POUGNES PHOENIX TSP

NET 3102 - Réseaux de données

Modèle OSI

Modèle théorique en couche. La plupart des systèmes susceptibles d'échanger des données essayent de suivre ce modèle, même si Internet fonctionne en fait grâce à TCP/IP, modèle plus simple.

Un protocole permet d'établir une communication entre 2 entités au sein d'une même couche.

Le transport des données se fait des couches hautes vers les basses.

7	Application
6	Présentation
5	Session
4	Transport
3	Réseau
2	Liaison de données
1	Physique

Rapide détail des couches

Couche Physique - 1 : Support physique de la transmission (bits).

Matériel associé : câble coaxial, paire torsadée, fibre optique, **hub**.

Topologies de réseaux : en bus, en anneau, en étoile... la dernière étant la plus fiable.

Couche Liaison de données - 2 : Interconnection des machines au sein d'un même réseau (trames).

Matériel associé : **switch = commutateur**

Protocoles associés : ex : Bit alterné, HDLC [1], Spanning Tree Protocol (STP)...

Identité : **adresse MAC**

Couche Réseau - 3 : Interconnection des réseaux entre eux. (paquets/datagrammes).

Matériel associé : **routeur**

Protocole : ex : IP

Identité : **adresse IP**

Couche Transport - 4 : Garantir la connexion.

Couche Session - 5 : Contrôle du dialogue.

Couche Présentation - 6 : Représentation des données.

Remarque. *Internet fonctionne aujourd'hui grâce à TCP/IP. Dans ce modèle, les couches 5-6 sont confondues avec la couche application et ne servent pas concrètement.*

Couche Application - 7 : Ce que l'utilisateur voit.

1. Même plus utilisé. Désolé...

Définitions importantes

Bande passante

Capacité de transmission entre l'émetteur et le récepteur.

Superficie du réseau [2]

$$\underset{(bluetooth)}{Personal(PAN)} \rightarrow \underset{(wifi)}{Local(LAN)} \rightarrow \underset{(ATM/wimax)}{Metropolitan(MAN)} \rightarrow \underset{(GSM)}{Wide(WAN)}$$

Quantité d'information

$$I = ln(\frac{1}{p})$$

I : quantité d'information
p : probabilité que le message soit reçu.

Débit

$$D = Bln(1 + \frac{S}{N})$$

D : débit
B : bande passante,
S : signal
N : bruit.

Types de communication

Simplex : un seul équipement peut peut émettre.

Half-duplex : plusieurs équipements peuvent émettre, mais à tour de rôle.

Full-duplex : les équipements peuvent émettre simultanément.

Types de service [2]

1 - Orienté connexion : fiabilité/protocole préliminaire pour étblir une connexion.
ex : commutation de circuits, commutation de circuits virtuels (TCP)

2 - Sans connexion : possibles pertes et erreurs, mais bien pour le broadcast [3].
ex : commutation de messages, commutation de paquet

2. AN = Area Network
3. message de diffusion envoyé à toutes les machines connectées sur un même réseau

Types de commutation [4]

1 - Messages : pas de réservation de ressources ; pour envoyer le message d'un équipement à un autre, il faut qu'il soit totalement émis, donc reçu.
Inconvénient : temps d'attente.

2 - Circuit : réservation des ressources.
Inconvénient : la monopolisation d'un circuit lors d'une liaison implique du gaspillage (réserver, ce n'est pas utiliser).

3 - Paquets : découpage des données en paquets, problème de réassemblage.

Unités de données

SDU Service Data Unit : ce que reçoit une couche de la part d'une couche supérieure.

PDU Protocol Data Unit : ce que reçoit une chouche de la part d'une entité.

PCI Protocol Control Information : données propres.

$$\begin{cases} PDU_{(N)} = SDU_{(N)} + PCI_{(N)} \\ PDU_{(N)} = PCI_{(N)} + PDU_{(N+1)} \end{cases}$$

Condition de synchronisation horloge du récepteur synchronisée avec celle de l'émetteur.

2 types de support Tangible (fibre optique, cable coaxiale, paire torsadée)
ou non tangible (hertzien,sans fil).

Délimitation

Pour synchroniser émetteur - récepteur, et savoir quand commence l'information.
3 types de délimitations :

1. Utiliser le silence (...)
2. Fanions : motifs prédéfinis en début et fin d'unité de données.
   Les fanions sont notamment utilisé par le protocole de couche 2 HDLC. Ils se concrétisent par le délimiteur 01111110 [4].
3. Champ de comptage = champ de contrôle = taille des données utiles.
   On le place *avant* les données à envoyer. Utilisé par le protocole IP.

Les erreurs

Définition. *On définit le BER (Bit Error Rate) par :* $BER = \frac{E}{R} \xrightarrow[ideal]{} 0$

E : Nombre de bits erronés
R : Nombre de bits reçus total

Pour détecter les erreurs, on utilise un code de contrôle :
ex : *CRC* ou *Bit de parité* (un bit valant 1 si le nombre de bits à 1 de la PDU est impair, 0 sinon)

Remarque. *À capacités de détection et de correction égales : coût de correction > coût de détection* [5].

4. Si l'unité de données à envoyer contient elle-même la suite de bit 01111110,
on utilise la transparence en remplaçant la suite par 011111010
5. Du coup, en général les paquets comportant des erreurs sont simplement détruits après détection.

Les pertes

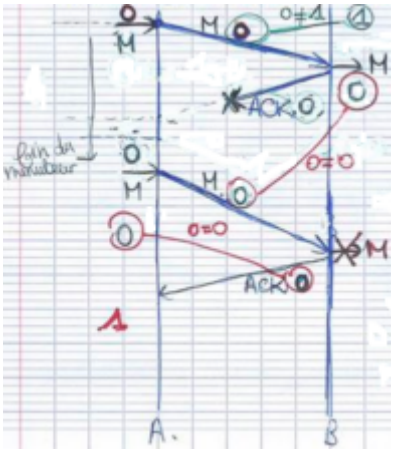
Origines : sautration, erreurs, saturation...
Pour les détecter : Protocole de reprise sur perte (chez l'émetteur ou le récepteur).

Le minuteur : Utilisation d'un temporisateur. Si le paquet n'est pas arrivé au récepteur avant la fin du minuteur, on pert le paquet.

L'acquittement - ACK : Sorte "d'accusé de réception" vers l'émetteur pour s'assurer que le message est bien reçu. Entraine la désactivation du minuteur.

Remarque. *Si l'acquittement se perd, l'info envoyée est dupliquée et cela implique des problèmes. 2 solutions : protocoles*[6] *du bit alterné ou HDLC.*

Protocole du bit alterné



— Comme $0 \neq 1$, le message M est délivré à B et le 1 passe à 0. L'acquittement se perd. Le message est renvoyé, mais :
— Comme $0 = 0$, M est détruit. Ensuite, comme l'acquittement est tel que $0 = 0$, on passe à 1 au niveau de l'émetteur.

Contrôle de flux[7] **:** contrôle le rythme d'émission : le contrôle de flux de la couche n permet à l'émetteur de la couche n de ne pas saturer le récepteur de la couche n+1.

Manière réactive (Ethernet, HDLC) ou préventive (TCP(IP), HDLC).

Fenêtre

Nombre max de messages envoyés sans avoir reçu d'acquittement.
Un envoi de données incrémente la taille de la fenêtre, la réception d'un acquittement la décrémente.

Exemple. *Si la taille de la fenêtre est de 2, on ne pourra pas envoyer plus de 2 messages tant que l'on n'a pas reçu l'acquittement : pour envoyer un* 3^e *message, on doit attendre que le récepteur nous envoie l'acquittement des 2 messages précédents.*

6. de couche 2

Protocole HDLC

Dans ce protocole, toute trame erronnée est perdue et jetée à la poubelle.
Une trame contient, dans l'ordre :
— Délimitation/transparence
— Détection d'erreur/perte
— Reprise sur perte
— Contrôle de flux

Trame type

Délimiteur	Adresse	Champ de contrôle	Données	CRC	Délimiteur
1 octet	1 octet	1 octet	n octets	2 octets	1 octet

3 types de trames :

→ Trame d'information : commence par un 0.
(taille en bits)

0 (1)	NS (3)	P/F (1)	NR (3)

→ Trame de supervision : commence par 10.
(taille en bits)

1 (1)	0 (1)	C (2)	P/F (1)	_ (3)

→ Trames non numérotées : commence par 11.

Légende :
C : Control
RR : Receive Ready 00
RNR : Receive Not Ready 10
REJ : Reject 01
SR : Selective Reject 11
NS, NR : Voir après.

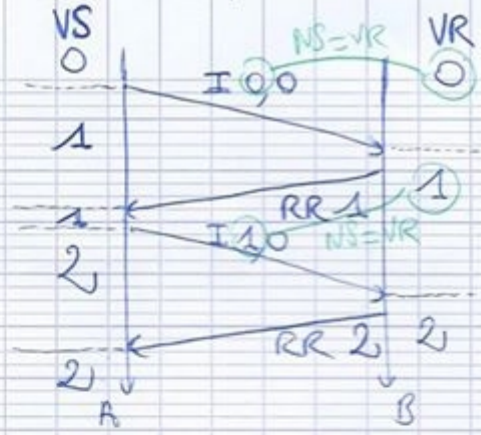
Mise en pipeline : Pour augmenter le débit.

Piggy backing : Pour envoyer l'ACK en même temps que l'émission d'un message.
⇒ Acquittement plus rapide
⇒ Pas de PDU [8] de contrôle lorsque les PDUs sont dispos.

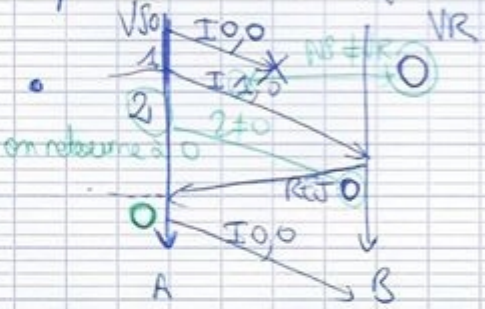
4 variables : VS ↔ NS, VR ↔ NR
→ X = R ou S, Received or Send
→ VX = variable de vérification (copie de NX)
→ NX = nombre de message reçu/envoyé
· Si NS = VR, on incrémente VR
· À chaque envoi de message, on incémente VS

8. Protocol Data Unit - Unité de donnée de protocole

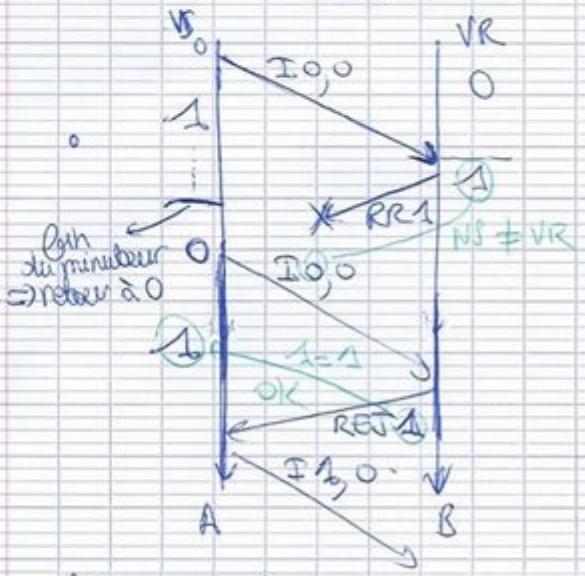
• Si tout se passe bien.



Acquittement : RR (valeur attendue VR).



si l'envoi est perdu en route.



si l'acquittement est perdu.

AUTOMATES :

○ : état

— : transitions (act° associées à chaque état)

Protocole Spanning Tree

Il permet de créer un chemin sans boucle dans un environnement commuté
et physiquement redondant. → Trouver l'arbre de commutation grâce aux "vecteurs".

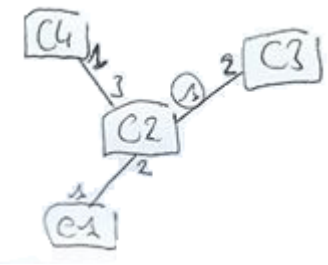
Un vecteur s'écrit sous la forme : $C1.2 : 2/C3.2/19$
→ C1.2 est ici le port 2 du commutateur 1 : c'est le port du commutateur émetteur du message
→ 2 est ici la priorité du port racine
→ C3.2 est ici le port racine supposé lors de l'envoi du vecteur
→ 19 est ici le coût du chemin entre C3.2 et C1.2
⇒ Le vecteur s'écrit $V(B_i, P_i, I_i, C_i)$

Le meilleur vecteur sera toujours **le plus petit**. Pour les comparer, on regarde dans l'ordre :
— Les priorités P_i
— Les coûts C_i
— L'émetteur B_i

Exemple. *Entre* $C1.2 : \mathbf{2}/C3.2/19$ *et* $C3.1 : \mathbf{3}/C1.1/19$*, le meilleur vecteur est le premier car* $2 < 3$.
Entre $C1.\mathbf{2} : 2/C3.2/19$ *et* $C1.\mathbf{1} : /2/C3.2/19$*, le meilleur est le second car* $1 < 2$.

Méthode pour déterminer l'arbre de commutation :

1. Trouver la racine de l'arbre en choisissant le plus petit vecteur.
   B_i est donc un port de la racine ⇒ Le commutateur racine est le commutateur de B_i
2. Trouver le port racine [9] des autres commutateurs.
   On regarde les vecteurs qui arrivent vers le commutateur considéré et on choisit le meilleur.



Ex : si on choisit de trouver le port racine de C2, on va regarder les vecteurs provenant de C4.1 C3.2 et C1.1 et les comparer.
Si le meilleur vient de C3.2, alors le port racine de C2 est C2.1.
Entourer les ports racine.

3. Trouver les ports désignés [10].

   On regarde les liens entre les commutateurs.
   — Ex : si entre C2 et C4 plusieurs vecteurs circulent, on les compare, on prend le meilleur, et le B_i correspondant sera un port désigné.
   — Ex : pour le lien entre C1 et C3, on a les vecteurs : $C3.1 : 2/C2.1/19$, $C3.1 : 1/C1.1/57$ et $C1.1 : 1/C1.1/0$, donc C1.1 est un port désigné.

   Traiter tous les liens, et **souligner les ports désignés**.
4. Trouver les ports bloqués.
   Il s'agit des ports non désignés.
5. Dessiner l'arbre de commutation/recouvrement.

À savoir sur les switchs

CAM : Il met à jour sa table de commutation CAM quand il voit passer une trame. Il envoie un trame à tout le monde ss'il n'a pas l'adresse MAC de destination dans sa table CAM.

Remarque : la table CAM a une durée de vie, elle se réinitialise si elle n'a pas reçu de trame depuis un temps supérieur à son TTL (Time To Live).

Noter que le switch n'a pas d'@MAC (car on n'a pas besoin de parler avec).

CSMA/CD : Gérance des collision lors de l'envoi des messages pour les configurations en bus.

→ en résumer, attendre avant d'envoyer "droit à la parole" :

Si A et B parlent en même temps, A attendra 2s et B 3s (par ex) avant de vouloir à nouveau communiquer.

Ici A recommencera à parler au bout de 2s. B verra que A parle et attendra la in de la communication de A...

Adressage IP

IPv4 : codage de l'adresse sur 4 octets / 32 bits (157.159.48.100 par exemple)

IPv6 : codage de l'adresse sur 16 octets / 128 bits (FE80 : :B3FF :AB35 :1583 par exemple [11])

Masque de sous-réseau

L'adresse IP [12] comporte en pratique dans le cas de **l'adressage sans classe** une partie servant à identifier le réseau et une servant à identifier la machine. La distinction se fait grâce au masque de sous-réseau. Il est indissociablement associé à l'@IP et est constitué de bits à 1 successifs puis de 0 successifs.

Ex : 255.255.128.0 ⇔ 11111111.11111111.10000000.00000000

Par justaposition [13] de l'@IP et du masque :

- La partie de l'@IP justaposée avec les 1 du masque est *la partie réseau.*
- La partie de l'@IP justaposée avec les 0 du masque est *la partie machine.*
  Elle est unique et propre à chaque machine.

2 cas particuliers d'@IP :

Adresse broadcast - 255.255.255.255 : envoie le message sur tout le réseau.

Adresse réseau : Constituée uniquement de la partie réseau de l'@IP, les autres bits étant à 0.

Pour ne pas recopier le masque, on notera une @IP complète ainsi : **157.159.54.2/20**,
où le 20 représente le nombre de bits à 1 du masque.

Remarque. *(Adressage par classe) Anciennement utilisé en IPv4.*

- *Classe A : partie réseau sur 1 octet (l'@IP commence par un 0)*
- *Classe B : partie réseau sur 2 octet (l'@IP commence par un 10)*
- *Classe C : partie réseau sur 3 octet (l'@IP commence par un 110)*

11. Le " : :" symbolise un octet de bits à 0)
12. Que l'on notera @IP dans la suite
13. ET logique bit à bit

Création de n sous-réseaux au sein d'un réseau donné

Notons p le b nombre initial de bits à 1 dans l'@IP.
• nombre de bits nécessaires : $p' = E(\log_2 n) + 1$
• nombre de bits à 1 dans le nouveau masque = $p + p'$

Ex : Pour le réseau 128.2.2.0/23 avec $n = 3$
— Par construction du masque (/23), les 23 premiers bits des adresses réseaux recherchées seront les mêmes : 10000000.00000010.0000001
— Pour créer les n sous-réseaux, il faut agrandir l'adresse réseau en lui ajoutant $p' = E(\log_2 3)+1 = 2$ bits. Puisque $n = 3$, on peut choisir par exemple les paires 00, 01 et 10.
— Les adresses réseau obtenues sont donc (en complétant par des 0 en place de la partie machine) :
  — 10000000.00000010.0000000**0.0**0000000
  — 10000000.00000010.0000000**0.1**0000000
  — 10000000.00000010.0000000**1.0**0000000
— Le nouveau masque est donc un /(23+p') ie un /25
Les adresses réseau finales sont donc 128.2.2.0/25, 128.2.2.128/25 et 128.2.3.0/25

"Protocole" ARP : C'est en fait plus un complément du protocole IP qu'un vrai protocole en soit. Associe une adresse MAC à une adresse IP (table ARP).

DHCP : C'est un protocole applicatif (couche 7) qui distribue dynamiquement une @IP et un masque à des machines.

DNS : C'est un serveur qui convertit le nom d'un site web (url) en @IP (résolution DNS).

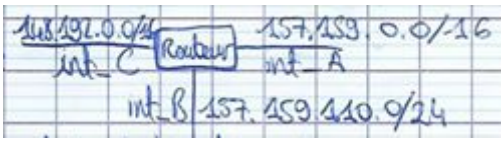
Nom de domaine : décomposé en *labels* séparés par des points (ex : www.google.fr)
Service WEB : met à disposition les pages web qu'il héberge/héberge des serveurs web
(ex : protocole HTTP)

Routage

Se fait grâce à une table de routage, à l'arrivée d'un paquet dans le routeur. Notons : A=@IP destinataire.
Alors A parcourt l'ensemble de la tabel de routage.

Sur quelle interface router un paquet ?

1. — ET logique entre A et le masque de la ligne (de la table de routage) considérée
   — ET logique entre l'@IP de la ligne et son masque
   Puis on compare les résultats : en cas d'égalité, on retient la ligne.
2. Règle du plus long masque (sur un exemple)



Paquet à destination de 157.159.110.15

Table de routage :

148.192.0.0/16 int_C

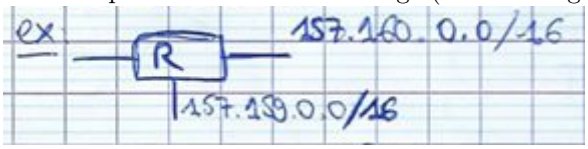
157.159.110.0/24 int_B

157.159.0.0/16 int_A

- Ici, A=157.159.110.15
  — 157.159.110.15 ET /16 = 157.159.0.0
  148.192.0.0 ET /16 = 148.192.0.0
  → 157.159.0.0 ≠ 148.192.0.0 donc int_C n'est pas retenue.
  — 157.159.110.15 ET /24 = 157.159.110.0
  157.159.110.0 ET /24 = 157.159.110.0
  → 157.159.110.0 = 157.159.110.0 donc int_B est retenue.
  — int_C retenue.
- 24 > 16 donc par la règle du plus grand masque, int_B est retenue.

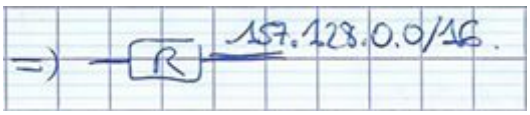
Agrégation d'adresses

Pour simplifier les tables de routage (moins de lignes).



En binaire, on garde la partie commune des adresses et on complète avec des 0 :

$$\left.\begin{array}{l} 159 : 0b\ 10|01\ 1111 \\ 160 : 0b\ 10|10\ 0000 \end{array}\right\} \Rightarrow (1000\ 0000)_2 = (128)_{10} \Rightarrow 157.128.0.0/10$$ ($Pourquoi$ 10? $Parce\ que$[14] $8+2$)



14. 8 correspond au premier octet, 2 les 2 bits rajoutés (10)