le service SNMP est activé.
   (a) Proposez un mécanisme pour découvrir les différentes machines présentes sur le réseau local de la station d'administration :
   (b) Expliquez comment vérifier qu'une machine est bien un routeur (la MIB-II standard définit un objet simple `ipForwarding`).
   (c) Comment obtenir le nom de ces routeurs (la MIB-II standard définie l'objet `system.sysName` de type chaîne de caractère...) ?
   (d) Sachant que la MIB-II propose un objet table `ipAddrTable` qui référence toutes les interfaces d'une machine avec leurs paramètres IP (addresse IP, masque de réseau, adresse de diffusion...), précisez comment obtenir toutes les adresses IP (champ `ipAdEntAddr`) d'un routeur.
   (e) Précisez comment modifier la valeur du masque de réseau (champ `ipAdEntNetMask`) associé à l'interface 3 d'un routeur (les entrées de l'objet table `ipAddrTable` sont indexés par le numéro de cette interface).
   (f) Connaissant les informations précédentes disponibles dans la MIB-II, proposez un mécanisme général pour découvrir tous les routeurs du réseau de l'entreprise. Indiquez les limitations de votre approche.
   (g) Quelles limitations notez-vous à cette approche ?

8 Peer-to-peer *(facultatif)*

1. Quelle est l'infrastructure réseau dans le contexte d'un système de partage de fichier *peer-to-peer* ? De tels réseaux sont-ils équipés de routeurs ? Comment accède-t-on à ces réseaux ? Comment la structure de ces réseaux est-elle mise à jour ?
2. Vous êtes en train de télécharger des MP3 à l'aide d'un système *peer-to-peer*. Votre machine est très puissante et le goulet d'étranglement se situe au niveau de votre accès réseau. Supposons que vous utilisiez une connexion ADSL 1024/256 Kbps. Pendant votre téléchargement, 16 utilisateurs accèdent à des données de votre machine. Comment leurs opérations interfèrent-elles avec les vôtres ?
3. Vous utilisez un système de recherche par inondation (type Gnutella) sur un réseau *peer-to-peer* où chaque poste est connecté à un nombre de voisin maximum est N et le nombre de saut maximum est K. Trouvez le nombre maximal de messages qui peuvent être générés par la requête.

U.E. ARES - Travaux Dirigés n°2

Couche transport: UDP et TCP

1 Introduction

1. Un client web souhaite accéder à un document dont il connait l'URL. L'adresse IP du serveur concerné est initialement inconnue. Quels protocoles des couches application et transport sont requis pour satisfaire cette demande ?

2 Protocole en mode non connecté, UDP

1. Rappelez les caractéristiques d'un protocole en mode non connecté.
2. Précisez les principales caractéristiques du protocole UDP.
3. Pour quelles raisons un développeur pourrait-il préférer le protocole UDP à TCP ?
4. A votre avis, une application peut-elle bénéficier d'un service de transfert de données fiable si elle repose sur UDP ? Justifiez votre réponse.

3 Protocole en mode orienté connexion, TCP

1. Rappelez les caractéristiques d'un protocole en mode orienté connection.
2. Explicitez les mécanismes nécessaires à la réalisation d'un transfert de données fiable.
3. Indiquez les principales caractéristiques du protocole TCP.

3.1 Gestion de connexion

1. Représentez le diagramme d'établissement d'une connexion réseau. Discutez du nombre de messages nécessaires. Dans le contexte de TCP, pourquoi procéder à un échange en trois phases ?
2. Quel sont les possibilités de numérotation des segments ? Dans le contexte de TCP, comment évoluent les numéros de séquence ? Deux segments successifs peuvent-ils contenir le même numéro de séquence ? Et en l'absence de données transmises, le numéro de séquence peut-il augmenter ?
3. Pourquoi ne pas commencer la numérotation de séquence à 0 ?
4. Quels sont les possibilités de terminaison d'une connexion ? Représentez les diagrammes correspondants.

3.2 Estimation du RTT d'une connexion

Lorsque l'on utilise le protocole TCP, le choix du RTT (*Round Trip Time*) est important puisque la détection de perte en découle directement et que les divers mécanismes de contrôle qui vont influer sur le débit d'émission en dépendent. Le calcul du RTT peut se faire à l'aide de la formule suivante $RTT = \alpha * RTT_{mesure} + (1-\alpha) * RTT_{ancien}$ avec α le coefficient de lissage.

1. Comment TCP mesure-t-il le délai aller-retour (RTT_{mesure}) pour un segment donné ?
2. Montrez que l'effet d'une valeur mesurée pour le RTT se réduit exponentiellement avec le temps.
3. Quelle est l'intérêt d'utiliser cette formule comparée à une moyenne mobile dans laquelle le RTT est la moyenne calculée sur une fenêtre de longueur L ?

4. Quelles sont les conséquences d'une valeur de α proche de 1 et proche de 0 ?
5. Quelles sont les précautions à prendre lors de la mesure du délai aller-retour d'un segment donné ?
6. A votre avis, quel est l'utilité de l'option TCP `timestamp` ? Pourquoi est-il conseillé d'utiliser cette option (on pourra consulter ultérieurement le RFC 1323 pour plus de détails) ?

3.3 Calcul du RTO de TCP *(facultatif)*

1. La première approche pour déterminer la valeur du temporisateur de retransmission RTO (*Retransmission TimeOut*) est $RTO = n*RTT$. Quelles sont les précautions à prendre quant au dimensionnement du paramètre n ?
2. La deuxième approche utilise $RTO = RTT + \delta D$ avec généralement $\delta = 4$.
   $D = \beta(|RTT_{mesure} - RTT_{ancien}|) + (1 - \beta)D_{ancien}$ avec généralement $\beta = 1/4$.
   Cette approche consiste à calculer la variance du RTT. Quelle est l'amélioration apportée ?
3. Comment calculer le RTO lorsqu'il y a des pertes ?

4 Contrôle de congestion TCP

TCP est utilisé pour le transport fiable de données dans l'Internet. Nous avons précédemment étudié la gestion des connexions et mécanismes TCP. Dans les exercices suivants, nour allons nous intéresser à un autre comportement fondamental de TCP : le contrôle de congestion.

4.1 Détection de la congestion

La conception de TCP date de la fin des années 70. Plusieurs algorithmes de contrôle de congestion ont été ajoutés depuis, principalement suite aux travaux de Van Jacobson publiés en 1988. Ces derniers continuent à évoluer dans les différentes variantes de TCP. Les exercices proposés dans la suite sont fondés sur les dernières mises à jour : les RFC 2581 et 2582 d'avril 1999 (TCP newReno).

1. Pour TCP, quel phénomène indique une congestion dans le réseau ?
2. Que se passe-t-il dans un routeur pour susciter ce phénomène ?
3. Pour TCP, ce phénomène permet de déduire la congestion. Mais celui-ci peut aussi se produire quand il n'y a pas de congestion dans le réseau. Dans quels autres cas un tel phénomène peut-il apparaître ?
4. Si ce phénomène n'indique pas toujours une congestion, pourquoi TCP se base-t-il sur cette inférence ? Pourquoi n'utilise-t-on pas une approche où le routeur constatant la congestion envoie un message explicite à l'émetteur ?

4.2 Algorithmes de contrôle de congestion

Pour le contrôle de congestion, TCP utilise un seuil qui indique le débit au-dessus duquel la congestion risque de se produire. Ce seuil est exprimé par le paramètre `limiteSS` (en octets). Pour obtenir le débit seuil on divise `limiteSS` par le RTT (*Round Trip Time*). Le débit peut varier en-dessous et au-dessus du seuil `limiteSS`$/RTT$. L'émetteur maintient un paramètre `fenCong` (taille de la fenêtre de congestion) qui indique le nombre d'octets qu'il peut envoyer avant de recevoir un acquittement. Quand `fenCong` > `limiteSS`, l'émetteur fait particulièrement attention à ne pas provoquer de congestion.

1. Supposons que `limiteSS` soit à 5000 octets, `fenCong` est à 6000 octets, et la taille d'un paquet est de 500 octets. Un émetteur envoie douze paquets de 500 octets dans une période `RTT`, et reçoit douze acquittements (un pour chaque paquet). Que deviennent les valeurs de `limiteSS` et `fenCong` ? Comment s'appellent ces changements de valeurs ?

2. Supposons que `limiteSS` soit toujours à 5000 octets, que `fenCong` est maintenant à 14.000 octets, que l'émetteur envoie 14.000/500 = 28 paquets, et que l'émetteur reçoive une indication de congestion avant de recevoir le premier acquittement. Que deviennent les valeurs de `limiteSS` et `fenCong` ? Comment s'appellent ces changements de valeurs ?
3. Nous venons de voir comment augmente et diminue `fenCong` en fonction de l'absence ou la présence d'indicateurs de congestion. Comment s'appelle cet algorithme ? Sur quel principe repose cet algorithme ?
4. Au démarrage, et après avoir reçu une indication de congestion, la valeur de `fenCong` est plus petite que la valeur de `limiteSS`. Décrivez la manière permettant d'augmenter `fenCong` quand celle-ci est inférieure à `limiteSS`, en fonction de l'exemple suivant. Supposons que `limiteSS` soit égal à 3000 octets et que `fenCong` soit égal à 500 octets, la taille d'un paquet. L'émetteur a plusieurs paquets prêts à être envoyés. Combien de paquets envoie l'émetteur pendant la première période RTT ? S'il reçoit des acquittements pour tous ses paquets, que devient la valeur de `fenCong` ? Combien de paquets envoie l'émetteur pendant la deuxième période RTT ? S'il reçoit des acquittements pour tous ses paquets, que devient la valeur de `fenCong` ? En général, comment évolue la taille de `fenCong` ?
5. Comment s'appelle la période pendant laquelle `fenCong` est plus petit que `limiteSS` ?
6. Que devient la valeur de `limiteSS` si l'émetteur reçoit une indication de congestion pendant que `fenCong` est plus petit que `limiteSS` ?

4.3 Débit moyen d'une connexion TCP

Supposons que nous souhaitions effectuer un transfert de données de taille importante à travers une connexion TCP.

1. En négligeant la période pendant laquelle `fenCong` est plus petit que `limiteSS`, montrez que le débit moyen d associé à une connexion TCP est égal à :

$$d = \frac{3}{4}\frac{W * MSS}{RTT}$$

où W est la taille de la fenêtre (en segment) au moment de la congestion, MSS la taille d'un segment (supposée maximale), et RTT est le délai aller-retour (supposé constant durant la période de la transmission).

2. Montrer que le taux de pertes p est égal à :

$$p = \frac{1}{\frac{3}{8}W^2 + \frac{3}{4}W}$$

3. Montrer que si le taux de pertes observé par une connexion TCP est p alors, son débit moyen d peut être approximé par :

$$d = \frac{1,22 * MSS}{RTT\sqrt{p}}$$

4. Quels autres paramètres peuvent influer sur le débit d'une connexion TCP ?
5. Quelle utilité voyez-vous à la relation calculée dans la dernière formule de d ?

5 Etude de la latence d'un serveur web (*facultatif*)

Nous souhaitons étudier la latence liée à la réponse à une requête HTTP[1]. Nous faisons les hypothèses simplificatrices suivantes :

- Le réseau n'est pas congestionné (pas de pertes ni de retransmissions) ;
- Le récepteur est doté de tampons de réception infinis (limitation de l'émetteur uniquement liée à la fenêtre de congestion) ;
- La taille de l'objet à recevoir du serveur est O, un multiple entier du MSS (MSS à pour taille S bits) ;
- Le débit de la liaison connectant le client au serveur est R (bps) et on néglige la taille de tous les entêtes (TCP, IP et liaison). Seuls les segments transportant des données ont un temps de transmission significatif. Le temps de transmission des segments de contrôle (ACK, SYN...) est négligeable ;
- La valeur du seuil initial du contrôle de congestion n'est jamais atteinte ;
- La valeur du délai aller-retour est RTT.

1. Dans un premier temps, nous supposons que nous n'avons pas de fenêtre de contrôle de congestion. Dans ce cas montrez que la latence $L = 2RTT + O/R$.
2. Nous supposons maintenant une fenêtre de congestion **statique** de taille fixe égale à W. Calculez la latence dans ce premier cas : $WS/R < RTT + S/R$
3. Nous supposons toujours une fenêtre de congestion **statique** de taille fixe égale à W. Calculez la latence dans ce second cas : $WS/R > RTT + S/R$
4. Comparez la latence obtenue avec une fenêtre de contrôle de congestion **dynamique** (*slow-start*) avec celle sans contrôle de congestion.
5. Application numérique :

R	O/R	Latence sans S.S.	K'	Latence TCP
56 Kbps				
512 Kbps				
8 Mbps				
100 Mbps				

K' est le nombre de fenêtres envoyées avant de passer au second cas ($log2(1 + RTT * R/S)$). Considérez trois cas :

(a) S= 512 octets, RTT= 100 msec, O=**100 Koctets** (=200S) ;

(b) S= 512 octets, RTT= 100 msec, O=**5 Koctets** (=10S) ;

(c) S= 512 octets, RTT= **1 seconde**, O=5 Koctets (=10S).

[1] Latence d'une requête HTTP : laps de temps pour la création de la connexion et la récupération de l'intégralité de l'objet demandé.

U.E. ARES - Travaux Dirigés n°3

Couche réseau: IP, CIDR, NAT...

1 Exercices

1. Pour les adresses IP suivantes, précisez le masque réseau (*netmask*), le préfixe réseau (*netid*) et l'identificateur d'interface (*hostid*) : `192.33.182.182 /24`, `81.217.9.35 /20`, `192.19.67.59 /22` , et `203.19.40.199 /26`.
2. Pour chacun des réseaux suivants, indiquez l'adresse de la première machine, celle de la dernière et celle de diffusion (*broadcast*) : `192.33.182.0 /24`, `10.0.0.0 /16`, `81.188.160.128 /26` et `81.188.160.0 /19`.
3. Je dispose d'une classe B (`150.44.0.0`) pour mon entreprise. J'ai au plus 1000 machines par sous-réseau. Quel découpage en sous-réseaux dois-je appliquer pour maximiser le nombre de sous-réseaux. Pour le 1^{er}, le 2^{nd} et le dernier sous-réseau, indiquez le préfixe, l'adresse de la première machine, celle de la dernière et celle de diffusion.
4. Quel est la perte liée au découpage en sous-réseau (nombre d'adresses utilisables pour des interfaces) ?
5. J'ai 20 réseaux de 50 machines à adresser. Quel type d'agrégat CIDR vais-je demander à mon fournisseur de service ? Supposons que l'on m'attribue le premier bloc de taille adéquat du préfixe `60.44.32.0 /20`. Pour le 1^{er}, le 2^{eme} et le dernier sous-réseau, indiquez le préfixe, l'adresse de la première machine, celle de la dernière et celle de diffusion.
6. Dans l'exercice précédent, si l'on indique directement le nombre de machines que l'on souhaite adresser à son fournisseur, celui-ci nous donnera-t-il un bloc CIDR de la taille choisie ci-dessus ?

2 Plan d'adressage

Une entreprise souhaite intégrer son réseau dans l'environnement TCP/IP. Elle possède un site central avec 6 réseaux de 50 machines maximum. Elle souhaite aussi intégrer ses trois succursales, chacune avec 20 machines maximum.

1. Dans le cadre d'un adressage basé sur des classes, quelle solution proposez-vous pour adresser le site principal ? Illustrez votre solution par un schéma où vous utiliserez le réseau `18.0.0.0`, `132.66.0.0` ou `198.5.203.0` selon votre choix de classe d'adressage.
2. Nous souhaitons à présent intégrer les 3 succursales. Sachant qu'elles sont reliées chacune par une liaisons spécialisée entre leur routeur et celui du site principal, étendez votre solution et complétez le schéma.
3. Quelle sera la table de routage du routeur central ? D'un routeur de succursale ? D'une machine ?
4. Passons à un cas plus réaliste avec un adressage basé sur CIDR. Quelle solution proposez vous pour adresser le site principal ? Illustrez votre solution par un shéma ou vous utiliserez le premier bloc adapté du préfixe `88.5.100.0`.
5. Nous intégrons aussi les 3 succursalles. Sachant qu'elles sont reliées chacunes par une liaison spécialisée entre leur routeur et celui du site principal, étendez votre solution CIDR et complétez le schéma.
6. Avec CIDR, quelle sera la table de routage du routeur central ? D'un routeur de succursale ? D'une machine ?
7. Par soucis d'économie, l'entreprise décide d'abandonner les lignes spécialisés et connecte directement les succursales à l'Internet. Quelles modifications cela introduit-il au niveau de l'adressage, du routage et de la sécurité ?
8. Finalement, l'entreprise opte pour un adressage privé et des VPN. Quelle est sa motivation ? Quelles modifications cela introduit-il au niveau de l'adressage, du routage, de la sécurité et de l'accès à Internet ?

3 Un peu de NAT *(facultatif)*

Etudions à présent les accès à Internet pour un ensemble de machines avec une seule adresse publique.

1. Rappelez le principe du NAT, en insistant sur les problèmes associés à cette technique.
2. Si vous ne possédez qu'une adresse IP et souhaitez connecter deux réseaux à l'Internet, comment pouvez vous procéder ? Quelle sera la configuration du routeur ? Comment accéder au serveur Web situé dans le premier réseau à partir de l'extérieur ? Les deux réseaux peuvent-ils communiquer ensemble ?
3. Dans le problème précédent, comment l'entreprise peut-elle fournir un accès à internet avec un adressage privé ? Comment serait modifiée la table de routage du routeur principal ?

U.E. ARES - Travaux Dirigés n°4

Architectures Supports

1 Ethernet partagé

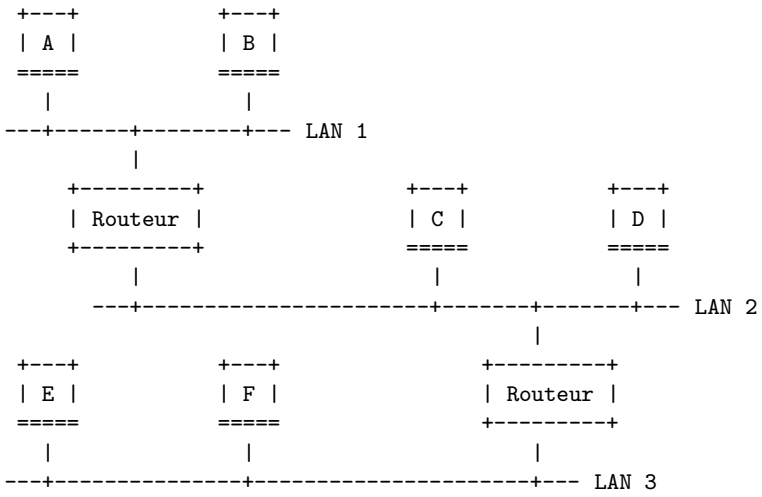
1. Rappelez l'algorithme d'accès au médium partagé **CSMA/CD** qui est utilisé dans les réseaux Ethernet.
2. Comment est calculée la longueur maximale d'un bus **Ethernet** à 10 Mbps ? La taille minimale d'une trame Ethernet est fixée à 64 octets par le standard Ethernet V2 et par les normes IEEE 802.3 / ISO 8802.3.
3. La technologie **Fast Ethernet** (100 Mbps) est compatible avec les réseaux Ethernet classiques. Quelle est la longueur maximale d'un brin avec Fast Ethernet ?
4. La technologie **Gigabit Ethernet** (1000 Mbps) est également compatible avec Ethernet. Quelle est la longueur maximale d'un brin Gigabit Ethernet ? Comment peut-on augmenter cette taille minimale ?

2 Interconnexion de réseaux Ethernet

1. Comparez les fonctionnalités d'un répéteur, d'un concentrateur (*hub*), d'un pont et d'un commutateur (*switch*). Lesquels permettent d'augmenter la taille d'un réseau Ethernet ?
2. Quel est l'intérêt d'utiliser un routeur pour interconnecter des réseaux Ethernet ?
3. Dans un commutateur qui gère les VLAN, expliquez comment deux machines de VLAN différents peuvent communiquer.

3 Problème d'adressage

Supposons 3 réseaux interconnectés par 2 routeurs de la manière suivante :



1. Recopiez ce schéma en y intégrant les adaptateurs réseaux.
2. Puis associez une adresse LAN à chaque adaptateur.
3. Attribuez une adresse IP à chaque interface, en utilisant les adresses réseaux suivantes : `LAN 1` : 81.177.11.0 /24 ; `LAN 2` : 81.177.22.0 /24 et `LAN 3` : 81.177.33.0 /24.

4. Un datagramme IP est envoyé du serveur `A` au serveur `F`. Toutes les tables ARP sont à jour dans les différents équipements impliqués. Énumérez les étapes successives liées à cette opération.
5. Supposons maintenant que la table ARP de l'expéditeur soit vide. Quelles modifications cela induit-il ?

4 Liaison point-à-point

1. Citez quelques protocoles de la couche liaison.
2. Quelles sont les fonctionnalités de la couche liaison dans le contexte du point-à-point ?
3. Comment est réalisé le découpage en trame (*framing*) selon les unités de transfert utilisées ?
4. De quelle manière est réalisé l'adressage ?
5. Quel est le protocole défini par l'IETF pour les liaisons point-à-point ? Quels sont les différents protocoles qui lui sont associés ? Décrivez leurs rôles respectifs.
6. Pour utiliser efficacement les liaisons à bas débit avec TCP/IP, quels mécanismes peut-on mettre en œuvre ? A quelle taille minimum peut-on réduire la taille d'un paquet ?
7. Quel est l'intérêt d'utiliser des liaisons point-à-point dans le contexte de réseaux offrant des accès multiples ? Justifiez votre réponse et citez plusieurs protocoles associés à cette technique.

5 PPP sur RTC

Nous nous situons dans le contexte de la mise en place d'une connexion IP sur une ligne RTC. Après l'établissement du circuit téléphonique (avec une signalisation externe) et la négociation des modems, une liaison asynchrone basée sur des octets est opérationnelle. Nous étudions dans la suite les différents échanges protocolaires liés à la couche liaison :

1. Quelles sont les premières trames émises. Pouvez vous détailler leurs contenus sachant que c'est la machine distante qui démarre et qu'elle souhaite négocier un MRU à 1500 octets, une table ACCM nulle, la compression du champs PPP `protocol` et la suppression des champs `address` et `control`. La machine locale négocie en retour les mêmes options sans le MRU.
2. Quel protocole peut intervenir immédiatement après ce premier échange ? Détaillez les possibilités et mécanismes mis en œuvre.
3. L'équipement distant négocie ensuite la compression des en-têtes TCP/IP de type Van Jacobson et propose l'adresse `10.1.1.1`. Quel protocole entre en jeux ? Détaillez les trames échangées.
4. Les paramètres réseaux étant établis, un échange DNS a lieu. Détaillez les trames échangées.
5. Puis un échange HTTP... Détaillez aussi les trames échangées.

6 PPP sur ADSL

De manière similaire à une connexion PPP sur RTC, les fournisseurs d'accès à Internet proposent actuellement des accès PPP sur ADSL. Différentes propositions sont envisageables. Nous les étudions dans la suite :

1. La proposition initiale repose sur l'approche PPPoE et l'utilisation d'un pont/modem Ethernet/ADSL. Décrivez cette approche et représentez l'architecture correspondante.
2. Détaillez les différents échanges protocolaires depuis le démarrage de la connexion jusqu'à la réception d'un courrier électronique d'un abonné
3. Combien d'adresses IP sont associées au client avec cette approche ? Citez une possibilité pour augmenter cette quantité. Quelle limitation impose cette approche ?

4. Une autre approche repose sur PPPoA. Quelle différence cela induit-il au niveau de l'équipement du client ? Représentez l'architecture correspondante.
5. Des opérateurs proposent de ne plus utiliser la technologie ATM, ce qui est possible si les réseaux d'accès utilisent une autre technologie. A votre avis, quel est cette technologie ? Proposez un schéma de l'architecture correspondante.
6. Et si l'opérateur n'utilise plus PPP ? Représentez la nouvelle architecture et discutez des avantages ou inconvénients de cette approche.

7 Problème de câblage *(facultatif)*

Nous proposons d'étudier la mise en place d'un réseau local pour l'entreprise représenté ci-dessous :

Voici le matériel dont vous disposez :

- Câble coaxial fin (1 EUR/m)
- Câble 4 paires torsadées UTP (1 EUR/m)
- Câble 1 paire de fibres optiques (2 EUR/m)
- NIC BNC : adaptateur coaxial fin (30 EUR)
- NIC RJ45 : adaptateur UTP (50 EUR)
- Répéteur : 2 ports BNC (500 EUR)
- Répéteur multiport : 8 BNC (1000 EUR)
- Répéteur optique multiport : 6 ports (2000 EUR)
- Pont : 2 ports de n'importe quelle combinaison de technologie (1200 EUR)
- Hub : concentrateur de 36 ports UTP (2000 EUR)
- Hub : concentrateur 6 ports fibres et 24 ports UTP (5000 EUR)
- Serveur de fichiers type PC industriel "rackable" : 30 utilisateurs max. (4000 EUR)

Nous nous fixons les contraintes suivantes :

- Chaque département doit avoir accès aux ressources de tous les autres départements.
- Le trafic généré par un employé sur le segment de son département ne doit pas affecter le reste du réseau.

- Chaque serveur ne peut prendre en charge plus de 30 utilisateurs et ne peut être partagé entre plusieurs départements.
- Tous les équipements réseaux doivent être rassemblés dans les armoires de répartition.

1. Élaborez un câblage[1] en coaxial fin (avec fibres optiques si nécessaire). Représentez votre architecture sous forme d'un diagramme, faites une liste du matériel nécessaire (en précisant les quantités et longueurs) et proposez un devis.
2. Élaborez cette fois un câblage en paires torsadées (UTP).
3. Quelles modifications structurelles imposerait le passage à la commutation ?

[1] Supposez que l'on câble le long des murs et que l'on compte 50m de câble en moyenne vers chaque machine pour irriguer un département.

U.E. ARES - Travaux sur Machine Encadrés n°1

Introduction à l'analyse de trames

1 Analyse manuelle d'une trame capturée sur un réseau Ethernet

Etudions une trame observée sur un réseau local Ethernet. Cette trace est obtenue à l'aide d'un analyseur multiprotocolaire (un *sniffer*, tel l'outil `tcpdump` utilisé sur une machine connectée à ce réseau). Les traces sont habituellement présentées selon trois colonnes :

❶	❷	❸
0000	00 01 02 a5 fb 3a 00 01 02 a5 fc 8d 08 00 45 00	...#û:.. .#ü...E.
0010	00 3c ec 26 40 00 40 06 cc cd 0a 21 b6 b6 84 e3	.<ì&@.@. ÌÍ.!ll.ã
0020	.. ..	.. ..

❶ indique, avec 4 chiffres hexadécimaux, le **rang** du premier octet de la ligne courante dans la trame ;

❷ affiche la **valeur hexadécimale** de 16 octets capturés (un octet est codé sur deux chiffres hexadécimaux) ;

❸ représente les caractères ASCII correspondants aux 16 octets de la seconde colonne (la correspondance n'est significative que lorsque du texte lisible se trouve encodé dans ces octets).

Les trames Ethernet présentées ne comportent ni préambule, ni CRC.
Veillez à respecter les conventions de représentation :
- **Adresses Ethernet** : hexadécimale double pointée (ex : `00:50:04:ef:6b:18`)
- **Type Ethernet** : hexadécimale (ex : `0x0806`)
- **Adresses IP** : décimale pointée (ex : `10.1.1.3`)
- **Numéro de protocole** et **numéro de port** : décimale (ex : `17`)

Trace à analyser :

```
0000  00 01 02 a5 fb 3a 00 01  02 a5 fc 8d 08 00 45 00   ...#û:.. .#ü...E.

0010  00 3c ec 26 40 00 40 06  cc cd 0a 21 b6 b6 84 e3   .<ì&@.@. ÌÍ.!¶¶.ã

0020  3c 0d 0e b5 00 50 a9 55  92 64 00 00 00 00 a0 02   <..é.P©U .d.... .

0030  3e bc a3 74 00 00 02 04  05 b4 04 02 08 0a 08 39   >¼£t.... .´.....9

0040  91 16 00 00 00 00 01 03  03 00                     ........ ..
```

Les structures des principaux protocoles rencontrés sont rappelées dans la suite (page 4).

1. Représentez la structure de la trame en dessinant directement les délimitations sur la trace à analyser.
2. Quelles informations de niveau liaison observez vous ?
3. Représentez la structure du paquet directement sur la trace à analyser.
4. Le paquet contient-il des options ? Justifiez.
5. Quelles sont la source et le destinataire du paquet ?

6. Représentez la structure des données transportées par le paquet directement sur la trace.
7. Quel est le protocole de transport utilisé ? Quels sont les ports utilisés ? Quelles sont leurs significations ?
8. *Il n'y a pas de documentation correspondant à la couche application à la fin du document,* malgré cela, pouvez vous observer des informations associées à ce niveau dans la trace ?

2 Utilisation de l'analyseur de trame `ethereal/wireshark`

Pour analyser les informations circulant sur les réseaux, les administrateurs disposent donc des *sniffers*. Ces outils se présentent sous la forme d'un équipement pouvant se connecter directement sur le réseau ou d'un logiciel installé sur un ordinateurs reliés au réseau à analyser.

Lorsque le réseau à étudier est de type à **médium partagé** — Ethernet par exemple — l'interface de toute machine connectée "voit" potentiellement tout le trafic échangé sur le réseau local. Pour ne pas juste "voir", mais "regarder" explicitement le trafic afin de récupérer les trames observées (y compris celles qui ne lui sont pas destinées) pour les analyser, un mode de "promiscuité" est habituellement disponible sur l'interface réseau. Ce mode ne perturbe ni le trafic du réseau, ni celui de la machine support. Cela permet ainsi d'ajouter la fonction "*sniffer*" à cette dernière avec un logiciel adéquat.

Les logiciels `ethereal` ou `wireshark` [1] sont des analyseurs de protocoles. Ceux-ci peuvent utiliser directement l'interface de votre machine pour réaliser la capture de toutes les informations circulant sur le réseau local sur lequel vous êtes connecté[2]. Pour des raisons évidentes de sécurité, nous vous fournirons des captures déjà réalisées. Vous utiliserez alors la fonction principale de ces outils : l'analyse multiprotocolaire.

2.1 Introduction à `ethereal/wireshark`

Sur une machine de la salle de T.M.E. sous le système **GNU/Linux**, accédez à votre compte utilisateur. Rechercher dans les menus d'application le sous-menu réseau/Internet. Cliquez sur l'item `ethereal/wireshark` et demandez une exécution sans les droits administrateur[3] (obtenez les informations de base avec la commande `man ethereal/wireshark`). Initialement l'écran est vide car aucune capture n'a été réalisée ou chargée. Cliquez sur le menu File et sélectionnez Open. Une fenêtre de sélection de fichier "Open Capture Files" apparaît. Sélectionnez le fichier :

`/Infos/lmd/2008/master/ue/ares-2008oct/tme1.dmp.gz`

Ne pas spécifier de filtres dans le champ Filter (nous y reviendrons plus loin). Désactivez : ❑Enable MAC name resolution, ❑Enable network name resolution et ❑Enable transport name resolution. Cliquez finalement sur [Ouvrir] : La trace d'une capture précédemment réalisée est chargée et vous aller pouvoir l'analyser.

1. Décrivez le contenu des trois fenêtres proposées par `ethereal/wireshark`.
2. Dans quels formats sont représentés les données de la troisième fenêtre ?
3. Quels sont les différents protocoles que vous pouvez observer dans la capture affichée ?

2.2 Filtres d'affichage et de coloriage `ethereal/wireshark`

1. A l'aide du `man`, décrivez la syntaxe utilisée par les filtres d'affichage et de coloriage *Display filters* (à ne pas confondre avec les filtres de capture qui répondent à une autre syntaxe que nous n'utiliserons pas).
2. Combien de protocoles est capable d'analyser la version d'`ethereal/wireshark` que vous utilisez ?
3. Décrivez un filtre qui ne sélectionne que les trames ARP de/vers l'interface avec l'adresse MAC `00:10:a4:86:2d:0b`. Pour vous aider, le menu Analyse propose Display Filters... qui affiche une fenêtre d'édition de filtre. Le bouton [+Expression] permet une aide à la création de l'expression correspondante. Attention, après avoir entré le nom du filtre et l'expression, il faut ensuite cliquer sur [Nouveau] pour valider la création du filtre. Appliquer ce filtre d'affichage pour n'observer que les trames correspondantes.
4. Supprimer le filtre précédent et coloriez en vert les trames ARP.

[1] `ethereal` et `wireshark` sont des logiciels libres. Ils sont disponibles sur un grand nombre de plates-formes matérielles et systèmes d'exploitation (outre les machines à architecture **i368** avec système **GNU/Linux** que vous utilisez actuellement). Vous pouvez les télé-charger sur `www.ethereal.com` ou `www.wireshark.org`.

[2] L'interface Ethernet doit être configurée par le logiciel dans le mode "*promiscuous*" afin d'accéder à tout le trafic diffusé sur le segment où votre machine est connectée. Pour changer de mode, l'application doit être exécutée avec les privilèges de l'administrateur.

[3] En cas d'échec, généralement lié à l'exécution proposée par votre environnement qui essaye d'utiliser le mode administrateur (mode par défaut la commande `ethereal/wireshark` relative au `$PATH` local), vous pouvez démarrer en mode textuel (dans un terminal) : Tapez alors la commande `/usr/sbin/ethereal` ou `/usr/sbin/wireshark`.

2.3 Analyse d'un trafic HTTP

Dans la continuité de la trame étudiée manuellement dans la section précédente :

1. Sélectionnez **toutes** les trames relatives à la connexion, puis affichez en rouge seulement celle contenant des données HTTP.
2. Décrivez ce que vous observez, et si il y a plusieurs connexions, quelle est leur relation ?
3. Peut-on visualiser le contenu applicatif d'une connexion TCP ?

Structure de la trame Ethernet

```
.......+-48bits--+-48bits--+16b-+- - - - -+......
.(Pré.)| adresse | adresse |type| données |(CRC).
.      | dest.   | source  |    |         |     .
.......+---------+---------+----+- - - - -+......
```

```
Quelques types : 0x0200 = XEROX PUP
                 0x0800 = DoD Internet
                 0x0806 = ARP
                 0x8035 = RARP
```

Structure ARP

```
+16b-+-16b-+8b+8b+16b+--lgHW--+lgP-+--lgHW--+lgP-+
|type|type |lg|lg|Op |Emetteur|Emt.|Récept. |Rcpt|
|HW  |Proto|HW|P |   |adr. HW |adrP|adr. HW |adrP|
+----+-----+--+--+---+--------+----+--------+----+
```

```
Quelques types : 0x0001 = Ethernet
                 0x0800 = DoD Internet
Opérations : 0x0001 = Requête
             0x0002 = Réponse
```

Structure du paquet IP

```
<-----------------32bits----------------------->
<-4b->      <--8bits--><--------16bits-------->
+-----+-----+-----------+----------------------+
| Ver | IHL | TOS       | Longueur totale (octet)
+-----+-----+-----------+--+-------------------+
| Identificateur        |Fl| FO                |
+-----------+-----------+--+-------------------+
| TTL       | Protocole | Somme de ctrl (entête)|
+-----------+-----------+----------------------+
| Adresse Source                               |
+----------------------------------------------+
| Adresse Destination                          |
+----------------------------------------------+
...     Options
+----------------------------------------------+
...     Données
+----------------------------------------------+
```

```
Ver = Version d'IP
IHL = Longueur de l'en-tête IP (en mots de 32 bits)
TOS = Type de service (zero généralement)
Fl (3 premiers bits) = Bits pour la fragmentation
 * 1er = Reservé
 * 2me = Ne pas fragmenter
 * 3me = Fragment suivant existe
FO (13 bits suivants) = Décalage du fragment
TTL = Durée de vie restante
```

```
Quelques protocoles transportés :
        1 = ICMP                 8 = EGP
        2 = IGMP                11 = GLOUP
        4 = IP (encapsulation)  17 = UDP
        5 = Stream              36 = XTP
        6 = TCP                 46 = RSVP
```

Structure du paquet ICMP

```
<-----------------32bits----------------------->
+-----------+-----------+----------------------+
|  Type     |  Code     | Somme de contrôle (msg)
+-----------+-----------+----------------------+
|  Variable (généralement non utilisé)         |
+----------------------------------------------+
...     Datagramme original + 8 octets
+----------------------------------------------+
```

```
Quelques types ICMP :  8 = Demande d'écho
                       0 = Réponse d'echo
                      11 = Durée de vie écoulée
                      12 = Erreur de paramètre
```

Structure de segment TCP

```
<-----------------32bits----------------------->
<-4b->         <-6bits-><--------16bits-------->
+----------------------+----------------------+
| Port Source          |  Port Destination    |
+----------------------+----------------------+
| Numéro de Séquence                          |
+---------------------------------------------+
| Numéro d'Acquittement                       |
+-----+-------+---------+----------------------+
| THL |       |  Flag   |  Taille Fenêtre      |
+-----+-------+---------+----------------------+
| Somme de ctrl (message)  Pointeur d'Urgence |
+----------------------+----------------------+
...      Options
+---------------------------------------------+
...      Données
+---------------------------------------------+
```

```
THL = Longueur de l'entête TCP sur 4 bits (*32bits)
Flags = indicateur codé sur 6 bits gauche à droite
 * 1er = Données urgentes
 * 2me = Acquittement (ACK)
 * 3me = Données immédiates (Push)
 * 4me = Réinitialisation (Reset)
 * 5me = Synchronisation (SYN)
 * 6me = Fin
Options = suites d'option codées sur
 * 1 octet à 00 = Fin des options
 * 1 octet à 01 = NOP (pas d'opération)
 * plusieurs octets de type TLV
   T = un octet de type:
     2 Négociation de la taille max. du segment
     3 Adaptation de la taille de la fenêtre
     4 Autorisation des acquittements sélectifs
     8 Estampilles temporelles
   L = un octet pour la taille totale de l'option
   V = valeur de l'option (sur L-2 octets)
```

Structure de datagramme UDP

```
<-----------------32bits----------------------->
+----------------------+----------------------+
| Port Source          | Port Destination     |
+----------------------+----------------------+
| Longueur UDP         | Somme de ctrl (message)
+----------------------+----------------------+
...      Données
+---------------------------------------------+
```

Services associés aux ports

ftp-data	20/tcp		
ftp	21/tcp		
ssh	22/tcp	ssh	22/udp
telnet	23/tcp		
smtp	25/tcp		
domain	53/tcp	domain	53/udp
		tftp	69/udp
finger	79/tcp		
www	80/tcp	www	80/udp
kerberos	88/tcp	kerberos	88/udp
pop-3	110/tcp	pop-3	110/udp
		snmp	161/udp
		snmp-trap	162/udp

U.E. ARES - Travaux sur Machine Encadrés n°2

Couche application (1) : Telnet, rlogin, ssh, ftp, tftp, scp et http

1 Connexions à distance

1.1 Protocole TELNET

Cette première analyse a pour but de percevoir la nature du trafic TELNET. A l'aide du logiciel `ethereal`, chargez et analysez la trace suivante : `/Infos/lmd/2008/master/ue/ares-2008oct/tme2-tel.dmp.gz`

1. Rappelez le fonctionnement du protocole TELNET (en particulier, décrivez les deux phases vues en cours).

1.1.1 Négociations TELNET

Les négociations ont principalement lieu au début de la connexion. Elles sont composées de commandes qui peuvent être envoyées dans chaque sens. Chaque commande démarre par la **commande d'échappement** qui est codée sur 1 octet (début d'une commande) : `IAC`=255. Les **commandes de négociation** d'option (sur 1 octet) sont immédiatement suivies de la valeur de l'option (sur 1 octet) : `WILL`=251 (indique ce qu'une entitée va faire), `WONT`=252 (ne pas faire), `DO`=253 (demande à l'autre entitée de faire), `DON'T`=254 (demande de ne pas faire).
Exemple :

`IAC`, `DO`, `24`

Les **sous-options** sont transmises (après une demande à l'aide d'un `WILL` puis d'une confirmation avec un `DO`) entre les deux commandes (sur 1 octet) suivantes : `SB`=250 et `SE`=240. Une sous-option se compose du code de l'option sur 1 octet, 1 octet nul puis la valeur de l'option.
Exemple :

`IAC`, `SB`, `24`, `0`, `'V'`, `'T'`, `'2'`, `'2'`, `'0'`, `IAC`, `SE`

$(valeur)_{10}$	**Nom de l'option**
1	Echo
3	Suppress Go Ahead
5	Status
6	Timing Mark
24	Terminal Type
31	Negotiate About Window Size
32	Terminal Speed
33	Remote Flow Control
34	Linemode
35	X Display Location
36	Environment variables
39	New Environment Option

Dans l'échange étudié, essayez en ne regardant que la partie présentant la capture des octets de trouver la signification des différents paramètres négociés.

1. Analysez les différentes options et sous-options échangées.
2. Trouvez combien de temps dure la négociation.

1.1.2 Échange de données TELNET

Passez les quelques trames de négociation.

1. Quelles informations sont véhiculées dans la suite de la communication ?
2. Que pensez-vous de l'efficacité du protocole ?
3. Quel est le degré d'interactivité ?

1.1.3 TELNET longue distance (*facultatif... à traiter si vous êtes nettement en avance par rapport au reste du groupe*)

A partir d'une trace contenant une heure de trafic longue distance entre le *Lawrence Berkeley Laboratory* et le reste du monde en janvier 1994, retrouvez des exemples de communications TELNET.

Ces traces [1], initialement au format `tcpdump` (le même que celui par défaut d'`ethereal`), ont été converties en ASCII en prenant soin de renuméroter les adresses IP et de supprimer le contenu des paquet.

En voici un extrait :

```
8.430376 22 21 23 33281 1
8.437539 3 4 3930 119 47
8.442644 4 3 119 3930 15
8.454895 26 11 4890 23 1
8.459398 5 2 14037 23 0
8.469004 4 23 4464 119 512
```

La première colonne contient une estampille temporelle relative au début de la capture (exprimée en secondes), les deux colonnes suivantes sont les adresses sources et destinations renumérotées par ordre d'apparition, ensuite se trouvent les numéros de port puis la taille des données (en octets).

Chargez la trace `/Infos/lmd/2008/master/ue/ares-2008oct/tme2-lbl.txt.gz` dans un répertoire **local** (ex : `/tmp`)[2].

A l'aide des outils UNIX standard (`awk`, `perl`, `sed`...), isolez un des flots TELNET et identifiez ses caractéristiques typiques. **Ne demandez pas à votre encadrant d'aide sur ces outils, il est là pour répondre à vos questions liées au réseau**.

1. Réalisez un chronogramme rapide des échanges réalisés. Que pouvez vous dire de l'interactivité ?
2. Pouvez vous deviner le type d'information échangée ?

1.2 Protocole RLOGIN

Cette seconde analyse a pour but de percevoir la nature du trafic RLOGIN. A l'aide du logiciel `ethereal`, chargez et analysez la trace suivante : `/Infos/lmd/2008/master/ue/ares-2008oct/tme2-rlo.dmp.gz`

1. Rappelez le fonctionnement du protocole RLOGIN.
2. Quelles informations sont véhiculées à travers le premier échange ?
3. Quelles différences constatez vous avec TELNET (l'échange réalisé est identique au précédent au niveau des données utilisateur échangées) ?
4. Toujours par rapport à la trace `tme2-lbl.txt.gz` chargée précédement, identifiez des communications RLOGIN (*facultatif*).

1.3 Protocole SSH

Cette troisième analyse a pour but de percevoir la nature du trafic SSH. A l'aide du logiciel `ethereal`, chargez et analysez la trace suivante : `/Infos/lmd/2008/master/ue/ares-2008oct/tme2-ssh.dmp.gz`

1. Rappelez le fonctionnement du protocole SSH.
2. Quelles différences constatez vous avec TELNET et SSH (l'échange réalisé est identique aux précédents en terme de données utilisateur) ?
3. Par rapport à la trace `tme2-lbl.txt.gz` chargée précédement, identifiez des communications SSH (*facultatif*).

2 Transfert de fichiers

2.1 Protocole FTP

Cette analyse a pour but de percevoir la nature du trafic FTP. A l'aide du logiciel `ethereal`, chargez et analysez la trace suivante : `/Infos/lmd/2008/master/ue/ares-2008oct/tme2-ftp.dmp.gz`

1. Rappelez le fonctionnement du protocole FTP (en particulier, expliquez le système de double connexion).

[1] The trace lbl-pkt-4 ran from 14 :00 to 15 :00 on Friday, January 21, 1994 (times are Pacific Standard Time) and captured 1.3 million TCP packets, the dropping about 0.0007 of the total. The tracing was done on the Ethernet DMZ network over which flows all traffic into or out of the Lawrence Berkeley Laboratory, located in Berkeley, California. The raw trace was made using tcpdump on a Sun Sparcstation using the BPF kernel packet filter. Timestamps have microsecond precision. The trace has been "sanitized" using the sanitize scripts. This means that the host IP addresses have been renumbered, and all packet contents removed. The trace was made by Vern Paxson (vern@ee.lbl.gov). The trace may be freely redistributed.

[2] La taille de la trace étant particulièrement importante, si vous travaillez sur votre compte qui est monté par NFS vous obtiendrez des temps de réponse très mauvais.

2.1.1 Analyse de la connexion FTP de contrôle

La trace correspond à l'échange suivant sur le terminal de l'utilisateur utilisant le client `ftp` standard d'UNIX.

```
test-res@galion:~$ ftp pirogue.l2ti.univ-paris13.fr
Connected to pirogue.l2ti.univ-paris13.fr.
220 ProFTPD 1.2.5rc1 Server (Debian) [pirogue.l2ti.univ-paris13.fr]
Name: test-res
331 Password required for test-res.
Password:
530 Login incorrect.
Login failed.
ftp> user test-res
331 Password required for test-res.
Password:
230 User test-res logged in.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> cd /etc
250 CWD command successful.
ftp> dir sh*
200 PORT command successful.
150 Opening ASCII mode data connection for file list.
-rw-r-----   1 root     shadow        938 Oct  9 11:38 shadow
-rw-------   1 root     root          896 Dec 15  2003 shadow-
-rw-r-----   1 root     shadow        832 Jul 15  2002 shadow.org
-rw-r--r--   1 root     root          185 Apr  7  2002 shells
226-Transfer complete.
226 Quotas off
ftp> get shell
local: shell remote: shell
200 PORT command successful.
550 shell: No such file or directory
ftp> get shells
local: shells remote: shells
200 PORT command successful.
150 Opening BINARY mode data connection for shells (185 bytes).
226 Transfer complete.
185 bytes received in 0.00 secs (62.9 kB/s)
ftp> quit
221 Goodbye.
test-res@galion:~$
```

Retrouvez la correspondance entre les messages échangés sur le réseau et ceux affichés par l'application.

1. Est-ce le serveur qui initie la communication ?
2. Quelle commande identifie l'utilisateur ? Dans cet exemple que pouvez vous dire sur l'utilisateur ?
3. Quelle est la commande qui authentifie ensuite l'utilisateur ? Le mot de passe apparait-il en clair sur le réseau ?
4. Quel est l'intérêt de la commande suivant l'authentification ?
5. A quoi sert la commande `PORT` ? Analysez ses paramètres. Pourquoi est-elle émise à ce point de l'échange ?
6. La commande `LIST` permet d'obtenir la liste des fichiers du répertoire courant au niveau du serveur. Pourquoi est-elle suivie de deux messages envoyés par le serveur ?
7. Quelles sont les autres commandes que vous observez ? A quoi servent-elles ?
8. A quel moment de la transaction ont lieu les transferts de fichiers ?

2.1.2 Analyse de la connexion FTP de données

1. A quoi correspondent les données échangées sur les connexions de tranfert de données ?
2. Sur quels ports ces données sont-elles envoyées ?

2.1.3 FTP longue distance (*facultatif*)

1. Par rapport à la trace `tme2-lbl.txt.gz` chargée précédement, identifiez des communications FTP avec les connexions `FTP` et `FTP-DATA` associées.
2. Tracez le chronogramme.
3. Que pouvez vous dire de l'interactivité par rapport à TELNET ?

2.2 Protocole SCP

Cette analyse a pour but de percevoir la nature du trafic SCP. A l'aide du logiciel `ethereal`, chargez et analysez la trace suivante : `/Infos/lmd/2008/master/ue/ares-2008oct/tme2-scp.dmp.gz`

La trace correspond à l'échange suivant sur le terminal de l'utilisateur :

```
test-res@galion:~$ scp pirogue.l2ti.univ-paris13.fr:/etc/shells .
test-res@pirogue.l2ti.univ-paris13.fr's password:
Permission denied, please try again.
test-res@pirogue.l2ti.univ-paris13.fr's password:
shells                100% |****************************|   185       00:00
test-res@galion:~$
```

1. Rappelez le fonctionnement du protocole SCP.
2. Pouvez-vous faire la correspondance entre les messages du terminal et les trames echangées ?
3. Quelles différences constatez-vous avec FTP ?

2.3 Protocole TFTP

L'analyse suivante a pour but de percevoir la nature du trafic TFTP. A l'aide du logiciel `ethereal`, chargez et analysez la trace suivante : `/Infos/lmd/2008/master/ue/ares-2008oct/tme2-tft.dmp.gz`

La trace correspond à l'échange suivant sur le terminal de l'utilisateur :

```
test-res@galere:~$ tftp pirogue.l2ti.univ-paris13.fr
tftp> get unixbott
Error code 1: File not found
tftp> get unixboot
Received 13240 bytes in 0.0 seconds
tftp> quit
test-res@galere:~$
```

1. Rappelez le fonctionnement du protocole TFTP.
2. Pouvez-vous faire la correspondance entre les messages du terminal et les trames echangées ?
3. Quelles différences constatez-vous avec FTP ?

3 Trafic web

3.1 Protocole HTTP

3.1.1 Analyse HTTP

La dernière analyse a pour but de percevoir la nature du trafic HTTP. A l'aide du logiciel `ethereal`, chargez et analysez la trace suivante : `/Infos/lmd/2008/master/ue/ares-2008oct/tme2-htt.dmp.gz`

1. Rappelez le fonctionnement du protocole HTTP.
2. Pouvez-vous reconstituer le parcours effectué a travers les pages webs ?
3. Quelles optimisations sont mises en œuvre pour accélérer le rapatriement des pages web ?

3.1.2 HTTP longue distance (*facultatif*)

1. S'il vous reste du temps, par rapport à la trace `tme2-lbl.txt.gz` chargée précédement, identifiez les trafic HTTP.
2. Tracez le chronogramme.
3. Que pensez-vous de l'interactivité ?

U.E. ARES - Travaux sur Machine Encadrés n°3

Couche application (2) : DNS, SNMP, SMTP, POP et IMAP...

Après avoir travaillé sur plusieurs protocoles applicatifs reposant sur TCP lors de la séance précédente, nous débuterons cette séance par l'étude de deux protocoles applicatifs reposant sur un autre protocole de la couche transport : UDP.

1 Annuaire

1.1 Requête DNS

Voici une trame observée sur le réseau :

```
0000  00 07 e9 0c 90 62 00 20  ed 87 fd e6 08 00 45 00   .....b.  ......E.
0010  00 39 00 00 40 00 40 11  a9 71 84 e3 3d 7a 84 e3   .9..@.@. .q..=z..
0020  4a 02 85 05 00 35 00 25  c0 74 a0 71 01 00 00 01   J....5.% .t.q....
0030  00 00 00 00 00 00 03 77  77 77 04 6c 69 70 36 02   .......w ww.lip6.
0040  66 72 00 00 01 00 01                              fr.....
```

1. Analysez manuellement la trame ci-dessus à l'aide du support de cours.
2. Quel est le but du message contenu dans cette trame ?
3. Quelle action de l'utilisateur a pu déclencher cette requête ?

1.2 Réponse DNS

Peu de temps après, on peut observer cette trame sur le réseau :

```
0000  00 20 ed 87 fd e6 00 07  e9 0c 90 62 08 00 45 00   . ...... ...b..E.
0010  00 cf 2a 2d 00 00 3f 11  bf ae 84 e3 4a 02 84 e3   ..*-..?. ....J...
0020  3d 7a 00 35 85 05 00 bb  a1 3b a0 71 85 80 00 01   =z.5.... .;.q....
0030  00 02 00 03 00 03 03 77  77 77 04 6c 69 70 36 02   .......w ww.lip6.
0040  66 72 00 00 01 00 01 c0  0c 00 05 00 01 00 00 54   fr...... .......T
0050  60 00 08 05 68 6f 72 75  73 c0 10 c0 29 00 01 00   `...horu s...)...
0060  01 00 00 54 60 00 04 84  e3 3c 0d c0 10 00 02 00   ...T`... .<......
0070  01 00 00 54 60 00 07 04  69 73 69 73 c0 10 c0 10   ...T`... isis....
0080  00 02 00 01 00 00 54 60  00 09 06 6f 73 69 72 69   ......T` ...osiri
0090  73 c0 10 c0 10 00 02 00  01 00 00 54 60 00 0e 06   s....... ...T`...
00a0  73 6f 6c 65 69 6c 04 75  76 73 71 c0 15 c0 4d 00   soleil.u vsq...M.
00b0  01 00 01 00 00 54 60 00  04 84 e3 3c 02 c0 60 00   .....T`. ...<..`.
00c0  01 00 01 00 00 54 60 00  04 84 e3 3c 1e c0 75 00   .....T`. ...<..u.
00d0  01 00 01 00 01 16 cb 00  04 c1 33 18 01            ........ ..3..
```

1. Analysez manuellement la trame ci-dessus[1].
2. Quelles informations sont renvoyées par le serveur DNS local ?
3. Les informations récupérées correspondent-elles à celles attendues par le client ?

[1] Attention au codage des noms avec renvoi (code `0xC0`+n indiquant sur un octet la distance n en octet du début du message DNS).

Vérifiez les résultats de l'exercice précédent à l'aide du logiciel `ethereal`. Chargez la trace suivante : `/Infos/lmd/2008/master/ue/ares-2008oct/tme3-dn1.dmp.gz`

1.3 Second échange DNS

Continuez à utiliser le logiciel `ethereal` pour la suite. Chargez cette seconde trace DNS : `/Infos/lmd/2008/master/ue/ares-2008oct/tme3-dn2.dmp.gz`

1. Analysez rapidement les deux trames contenues dans la trace.
2. Expliquez le but de cet échange.
3. A votre avis, pourquoi la résolution de nom `www.apple.com` est-elle renvoyée vers des serveurs du domaine `aka*.net` ?

2 Administration

Des messages observés à proximité d'une station d'administration sont présentés dans la trace : `/Infos/lmd/2008/master/ue/ares-2008oct/tme3-snm.dmp.gz`

2.1 Requête SNMP

1. Analysez la première trame.
2. Quel est le but de cette requête ?
3. Qui a généré ce message ?

2.2 Réponse SNMP

1. Suite à l'émission de la trame précédente, une seconde trame est émise. Analysez cette dernière.
2. Quel est le type d'équipement qui a été impliqué ?
3. A la vue de cet échange, que pensez-vous de la sécurité associée à SNMP ?

2.3 Deuxième échange SNMP

1. Après ce premier échange, analysez les deux trames échangées ensuite.
2. Quelle nouvelle opération est réalisée dans cet échange ? Quelles possibilités offre ce type de requête ?

3 Messagerie

Nous allons détailler les différents protocoles associés à l'émission et à la réception d'un message électronique.

3.1 Emission du message

3.1.1 Protocole SMTP

Chargez et analysez la trace : `/Infos/lmd/2008/master/ue/ares-2008oct/tme3-smt.dmp.gz`

1. Quelles sont les commandes utilisées par le protocole SMTP lors de l'émission d'un courrier ? Pouvez-vous indiquer leur utilité et le type de réponses produit ?
2. Quelles sont les contraintes imposées à la forme du courrier ? Expliquez la structure de ce dernier et détaillez les champs qui compose son entête.
3. Que pensez-vous des possibilités d'identification du protocole SMTP ?

3.1.2 Protocole HTTP

Chargez et analysez la trace : `/Infos/lmd/2008/master/ue/ares-2008oct/tme3-wm1.dmp.gz`

1. Quel est le but de l'échange présenté dans cette trace ?
2. Pouvez-vous retrouver le message original dans la réponse du serveur ?
3. Que pensez-vous de la confidentialité lorsque vous consultez votre courrier de cette manière ?

3.2 Réception du message

3.2.1 Protocole POP

Chargez et analysez la trace : `/Infos/lmd/2008/master/ue/ares-2008oct/tme3-pop.dmp.gz`

1. Quelles sont les commandes utilisées par le protocole POP lors de la récupération d'un courrier ? Pouvez-vous indiquer leur utilité et le type de réponse produit ?
2. A votre avis, quelles seraient les réponses du serveur POP s'il y avait plusieurs messages en attente ?
3. Quelles sont les différences entre le message envoyé précédement et celui reçu ici ?

3.2.2 Protocole IMAP

Chargez et analysez la trace : `/Infos/lmd/2008/master/ue/ares-2008oct/tme3-ima.dmp.gz`

1. Quels types d'échanges sont réalisés entre le client et le serveur IMAP ?
2. Quelles différences protocolaires observez-vous entre POP et IMAP ?
3. Quelles sont les différences entre le message envoyé précédement et celui reçu ici ?
4. Pensez vous que l'authentification soit plus sécurisée avec IMAP ?

3.2.3 Protocole HTTP

Chargez et analysez la trace : `/Infos/lmd/2008/master/ue/ares-2008oct/tme3-wm2.dmp.gz`

1. Quel est le but de l'échange présenté dans cette dernière trace ?
2. Pouvez-vous retrouver le message original dans la réponse du serveur ?
3. La consultation du message retourne beaucoup de trafic HTTP. Discutez des performances d'une consultation à travers le web.

3.3 Analyse longue distance (*facultatif*)

A partir de la trace contenant une heure de trafic longue distance entre le *Lawrence Berkeley Laboratory* et le reste du monde en janvier 1994, vous pouvez retrouver des exemples de communications SNMP, POP et IMAP. Pour cela, chargez la trace étudiée lors de la dernière séance (`/Infos/lmd/2008/master/ue/ares-2008oct/tme2-lbl.txt.gz`) dans un répertoire local (`/tmp`)[2]. A l'aide des outils UNIX standard (`awk`, `perl`, `sed`...), isolez un des flots intéressants et analysez-en les caractéristiques.

[2]La taille de la trace étant particulièrement importante, si vous travaillez sur votre compte qui est monté par NFS vous obtiendrez des temps de réponse très mauvais.

U.E. ARES - Travaux sur Machine Encadrés n°4

Transport: Mécanismes de TCP

1 Rappel des fonctionalités TCP

Cette première analyse a pour but d'observer les différents mécanismes protocolaires de TCP. Pour cela, nous nous appuierons sur la capture d'une connexion TCP encapsulant un échange HTTP.

1.1 Mise en place de la connexion HTTP

Voici la première trame échangée lors de l'ouverture de la connexion :

```
0000  00 50 7f 05 7d 40 00 10  a4 86 2d 0b 08 00 45 00   .P..}@.. ..-...E.
0010  00 3c 17 96 40 00 40 06  6d f3 0a 21 b6 b2 c0 37   .<..@.@. m..!...7
0020  34 28 84 b3 00 50 b6 94  b0 b7 00 00 00 00 a0 02   4(...P.. ........
0030  16 d0 e8 23 00 00 02 04  05 b4 04 02 08 0a 00 6f   ...#.... .......o
0040  a7 21 00 00 00 00 01 03  03 00                     .!...... ..
```

1. Pour vous échauffer, analysez **manuellement** (sans `ethereal`) la trame présentée ci-dessus.
2. Utilisez à présent (et seulement après avoir analysé trame précédente) le logiciel `ethereal` pour la suite. Chargez le fichier suivant : `/Infos/lmd/2008/master/ue/ares-2008oct/tme4-htt.dmp.gz`. Identifiez les hôtes impliqués dans l'échange HTTP. Quels vont être leurs rôles respectifs dans l'échange présenté ?
3. Dans les premières trames, quels sont les bits de contrôle (TCP *flags*) positionnés ? Que signifient-ils ?
4. Quelles informations peut-on déduire des ports contenus dans les segments ci-dessus ?
5. Rappelez le fonctionnement des numéros de séquences de TCP. Justifiez les valeurs présentes dans ces segments.
6. Que pouvez-vous dire du contrôle de flux pour les segments étudiés ?
7. Pouvez-vous trouvez des options dans les entêtes TCP ? Si oui, que signifient-elles ?

1.2 Suite de l'échange HTTP

Nous vous proposons de continuer l'analyse de cet échange...

Voici la représentation partiellement décodée de chaques trames grâce à l'outils UNIX `tcpdump` (basé sur la même bibliothèque de capture, `libpcap`, mais plus adapté pour une présentation textuelle) :

```
10:12:21.406418 galion.33971 > 192.55.52.40.www: S 3063197879:3063197879(0) win 5840
[A]                                              <mss 1460,sackOK,timestamp 7317281 0,nop,wscale 0> (DF)

10:12:21.576976 192.55.52.40.www > galion.33971: S 610765288:610765288(0) ack 3063197880 win 64240
[B]                                              <mss 1402,nop,wscale 0,nop,nop,timestamp 0 0,nop,nop,sackOK> (DF)

10:12:21.577036 galion.33971 > 192.55.52.40.www: . ack 1 win 5840
[C]                                              <nop,nop,timestamp 7317298 0> (DF)

10:12:21.577237 galion.33971 > 192.55.52.40.www: P 1:486(485) ack 1 win 5840
[D]                                              <nop,nop,timestamp 7317298 0> (DF)

10:12:21.776923 192.55.52.40.www > galion.33971: . 1:1391(1390) ack 486 win 63755
[E]                                              <nop,nop,timestamp 19332362 7317298> (DF)

10:12:21.776978 galion.33971 > 192.55.52.40.www: . ack 1391 win 8340
[F]                                              <nop,nop,timestamp 7317318 19332362> (DF)

...
```

1. Tracez rapidement le chronogramme correspondant à cet échange.
2. Nous souhaitons étudier la demi-connexion correspondant à l'émission de données du serveur (`192.55.52.40.www`) vers le client (`10.33.182.178.33971`). Complétez le tableau suivant (numéros de séquence relatifs) :

action	base de la fenêtre	pointeur d'émission	fin de la fenêtre	taille de la fenêtre	*commentaire*
réception `A`	—	—	—	5840	*win 5840*
émission `B`	0 (610765288)	1	5840	—	*SYN, +1 pas d'émission*
réception `C`	1	1	5841	5840	*ACK*
réception `D`	1	1	5841	5840	*ACK*
émission `E`	1	1391	5841	—	
réception `F`	1391	1391	9731	8340	*ACK, win 8340*
...					

3. Commentez l'évolution des numéros de séquence.
4. Que pouvez-vous dire sur la gestion des tampons ?
5. Observez-vous de nouvelles options ? Pouvez-vous les expliquer ?
6. Que pouvez-vous dire à propos de la génération des acquittements par le récepteur ?
7. Comment se termine la communication ? Détaillez les échanges finaux.

2 Echanges TCP imbriqués *(facultatif)*

Utilisez `ethereal` pour chargez le fichier suivant : `/Infos/lmd/2008/master/ue/ares-2008oct/tme4-ftp.dmp.gz`.

1. Observez l'échange capturé et expliquez les actions réalisées au niveau applicatif.
2. Tracez le chronogramme correspondant à ces échanges en utilisant une couleur différente par connexion.
3. Que pouvez-vous dire de l'utilisation du *flag* `PUSH` ?

U.E. ARES - Travaux sur Machine Encadrés n°5

Transport : Contrôle de congestion TCP

Voici cinq captures de trafic HTTP réalisées avec une sonde proche du client. Pour chacune d'elles, tracez le chronogramme et étudiez les mécanismes de contrôle de congestion mise-en-œuvre. Discutez en particulier des points suivants :

- Quel est le RTT moyen ?
- Reconnaissez-vous les mécanismes de contrôle de congestion ?
- Jusqu'a combien de segments sont transmis par RTT ?
- Quel est le débit moyen alors atteint ?
- Un envoi continu apparait il ?
- Des perturbations sont-elles présentes (déséquencement, retransmission...) ?

1 Trafic HTTP Paris-Brisbane (WAN Intercontinental : 17000km)

Utilisez le logiciel `ethereal` pour cette première trace. Chargez le fichier :
`/Infos/lmd/2008/master/ue/ares-2008oct/tme5-wau.dmp.gz`

2 Trafic HTTP Paris-Pekin (WAN Intercontinental : 8300km) *(facultatif)*

Utilisez le logiciel `ethereal` pour cette deuxième trace. Chargez le fichier suivant :
`/Infos/lmd/2008/master/ue/ares-2008oct/tme5-wcn.dmp.gz`

3 Trafic HTTP Paris-Budapest (WAN Continental : 1200km)

Utilisez le logiciel `ethereal` pour cette troisième trace. Chargez le fichier suivant :
`/Infos/lmd/2008/master/ue/ares-2008oct/tme5-whu.dmp.gz`

4 Trafic HTTP Paris-Evry (MAN : 36km) *(facultatif)*

Utilisez le logiciel `ethereal` pour cette quatrième trace. Chargez le fichier ci-dessous :
`/Infos/lmd/2008/master/ue/ares-2008oct/tme5-man.dmp.gz`

5 Trafic HTTP local (LAN)

Utilisez encore une fois le logiciel `ethereal` pour cette dernière trace. Chargez le fichier se trouvant à la référence :
`/Infos/lmd/2008/master/ue/ares-2008oct/tme5-lan.dmp.gz`

U.E. ARES - Travaux sur Machine Encadrés n°6

Couche réseau : IP, ICMP...

Nous nous intéressons à la couche réseau dans l'environnement TCP/IP. Lors des scéances précédentes, nous avons déjà observé de nombreux de paquets IP dans les traces analysées. Nous nous attacherons donc à observer du trafic spécifique à la couche 3 à travers deux outils fondamentaux pour la supervision des réseaux TCP/IP : `ping` et `traceroute`.

1 Etude de l'outil `ping`

Voici le début de la description du `man` UNIX sur la commande `ping` :

> `Ping uses the ICMP protocol's mandatory ECHO_REQUEST datagram to elicit an ICMP ECHO_RESPONSE from a host or gateway. ECHO_REQUEST datagrams (''pings'') have an IP and ICMP header, followed by a ''struct timeval'' and then an arbitrary number of ''pad'' bytes used to fill out the packet.`

1.1 Test d'une machine distante

La commande `ping` permet de tester la connectivité vers une machine distante, et — en envoyant plusieurs paquets à la suite — d'effectuer des statistiques sur les caractéristiques du chemin suivi (RTT, taux de perte, variabilité des résultats en fonction de la taille des datagrammes émis...). Voici un exemple d'utilisation :

```
pirogue:~# ping sphinx.lip6.fr
PING sphinx.lip6.fr (132.227.74.253): 56 data bytes
64 bytes from 132.227.74.253: icmp_seq=0 ttl=247 time=5.5 ms
64 bytes from 132.227.74.253: icmp_seq=1 ttl=247 time=9.3 ms
64 bytes from 132.227.74.253: icmp_seq=2 ttl=247 time=8.0 ms
64 bytes from 132.227.74.253: icmp_seq=3 ttl=247 time=6.3 ms
64 bytes from 132.227.74.253: icmp_seq=4 ttl=247 time=4.8 ms
64 bytes from 132.227.74.253: icmp_seq=5 ttl=247 time=7.6 ms
64 bytes from 132.227.74.253: icmp_seq=6 ttl=247 time=5.8 ms

--- sphinx.lip6.fr ping statistics ---
7 packets transmitted, 7 packets received, 0% packet loss
round-trip min/avg/max = 4.8/6.7/9.3 ms
pirogue:~#
```

1. Pour vous échauffer, analysez **manuellement** (sans `ethereal`) la première trame correspondant à cet échange :

```
0000  00 08 21 59 66 42 00 04  76 21 1b 95 08 00 45 00   ..!YfB.. v!....E.
0010  00 54 64 db 00 00 40 01  df 38 c2 fe a3 b6 84 e3   .TdÛ..@. ß8ÂP£¶.ã
0020  4a fd 08 00 d8 2d 6e 5b  00 00 3f 9f 10 01 00 0d   Jý..Ø-n[ ..?.....
0030  76 c6 08 09 0a 0b 0c 0d  0e 0f 10 11 12 13 14 15   vÆ...... ........
0040  16 17 18 19 1a 1b 1c 1d  1e 1f 20 21 22 23 24 25   ........ .. !"#$%
0050  26 27 28 29 2a 2b 2c 2d  2e 2f 30 31 32 33 34 35   &'()*+,- ./012345
```

```
0060  36 37                                                  67
```

2. A votre avis, a quoi correspondent les données transportées par le messages ICMP analysé précédement ? Utilisez le `man` UNIX pour avoir des informations complémentaires sur `ping`.
3. Utilisez à présent le logiciel `ethereal` pour la suite.
   Chargez le fichier suivant : `/Infos/lmd/2008/master/ue/ares-2008oct/tme6-pin.dmp.gz`.
   Quel échange observez vous dans cette trace ?
4. Quelle réponse est retournée à la requête analysée précédement ?
5. Analysez les champs ICMP de plusieurs paquets échangées. Développez leur signification et inférez leur utilité pour la commande `ping`.

1.2 Test d'une machine proche avec enregistrement du chemin

La commande `ping -R` est une variante de l'exécution illustrée précédement où des options de l'entête IP sont introduites. A vous de les découvrir dans l'exemple suivant :

```
pirogue:~# ping -R rap-jussieu.cssi.renater.fr
PING rap-jussieu.cssi.renater.fr (193.51.182.201): 56 data bytes
64 bytes from 193.51.182.201: icmp_seq=0 ttl=252 time=11.2 ms
RR:     pirogue.l2ti.univ-paris13.fr (194.254.163.182)
        192.168.208.252
        paris13-jussieu.cssi.renater.fr (193.51.182.205)
        jussieu-a1-0-65.cssi.renater.fr (193.51.182.202)
        rap-jussieu.cssi.renater.fr (193.51.182.201)
        jussieu-f3-3.cssi.renater.fr (193.51.182.206)
        192.168.208.254
        gw163.univ-paris13.fr (194.254.163.254)
        pirogue.l2ti.univ-paris13.fr (194.254.163.182)

--- rap-jussieu.cssi.renater.fr ping statistics ---
5 packets transmitted, 1 packets received, 80% packet loss
round-trip min/avg/max = 11.2/11.2/11.2 ms
pirogue:~#
```

1. Chargez le fichier suivant : `/Infos/lmd/2008/master/ue/ares-2008oct/tme6-pRp.dmp.gz`.
   Quel informations nouvelles découvrez-vous dans ces trames ?
2. Analysez les champs ICMP. Développez leur signification et inférez, avec le contenu de l'option IP, leur utilité pour la commande `ping -R`.
3. Pouvez-vous réaliser un schéma des réseaux traversés ?

1.3 Test d'une machine éloignée avec enregistrement du chemin

La commande `ping -R` utilise le champ d'option limité de l'entête IP. Que se passe-t'il dans l'exemple suivant ?

```
pirogue:~# ping -R sphinx.lip6.fr
PING sphinx.lip6.fr (132.227.74.253): 56 data bytes
64 bytes from 132.227.74.253: icmp_seq=0 ttl=247 time=8.8 ms
RR:     pirogue.l2ti.univ-paris13.fr (194.254.163.182)
        192.168.208.252
        paris13-jussieu.cssi.renater.fr (193.51.182.205)
        jussieu-a1-0-65.cssi.renater.fr (193.51.182.202)
        gw-rap.rap.prd.fr (195.221.126.78)
        rap-jussieu.rap.prd.fr (195.221.127.181)
        r-jusren.reseau.jussieu.fr (134.157.254.126)
        r-olympe.lip6.fr (132.227.109.254)
        132.227.74.254
```

```
64 bytes from 132.227.74.253: icmp_seq=1 ttl=247 time=3099.5 ms (same route)
64 bytes from 132.227.74.253: icmp_seq=2 ttl=247 time=2099.7 ms (same route)
64 bytes from 132.227.74.253: icmp_seq=3 ttl=247 time=1099.9 ms (same route)
64 bytes from 132.227.74.253: icmp_seq=4 ttl=247 time=100.1 ms  (same route)
64 bytes from 132.227.74.253: icmp_seq=5 ttl=247 time=8.6 ms    (same route)
64 bytes from 132.227.74.253: icmp_seq=6 ttl=247 time=14.0 ms   (same route)

--- sphinx.lip6.fr ping statistics ---
7 packets transmitted, 7 packets received, 0% packet loss
round-trip min/avg/max = 8.6/918.6/3099.5 ms
pirogue:~#
```

1. Chargez le fichier suivant : `/Infos/lmd/2008/master/ue/ares-2008oct/tme6-pRl.dmp.gz`.
   Quel informations nouvelles découvrez-vous dans ces trames ?
2. Analysez la route enregistrée dans les paquets IP de retour. Quelles différences observez-vous ?
3. Quelle limitations constatez-vous pour l'approche `ping -R` pour déduire la route suivie par les paquets vers un destinataire donné ?

2 Etude de l'outil `traceroute`

Voici le début de la description du `man` UNIX sur la commande `traceroute` :

> `The Internet is a large and complex aggregation of network hardware, connected together by gateways. Tracking the route one's packets follow (or finding the miscreant gateway that's discarding your packets) can be difficult. Traceroute utilizes the IP protocol 'time to live' field and attempts to elicit an ICMP TIME_EXCEEDED response from each gateway along the path to some host.`

La commande `traceroute` permet de récupérer les adresses des interfaces des machines intermédaires vers une machine distante. Voici un exemple d'utilisation :

```
pirogue:~# traceroute sphinx.lip6.fr
traceroute to sphinx.lip6.fr (132.227.74.253), 30 hops max, 38 byte packets
 1  gw163.univ-paris13.fr (194.254.163.254)  2.579 ms  0.496 ms  0.415 ms
 2  192.168.208.254 (192.168.208.254)  0.640 ms  0.602 ms  0.216 ms
 3  jussieu-f3-3.cssi.renater.fr (193.51.182.206)  1.813 ms  2.100 ms  1.778 ms
 4  rap-jussieu.cssi.renater.fr (193.51.182.201)  1.986 ms  2.167 ms  2.216 ms
 5  cr-jussieu.rap.prd.fr (195.221.126.77)  2.867 ms  2.799 ms  2.642 ms
 6  jussieu-rap.rap.prd.fr (195.221.127.182)  3.021 ms  3.797 ms  2.315 ms
 7  r-scott.reseau.jussieu.fr (134.157.254.10)  3.692 ms  4.402 ms  5.438 ms
 8  olympe-gw.lip6.fr (132.227.109.1)  4.881 ms !N  3.932 ms !N  3.917 ms !N
pirogue:~#
```

1. Chargez le fichier suivant : `/Infos/lmd/2008/master/ue/ares-2008oct/tme6-tcr.dmp.gz`.
   Analysez la première trame `traceroute` envoyée. Quel est son but ?
2. Quel événement génère la trame suivante ? Quel intérêt peut-on retirer de ce phénomène ?
3. A votre avis, de quelle manière la commande `traceroute` génère-t-elle ses réponses ?
4. Pourquoi les numéros de port UDP destination évoluent-ils ?
5. Finalement, de quelle manière la commande `traceroute` termine-t-elle ?

U.E. ARES - Travaux sur Machine Encadrés n°7

Routage + (VLAN)

Nous nous intéressons toujours à la couche réseau et, en particulier, aux protocoles de **routage** dans l'environnement TCP/IP. Lors des séances précédentes, nous avons observé de nombreux paquets IP dans les traces analysées. Ici, nous nous attacherons à étudier du trafic généralement échangé entre routeurs pour les protocoles : RIP, OSPF et BGP.

1 Protocole de routage interne : RIP

1. Pour vous échauffer, analysez **manuellement** (sans `ethereal`) cette première trame. Détaillez les différentes encapsulations présentes :

```
0000  01 00 5e 00 00 09 00 08  c7 a4 d4 be 08 00 45 00   ..^..... ......E.
0010  00 98 73 2f 00 00 01 11  74 38 84 e3 6d 01 e0 00   ..s/.... t8..m...
0020  00 09 02 08 02 08 00 84  29 77 02 02 00 00 00 02   ........ )w......
0030  00 00 84 e3 3d 00 ff ff  ff 00 00 00 00 00 00 00   ....=... ........
0040  00 01 00 02 00 00 84 e3  3e 00 ff ff ff 00 00 00   ........ >.......
0050  00 00 00 00 00 01 00 02  00 00 84 e3 48 00 ff ff   ........ ....H...
0060  ff 00 00 00 00 00 00 00  00 01 00 02 00 00 84 e3   ........ ........
0070  4a 00 ff ff ff 00 00 00  00 00 00 00 00 01 00 02   J....... ........
0080  00 00 84 e3 6b 00 ff ff  ff 00 00 00 00 00 00 00   ....k... ........
0090  00 01 00 02 00 00 84 e3  6e 00 ff ff ff 00 00 00   ........ n.......
00a0  00 00 00 00 00 01                                  ......
```

2. Généralement, les protocoles de routage diffusent leurs informations. Pour cela des adresses de diffusion *(broadcast)* sont utilisées. Ici RIPv2 est utilisé. Ce dernier favorise la diffusion limitée *(multicast)*. Typiquement, ce type d'adresse se reconnaît au niveau de la couche MAC par le premier bit d'adresse transmis à 1 et au niveau de la couche IP par les 4 bits de poids fort du premier octet de l'adresse positionné à `0xE` (ex-classe D). Que pouvez-vous dire à propos des adresses présentes dans la trame ci-dessus ?
3. Justifiez la valeur du TTL.
4. A votre avis, quel est le mécanisme de fiabilité mis en œuvre ?
5. Vous n'avez pas d'information sur la structure de la couche applicative transportée. Sachant ce que doit transporter un protocole de routage basé sur les vecteurs de distances pour IP avec un adressage CIDR, essayez de déduire les champs transportés.

6. Utilisez à présent le logiciel `ethereal` pour le fichier `/Infos/lmd/2008/master/ue/ares-2008oct/tme-rip.dmp.gz`. Qu'observez-vous dans cette trace ?
7. Confirmez les hypothèses d'analyse des divers champs des messages du protocole RIP.
8. A votre avis, quel serait le vecteur construit à son tour par le routeur suivant après avoir reçut les messages observés ?

2 Protocole de routage interne : OSPF

1. Pour ne pas vous refroidir, analysez encore **manuellement** et rapidement la trame suivante :

```
0000  01 00 5e 00 00 05 00 e0  18 b1 0c ad 08 00 45 c0   ..^..... ......E.
0010  00 40 08 12 00 00 01 59  65 dd c0 a8 aa 08 e0 00   .@.....Y e.......
0020  00 05 02 01 00 2c c0 a8  aa 08 00 00 00 01 27 3b   .....,.. ......';
0030  00 00 00 00 00 00 00 00  00 00 ff ff ff 00 00 0a   ........ ........
0040  02 01 00 00 00 28 c0 a8  aa 08 00 00 00 00         .....(.. ......
```

2. Quelles différences observez vous par rapport à l'échange d'information RIP ? Justifiez.
3. Utilisez à présent le logiciel `ethereal` pour la suite.
   Chargez le fichier suivant : `/Infos/lmd/2008/master/ue/ares-2008oct/tme7-ospf.dmp.gz`.
   Quel échange observez-vous dans cette trace ?
4. Quelle est la signification des différents types de messages successivement échangés ?

3 Protocole de routage externe : BGP

1. Dernière analyse **manuelle** : étudiez la trame suivante :

```
0000  00 00 0c 35 0e 1c 00 c0  4f 23 c5 95 08 00 45 00   ...5.... O#....E.
0010  00 45 48 e9 40 00 40 06  70 49 c0 a8 00 0f c0 a8   .EH.@.@. pI......
0020  00 21 08 4c 00 b3 d6 33  9d 62 7a 40 e0 46 50 18   .!.L...3 .bz@.FP.
0030  7d 78 19 03 00 00 ff ff  ff ff ff ff ff ff ff ff   }x...... ........
0040  ff ff ff ff ff ff 00 1d  01 04 fe 09 00 b4 c0 a8   ........ ........
0050  00 0f 00                                          ...
```

2. Quelles différences observez-vous par rapport aux échangex d'information RIP et OSPF ? Justifiez.
3. Utilisez à nouveau `ethereal` pour le fichier suivant : `/Infos/lmd/2008/master/ue/ares-2008oct/tme7-bgp.dmp.gz`.
   Quel échange observez-vous dans cette trace ?
4. Quelle est la signification des messages successivement échangés ?
5. Si le trafic RIP observé précédement atteignait le routeur BGP de bordure de son AS, quel type d'information pourrait être alors observée dans la connexion BGP ?

4 Etude de trafic de VLAN *(facultatif)*

1. Utilisez le logiciel `ethereal` pour le fichier : `/Infos/lmd/2008/master/ue/ares-2008oct/tme7-vlan.dmp.gz`. Qu'observez-vous dans cette trace ?
2. Définissez les grandes catégories de trafic observées.
3. Pouvez-vous les regrouper grâce à un découpage logique du réseau ?
4. Ces trafics peuvent-ils communiquer entre eux ?

U.E. ARES - Travaux sur Machine Encadrés n°8

Introduction à NS2/NAM

Nous nous intéressons dans ce TME à un outil de simulation de réseau appelé NS2 *(Network Simulator 2)*. Ce logiciel a été développé dans un contexte académique au sein du LBL (`http://www.lbl.gov/`), du Xerox PARC(`http://www.parc.xerox.com/`), de l'UCB(`http://www.berkeley.edu/`) et de USC/ISI (`http://www.isi.edu/`, c'est cette dernière institution qui maintient le code actuellement).

NS2 est un simulateur à événements discrets qui permet d'exécuter tout type de scenarii sur des topologies définies par l'utilisateur. Pour visualiser les résultats, un outil graphique a été développé : NAM *(Network AniMator)*. C'est principalement avec ce dernier que nous travaillerons. NAM permet, outre la visualisation des résultats de simulation, l'édition de scenarii simples permettant d'éviter de descendre au niveau programmation de NS2 (basée sur du Tcl objet).

1 Remarques préalables

1. Avant d'exécutez l'outil graphique, veuillez consulter la **documentation** de NAM en ligne (`http://www.isi.edu/nsnam/nam/index.html`). En particuliez, consultez le manuel de NS (`http://www.isi.edu/nsnam/ns/doc/index.html`) qui contient une section sur NAM.
2. La réalisation des simulations génère des fichiers de taille imposante. Il est impératif d'utiliser un répertoire local à la machine pour votre travail. Par exemple `/tmp/mon_login/` :

   ```
   [mon_login@localhost:~] $ mkdir /tmp/mon_login
   [mon_login@localhost:~] $ cd /tmp/mon_login
   [mon_login@localhost:/tmp/mon_login] $ nam
   ...
   ```

   Les configurations créées avec l'éditeur doivent utiliser le suffixe `.ns`, et les traces des simulations terminent par `.nam`. Noubliez pas de recopier les configurations (en déplaceant les fichiers locaux sur votre compte utilisateur) et d'effacer les traces à la fin de la séance :

   ```
   [mon_login@localhost:/tmp/mon_login] mv *.ns ~/
   [mon_login@localhost:/tmp/mon_login] rm *.nam
   [mon_login@localhost:/tmp/mon_login] cd ; rmdir /tmp/mon_login
   ```

2 Edition avec NAM

Après avoir exécuté NAM, une fenêtre de contrôle apparaît. Dans le menu "File" choisissez "New Nam Editor...". Un éditeur de scenario NS apparaît. Cet outil permet de réutiliser l'environnement graphique de NAM pour générer des scripts NS simples que vous pourrez simuler ultérieurement avec NS puis visualiser le résultat avec à nouveau NAM.

1. A partir de l'éditeur, vous pouvez manipuler quatre types d'**objets de base** :
   - noeuds *(nodes)*
   - liens *(links)*
   - agents *(agents)*
   - sources *(sources)*

   Que représentent-ils ? Quels paramètres pouvez-vous modifier ?
2. Avec l'éditeur, vous pouvez créer des objets, les manipuler (position, paramètre...) ou les détruire. Lorsque vous êtes satisfait de votre configuration, sauvegardez-la. Le fichier résultant est un script NS (utilisez le sufixe `.ns`). Vous pouvez alors lancer une simulation. Cette action correspond à exécuter la commande NS sur le script correspondant à votre configuration. Un fichier trace de la simulation est généré (suffixe `.nam`). NAM affichera automatiquement une fenêtre de visualisation de cette trace.

3 Exécution avec NS2

1. Vous allez tester une première configuration avec deux noeuds. Cliquez sur l'icône représentant un noeud et placez deux instances dans la fenêtre d'édition. Cliquez ensuite sur sur l'icone représentant un lien et reliez les deux noeuds précédents. Editez les paramètres associés au lien : indiquez un débit de 2Mbits et un délais de 200ms. Définissez ensuite deux agents : **TCP** sur le premier noeud et **TCPSink** sur le second noeud. Reliez les deux agents entre eux et paramétrez une taille de paquets de 41 octets typique du trafic `telnet`. Pour finir, intégrez une source telnet avec un temps inter-emission moyen de 200ms sur l'agent TCP. Sauvegardez votre configuration. Observez le script NS générée dans le répertoire avec votre éditeur de texte habituel.
2. Lancez la simulation. A la fin de la génération de la trace, une fenêtre NAM de visualisation est lancée. Avant de l'utiliser, observez la trace NAM générée dans le répertoire. Revenez à la visualisation et démarrez l'animation correspondant à la simulation. Expliquez ce que vous observez. *Pour affiner les paramètres de simulation, vous pouvez directement modifier le script NS avec votre éditeur de texte puis lancer la simulation manuellement avec la commande ns.*

4 Scenarii à réaliser

1. **Scénario avec pertes** :
   Réalisez une topologie linéaire avec trois noeuds et recherchez quelles sont les possibilités d'intégrer des pertes au niveau du noeud central. Visualisez le phénomène avec l'agent et la source de votre choix. Utilisez ensuite une configuration avec TCP permettant d'observez les mécanismes de fiabilité mis-en-oeuvre par ce dernier.
2. **Scénario multi-trafics** : Réalisez une topologie en "H" avec 6 noeuds (les noeuds 1 et 2 sont reliés au noeud 3, lui-même relié au noeud 4, qui à son tour est relié aux noeuds 5 et 6) et deux agents : un TCP (entre les noeuds 1 et 5) et un UDP (entre les noeuds 2 et 6). Chaque lien a un délai de 20ms et un débit de 1Mbps. La connexion TCP supporte un générateur de type FTP qui démarre à t=5s. La transmission UDP supporte un générateur de type CBR qui démarre t=0 à débit nul, puis émet à t=10s à 0,5 Mbps pendant 5s, puis débit nul, puis émet à t=20s à 1Mbps pendant 5s, puis finalement débit nul. Observez et analysez les pertes et la réaction de TCP.

U.E. ARES - Travaux sur Machine Encadrés n°9

Contrôle de congestion TCP

Les études s'effectueront avec le simulateur ns-2 et la bibliothèque de scripts Tcl développée à l'Université de la Réunion. La page compagnon au TME est à l'URL :

`http://www.univ-reunion.fr/~panelli/enseignement/TP-NS-2/`

Le protocole TCP est complexe, ce qui le rend particulièrement difficile à analyser et à modéliser mathématiquement. La mesure directe sur un réseau réel pose également des problèmes car l'environnement n'est pas maîtrisable. Dans ces conditions, la simulation est une alternative intéressante.

Dans ce TME, nous allons étudier le comportement de TCP en présence d'un goulot d'étranglement. Nous proposons d'analyser les performances induites par le contrôle de congestion[1] pour une connexion. Les études s'effectueront avec le simulateur ns-2. et une bibliothèque de scripts qui nous simplifiera la tache.

1 Installation de la bibliothèque

L'ecriture d'un modèle de simulation ns-2 est un travail fastidieux qui demande une bonne connaissance de l'interface OTcl. L'Université de la Réunion a développé une bibliothèque de scripts afin de pouvoir constituer rapidement et sans connaissance du langage Tcl, un modèle de simulation pour ns-2. L'idée de base est de séparer les données des instructions. Les données sont les paramètres qui décrivent le modèle de simulation et les traitements d'analyse à réaliser sur les traces de simulations. La valeur des paramètres sont définies dans un fichier dit de ressources. Ce fichier est interprété par les fonctions de la bibliothèque afin de réaliser une étude.

Récupérez l'archive de la bibliothèque de scripts et le guide d'utilisation sur la pages web indiquée ci-dessus et décompressez l'archive dans votre répertoire de travail :

```
mkdir TME9; cd TME9
wget --user=<id-a-demander> --password=<mdp-a-demander> \
     http://www2.univ-reunion.fr/~panelli/enseignement/TP-NS-2/Scripts/tp_ing<version>.tar.gz
tar xvf tp_ing<version>.tar.gz
```

Configurer votre environnement UNIX (ici pour bash) :

```
which ns # doit afficher la version de ns-2 : /usr/local/ns-allinone-2.<vers-ns-2>/bin/ns
ln -s /usr/local/ns-allinone-2.<vers-ns-2>/ns-2.<vers-ns-2> ~/ns-2
LANG=POSIX
PATH=$PATH:\.
```

Exécution d'une simulation (veillez à l'option `-wd /tmp/<mon-rep-local>` pour une exécution locale) :

```
ns test-suite-masterv1.0.tcl Simple -f <non-du-fichier-de-ressource> -wd /tmp/<mon-rep-local>
```

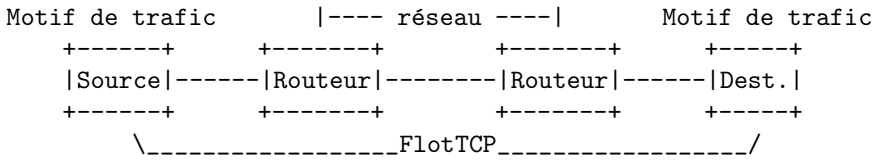
La bibliothèque de scripts vise à constituer rapidement des modèles de simulations sans avoir à manipuler le code Tcl. Elle offre des fonctions d'analyse des traces produites par la simulation et est constituée par un ensemble de fichiers Tcl localisés dans le répertoire 0-Lib. Les éléments contenus dans le fichier ressources décrivent une topologie linéaire avec :

- les routeurs et les artères de communication (partie du réseau partagée par un ensemble de flux où se produit la contention entre les flots).
- les hôtes (arrangés par paire ; à un hôte source correspond un hôte destination), les sources et puits de trafic associés, et leur lien d'accès
- les analyses faites en fin de simulation,
- les paramètres de calculs des analyses *(post process)*.

Lorsque de nombreuses études sont à effectuer, un fichier de ressources doit être écrit pour chaque étude. Ainsi il est possible de rejouer une simulation ou de relire les paramètres du modèle simulation lors de l'analyse de ses résultats. De plus, les résultats de simulation peuvent se retrouver facilement. Ainsi une opération de sauvegarde ne porte que sur les fichiers ressources.

[1] Le contrôle de congestion de TCP est décrit dans le RFC 2581

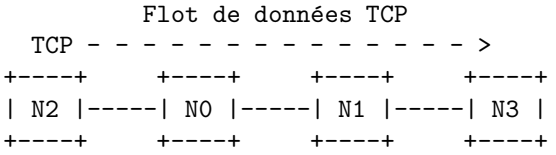
Les modèles de simulation qui peuvent être constitués avec cette bibliothèque se limitent à l'étude de la contention avec un ou plusieurs goulot d'étranglements. Un modèle de simulation dans le cas d'un simple goulot d'étranglement se représente selon la figure suivante :



2 Auto-synchronisation de TCP

Le contrôle de congestion de TCP est mis en oeuvre au moyen d'une fenêtre coulissante. La fenêtre coulisse (s'ouvre) chaque fois qu'un acquittement est reçu. Il s'ensuit que l'émission des segments est synchronisée sur la réception des acquittements. TCP utilise cette auto-synchronisation *(self clocking)* pour adapter son débit d'émission à celui du débit du goulot d'étranglement du chemin. Ainsi TCP peut s'utiliser sur une route où les liens ont chacun une bande passante différente.

Pour démontrer cette capacité, réaliser le modèle de simulation à 4 nœuds représenté par la figure suivante :



- Le temps de propagation de chaque lien est de 10 ms sauf celui central qui a un délai de 100ms.
- Les liens offrent une bande passante de 1Mbit/s sauf le goulot d'étranglement qui offre une bande passante de 100kbits/s.
- La valeur par défaut de la longueur d'une file d'attente dans ce réseau est de 10 paquets.
- La taille d'un paquet est de 625 octets.
- La taille maximum de la fenêtre de contrôle de flux est fixée à 10 segments.
- Une durée de la simulation de 5 secondes sera suffisante.

1. Placer le goulot d'étranglement sur le lien central, observer avec NAM comment TCP synchronise son débit d'émission à celui du goulot d'étranglement.
2. Est-ce que l'adaptation du débit de TCP dépend de la localisation du goulot d'étranglement par rapport à la source ? faites varier la localisation en plaçant le goulot d'étranglement proche de la source (lien `N2-N0`) puis loin de la source (`N1-N3`). Evaluer l'impact que cela a sur l'auto-synchronisation de TCP et sur la performance de TCP. Dans chacun des 2 cas, observer l'évolution des segments de données émis (paramètre t_seqno) sur une simulation de 5 secondes.

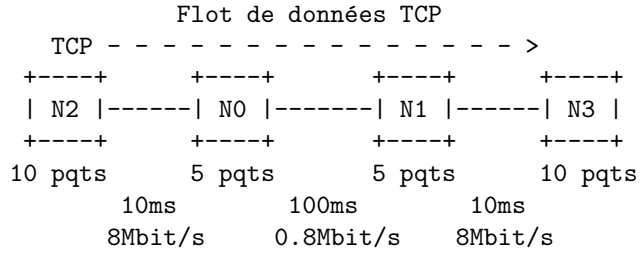
   En déduire le débit écoulé de la connexion.

3 *Slow-start* et *congestion avoidance*

1. Après une période d'inactivité ou au démarrage d'une connexion, TCP entame une phase de *slow-start*. Mais quelle est l'utilité du *slow-start* ? Nous allons mettre en évidence son utilité. En reprenant le modèle représenté par la figure précédente
   – Le goulot d'étranglement est placé sur le lien central.
   – La file d'attente au niveau du goulot d'étranglement est fixée à 5 paquets.
   – La simulation durera 10 secondes.
   Faire une simulation sans puis une autre avec le *slow-start*. La présence ou l'absence du slowstart s'indique par le booléen : `add793slowstart` quand l'agent TCP de la source est `TCP/RFC793edu`.
   Analyser les différences entre les 2 simulations. Vous pouvez essayer d'appréhender les différences avec NAM (argument `-nam 1` ou `-namgraph 1` sur la ligne de commande du lancement de l'exécution de la simulation). De manière plus précise vous pouvez voir les différences en demandant à tracer les courbes d'activité du flot TCP, du débit écoulé, du débit écoulé total. **Attention :** Les courbes sont tracées uniquement si NAM n'est pas demandé. Si vous faites la simulation avec l'argument `-namgraph 1` vous pouvez demander à afficher le graphe de la session dans une fenêtre de NAM (menu Analysis, item Active session). Décrire le fonctionnement du *slow-start*. En conclusion, indiquer à quoi sert le *slow-start* ?
2. Une fois la phase de démarrage achevée, TCP rentre dans une phase de *congestion avoidance* dans lequel il sonde progressivement la bande passante disponible. L'augmentation de la taille de fenêtre de contrôle de congestion est dans cette phase est linéaire. Reprendre le modèle précédent et exécuter une nouvelle simulation avec le booléen `add793additiveinc` à vrai. Tracer la courbe d'activité du flot TCP et la courbe du débit écoulé. Identifier les périodes pour lesquelles le débit écoulé est de 100%. Observer avec NAM comment se traduit en terme d'émission l'augmentation de la fenêtre de contrôle de congestion. Dans quelle condition l'augmentation de la fenêtre se traduit elle par une augmentation du débit écoulé ? Au delà de cette condition quelle est la conséquence d'une augmentation de la fenêtre ?
3. Le contrôle de congestion de TCP repose sur la fonction de contrôle de fenêtre AIMD (Additive Increase, Multiplicative Decrease). Nous allons étudier la dynamique d'évolution de la fenêtre de contrôle de congestion de TCP. Pour cela, nous allons retenir la version Tahoe de TCP qui est la première mise en oeuvre du contrôle de congestion de TCP. En NS, la version Tahoe s'appelle TCP tout simplement. Modifier le modèle précédent, de la manière suivante :
   – La taille maximum de la fenêtre vaut 12 segments.
   – Une simulation durera 15 secondes.
   Dans le fichier des ressources, vous prendrez soin d'indiquer les paramètres de TCP que vous voulez afficher dans le monitor TCP de NAM :
   – `ack` : numéro d'acquittement
   – `t_seqno` : numéro de séquence
   – `cwnd` : fenêtre de contrôle de congestion
   – `dupacks` : nombre des acquittements dupliqués
   – `maxseq` : plus grand numéro de séquence émis
   Ainsi vous pourrez suivre dans NAM l'évolution des principaux paramètres des mécanismes du contrôle de congestion de TCP. A la fin de la simulation, NAM s'exécute :
   (a) Visualiser à l'aide du *monitor* le fonctionnement du contrôle de congestion de TCP. Observer la croissance et la diminution de la fenêtre de contrôle de congestion. Expliquer comment s'effectue la diminution de la fenêtre.
   (b) Identifier le passage du *slow-start* au congestion avoidance. Par quoi se différencie au niveau de la source ces deux phases du contrôle de congestion de TCP ?
   (c) Rejouer l'animation NAM en affichant le graphe de la session. Faites le parallèle entre la taille de fenêtre de congestion et les segments de données émis. Observez vous des pertes ? si oui comment sont-elles détectées ? Déduisez les caractéristiques de cette version de TCP.
4. Nous allons maintenant abandonner NAM et tracer l'évolution de la fenêtre de contrôle de congestion en fonction du temps. Vous allez demander aussi à tracer l'activité du flot TCP. Relancer l'exécution de la simulation du modèle précédent en omettant l'argument `-namgraph`. Identifier les phases de slow-start et de congestion avoidance sur la courbe cwnd. Avec la courbe d'activité du flot TCP, identifier le moment où l'algorithme du fast retransmit est utilisé. Faites le rapprochement avec les conclusions de la question précédente. Pourquoi la variation de la taille de la fenêtre de contrôle de congestion est elle cyclique ?
5. En terme de performance, quel est le débit utile écoulé *(goodput)* ? Exécuter une simulation avec la version `TCP/RFC793edu` et la version TCP (Tahoe) pour comparer l'amélioration apportée par le contrôle de congestion. Au terme de cette étude, expliquer comment TCP contrôle son débit avec une fenêtre de congestion. Un rappel synthétique des principes du fonctionnement du contrôle de congestion de TCP pourrait être fait dans le rapport à l'aide du RFC 2581.

4 Fast recovery

Depuis la première proposition, le contrôle de congestion de TCP a évolué afin d'améliorer la performance à la fois en terme de débit écoulé et en terme de taux d'utilisation des ressources du réseau. Le *fast recovery* est le dernier mécanisme ajouté à TCP. Il s'active suite à une reprise par *fast retransmit*. Par la suite l'étude du *fast recovery* portera sur une simple perte à l'intérieur d'une même fenêtre. Le modèle de simulation utilisé *(Long Fat Network)* est représenté par la figure suivante :



- La taille maximum de la fenêtre est fixée à 40 segments.
- Les paquets ont une taille de 1000 octets.
- La durée de simulation sera fixée à 30 secondes.

1. Identifier le défaut dont souffre TCP Tahoe en terme de performance. Pour cela tracer le débit écoulé et la fenêtre de contrôle de congestion. Existe t-il une marge de progression du débit ? si oui expliquer d'où elle peut provenir ?
2. La version TCP Reno comporte les mêmes fonctions que TCP Tahoe à l'exception du *fast recovery*. Refaire la simulation précédente en adoptant la version TCP/Reno. Evaluer l'amélioration du débit apporté. Quelle est l'explication ?
3. Afin de comprendre le fonctionnement du *fast recovery*, nous allons utiliser NAM. Relancer l'exécution du modèle de simulation avec l'argument -namgraph 1 sur une durée de simulation de 15 secondes. Expliquer comment TCP peut maintenir une émission pendant la phase de reprise ? A l'aide du graphe de la session, identifier quand se termine la phase de fast recovery ?
4. Pour finir, relancer l'exécution du modèle de simulation en demandant de tracer la courbe d'activité du flot (l'argument `-namgraph` doit être absent). Identifier les segments émis pendant la phase de *fast recovery* et le moment auquel se termine cette phase.

En conclusion, expliquer en quoi TCP Reno améliore la performance de la connexion TCP. Rappeler le principe du *fast recovery*.