Réseaux et Transmission de Données

5 - *Couche Réseau (1)*

- *Présentation*
- *Routage*
- *Internetworking*

Maîtrise EEA

Olivier Fourmaux

Basé sur la 3éme édition du livre du Pr. A. S. Tanenbaum : ***Computer Networks***

Plan du cours de RTD

1. Introduction (3h)
2. Couche Physique (3h)
3. Couche Liaison (3h)
4. Couche d'Accès au Médium (3h)
5. **Couche Réseau (9h)**
6. Couche Transport (6h)
7. Applications (6h)

Introduction

- Gestion des paquets de la source au destinataire (1ère couche *end-to-end*)
- Utilisation de **routeurs** intermédiaires
  - Calcul de route
  - Relayage des paquets

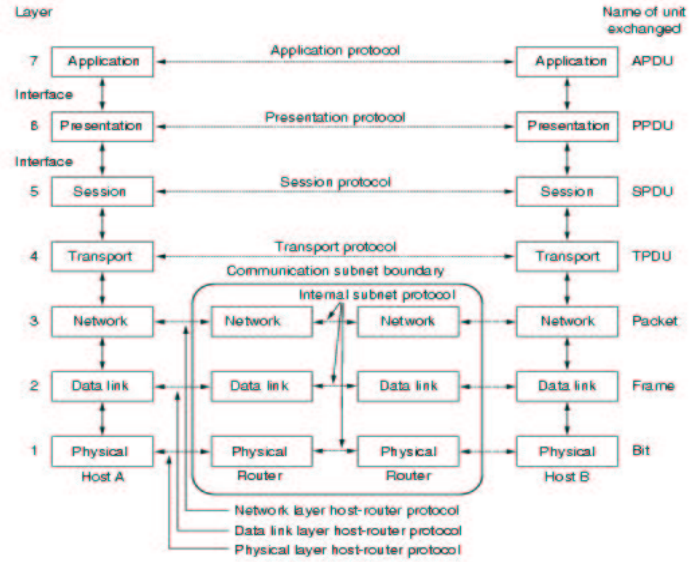
Plan - *Couche Réseau*

- **Présentation**
- **Routage**
- ***Internetworking***
- Couche Réseau dans Internet
- Couche Réseau dans ATM
- Contrôle de Congestion

Plan - *Couche Réseau (1)*

- **Présentation**
- Routage
- *Internetworking*

Services pour la Couche Transport



Buts de la Couche Réseau

- 3 buts :
  - Service Réseau **indépendant** de la technologie sous-jacente
  - **Topologie** physique masquée aux Couches Transport et supérieures
  - Plan d'**adressage** global et uniforme

Deux Modes de Service

- Mode « sans connexion » (*connectionless*)
  - Point de vu informatique
  - Service minimum de la part du réseau
  - Envoi et réception de paquets sans garanties
- Mode « orienté connexion » (*connection oriented*)
  - Point de vu télécom
  - Service minimum dans les hôtes
  - Création de circuits avec garanties de services

Mode Sans Connexion

- Communauté *Internet*
  - 35 ans d'expérience des réseaux informatiques
  - le réseau fait le minimum :
    - Service sous-jacent **non-fiable**
    - **Hôte** gérant la **fiabilité** (détection et correction des erreurs) et le **contrôle de flux**
- -> Mode sans connexion
    - primitives : *SEND_PACKET* et *RECEIVE_PACKET*
    - pas de fonctionnalités qui se trouvent dans les hôtes (ré-ordonnancement, contrôle de flux...)

Mode Orienté Connexions

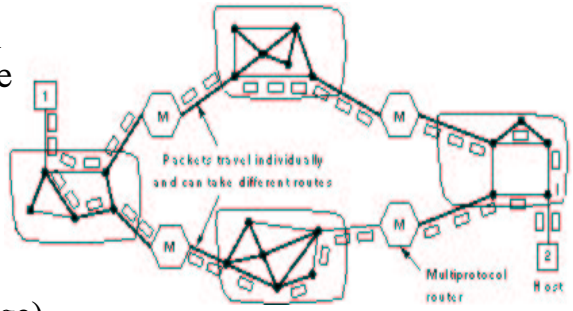
- Communauté Téléphonique
  - 100 ans d'expérience de réseaux **non-informatiques**
  - Le réseau fait le maximum :
    - Service sous-jacent **fiable**
- -> Mode Orienté Connexion
    - Création d'une connexion préalable, maintenue pendant toute la communication et explicitement terminé.
    - Négociation des paramètres de qualité de service à l'établissement
    - Communication bidirectionnelle et ordonnée
    - Contrôle de congestion systématique

Deux Techniques Support

- Approche basée sur les **Datagrammes**
  - permet de construire un service en mode non-connecté ou orienté connexion
  - généralement associé au **mode sans connexion**
- Approche basée sur les **Circuits Virtuels**
  - permet de construire un service en mode orienté connexion ou non
  - généralement associé au **mode orienté connexion**

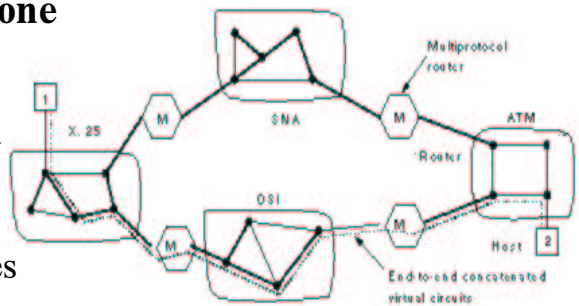
Approche Datagrammes

- Analogie avec les **télégrammes**
- Buts :
  - pas de mémorisation du trajet d'un type de paquet
  - possibilité de routes diverses
  - robustesse
  - efficacité (multiplexage)
  - tarification au débit



Approche Circuits Virtuels

- Analogie avec les circuit physiques du **téléphone**
- Buts :
  - éviter de recalculer la route pour chaque paquet
  - traitement efficace des paquets (temps réel possible)
  - tarification à la durée



Comparaison des Deux Techniques

Technologie	Datagramme	Circuit Virtuel
Initialisation	Non	Oui
Adressage	Adresse complete de la source et du destinataire	Identificateur de VC
Information d'Etat	Non	Oui pour chaque VC
Calcul de Route	Pour chaque paquets	Initial
Robustesse	Perte des paquete en cours de traitement en cas de crash d'un routeur	Perte de tous les circuits en cas de crash
Contrôle de Congestion	Complexe	Simple

Combinaison des Modes et Techniques de Transmission

Upper layer	Type of subnet: Datagram	Type of subnet: Virtual circuit
Connectionless	UDP over IP	UDP over IP over ATM
Connection-oriented	TCP over IP	ATM AAL1 over ATM

Plan - *Couche Réseau (1)*

- Présentation
- **Routage**
  - **Principes de base**
  - Trouver le plus court chemin
  - Inondation
  - Routage par vecteur de distance
  - Routage par état des liaisons
  - Routage *multicast*
  - Autres routages
  - Exercices
- *Internetworking*

Principe du Routage

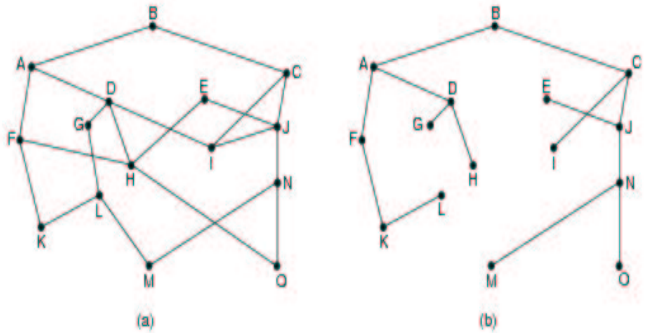
- Acheminement des paquets de la **source** à la **destination** quelque soit le nombre de saut
- Fonction principale de la Couche Réseau
- utilisation d'un **Algorithme de routage**
  - choix de l'interface de sortie
    - pour chaque paquet pour les datagrammes
    - pour le 1er paquet d'un circuits virtuels (*session routing)*

Propriétés des Algorithmes de Routage

- Exactitude (*correctness*)
- Simplicité (*simplicity*)
- Robustesse (*robustness*)
  - doit toujours fonctionner qqs les évenements
- Stabilité (*stability*)
  - convergence rapide vers un équilibre
- Équité (*fairness*)
  - même traitement pour tout les paquets
- « Optimalité » (*optimality*)

Principe d' « Optimalité »

- Si *B* est sur la route optimale de *A* à *C,* alors la route optimale de *B* à *C* suit la même route
- *conséquence : les routes optimales vers une destination donnée forme un arbre (*sink tree*)*
  - pas de boucle
  - nb de saut fini



Types d'Algorithme de Routage

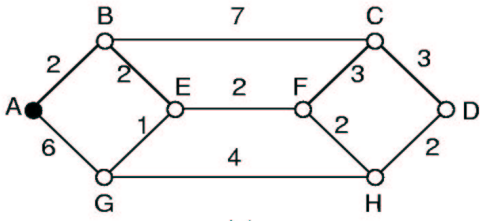
- Routage **statique** (non adaptatif)
  - décisions précalculées et chargées initialement dans le routeur
- Routage **dynamique** (adaptatif)
  - décisions de routage évoluant selon les modification de topologie et de trafic (information locale, de routeur adjacent ou globale)

Plan - *Couche Réseau (1)*

- Présentation
- **Routage**
  - Principes de base
  - **Trouver le plus court chemin**
  - Inondation
  - Routage par vecteur de distance
  - Routage par état des liaisons
  - Routage *multicast*
  - Autres routages
  - Exercices
- *Internetworking*

Routage par le Plus Court Chemin

- Algorithme de routage statique
- Représentation à l'aide d'un graphe :
  - routeur = **nœud, sommet**
  - liaisons = **arcs** (valeur = **métrique**)



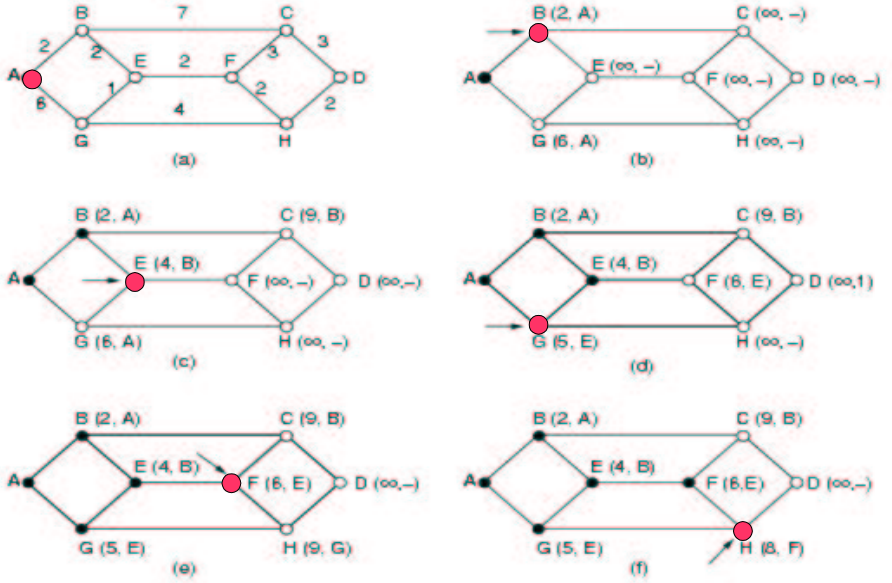
Routage par le Plus Court Chemin

- Utilisation de l'algorithme de SPF de **Dijkstra**
  - SPF = *Shortest Path First*
  - une étiquette pour chaque nœud, contenant la distance minimum ainsi que le nœud précédent
  - initialement le meilleur chemin n'est pas connu, la source (destination) est instaurée comme nœud initial
  - mise a jour progressive des étiquettes des nœuds à partir de la route à la distance minimum

Routage par le Plus Court Chemin

- Algorithme de SPF de **Dijkstra**
  - pour chaque sommet *s* du graphe faire :
    - d[*s*] = inf., pred[*s*] = NIL
  - d[*s-init*] = 0, **E** = vide, **R** = tous les sommets
  - tant que **R** n'est pas vide, faire :
    - soit *m* le sommet de **R** tq d[*m*] = min des d[*s*] de **R**
    - retirer *m* de **R** et l'inclure dans **E**
    - pour chaque sommet adjacent de *m* faire :
      - si d[*s*] > d[*m*] + métrique[*s,m*] alors
        - d[*s*] = d[*m*] + métrique[*s,m*]
        - pred[*s*] = *m*

Routage par le Plus Court Chemin



Plan - *Couche Réseau (1)*

- Présentation
- **Routage**
  - Principes de base
  - Trouver le plus court chemin
  - **Inondation**
  - Routage par vecteur de distance
  - Routage par état des liaisons
  - Routage *multicast*
  - Autres routages
  - Exercices
- *Internetworking*

Flooding

- Algorithme de routage statique
- Envoi vers toutes les interfaces sauf celle d'arrivée
- Simple mais coûteux
- Robuste et choix du meilleur chemin
- Limitation de la diffusion :
  - compteur
  - identifiant

Selective Flooding

- Idem *flooding*
- Sélection des interfaces (« bonne direction »)

Plan - *Couche Réseau (1)*

- Présentation
- **Routage**
  - Principes de base
  - Trouver le plus court chemin
  - Inondation
  - **Routage par vecteur de distance**
  - Routage par état des liaisons
  - Routage *multicast*
  - Autres routages
  - Exercices
- *Internetworking*

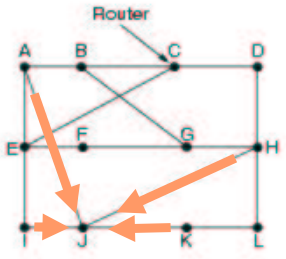
Distance Vector Routing

- Algorithme de routage dynamique
- Chaque routeur maintient une table (***vector***) de distances vers chaque destinataire possible avec l'interface à utiliser
- Algorithme de routage initial d'ARPANET puis d'Internet (RIP)
- Utilise l'algorithme de routage distribué de **Bellman-Ford** (ou algorithme de **Ford-Fulkerson)**

Algorithme de Bellman-Ford

- Le routeur connaît la distance à ses voisins
  - métrique = nb de saut, délais, queue...
- Échanges réguliers de vecteurs avec ses voisins
- Estimation des distances en calculant le minimum des vecteurs des voisins additionné à la distance à ceux-ci

Distance Vector Routing



(a)

To	A	I	H	K	New estimated delay from J	Line
A	0	24	20	21	8	A
B	12	36	31	28	20	A
C	25	18	19	36	28	I
D	40	27	8	24	20	H
E	14	7	30	22	17	I
F	23	20	19	40	30	I
G	18	31	6	31	18	H
H	17	20	0	19	12	H
I	21	0	14	22	10	I
J	9	11	7	10	0	—
K	24	22	22	0	6	K
L	29	33	9	9	15	K
	JA delay is 8	JI delay is 10	JH delay is 12	JK delay is 6	New routing table for J	

Vectors received from J's four neighbors

(b)

Problème du Comptage à l'Infini

- Convergence lente dans certain cas
  - exemple pour le nœud A :
    - (a) démarage
    - (b) **A** tombe en panne -> rebond (ping-pong)

A	B	C	D	E	
	∞	∞	∞	∞	Initially
	1	∞	∞	∞	After 1 exchange
	1	2	∞	∞	After 2 exchanges
	1	2	3	∞	After 3 exchanges
	1	2	3	4	After 4 exchanges

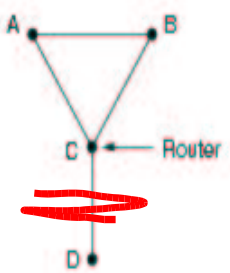
(a)

A	B	C	D	E	
	1	2	3	4	Initially
	3	2	3	4	After 1 exchange
	3	4	3	4	After 2 exchanges
	5	4	5	4	After 3 exchanges
	5	6	5	6	After 4 exchanges
	7	6	7	6	After 5 exchanges
	7	8	7	8	After 6 exchanges
	∞	∞	∞	∞	

(b)

Horizon Partagé

- Modification pour éviter le comptage à l'infini
  - Pour éviter les modifications du vecteur par les destinations provenant de l'émetteur des information
  - Astuce : suppression des métriques provenant du destinataire dans les vecteurs lui étant destinés
    - fonctionne avec l'exemple précédent
    - mais pas ici (bouclage)



Plan - *Couche Réseau (1)*

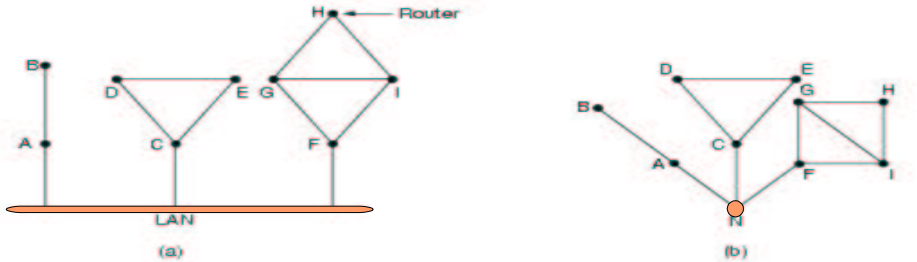
- Présentation
- **Routage**
  - Principes de base
  - Trouver le plus court chemin
  - Inondation
  - Routage par vecteur de distance
  - **Routage par état des liaisons**
  - Routage *multicast*
  - Autres routages
  - Exercices
- *Internetworking*

Link State Routing

- Algorithme de routage dynamique
- Fonctionnalités (par chaque routeur) :
  - découverte de ses voisins et de leurs adresses
  - mesure et calcul de coût vers ses voisins
  - construction de paquets avec les informations connues et envoi de celui-ci vers tous les routeurs
  - calcul du plus court chemin vers tous les routeurs
- Représentation complète du réseau dans tous les routeurs : calcul du plus court chemin avec SPF de **Dijkstra**

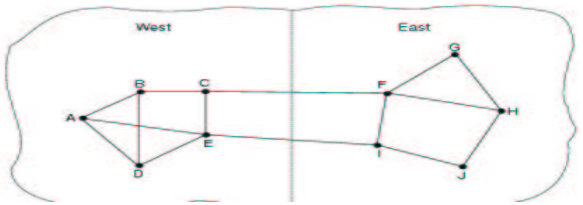
Information de Voisinage

- À l'initialisation, envoi d'un paquet ***HELLO***
  - vers toutes les interface point-à-point
  - considération des réseaux à diffusion comme un nœud :



Calcul du Métrique des Liaisons

- Calcul du temps d'aller-retour à l'aide de paquets ***ECHO***
  - ne pas tenir compte du temps de passage dans les files d'attente (routage selon la charge : risque d'oscillations)



Construction des Paquets d'État

- Avec les informations locales, ajout de son identité et envoi aux autres routeurs
  - périodiquement
  - sur événement

(a)

(b)

Distribution des Paquets d'État

- *Flooding* avec numéro de séquence
  - exemple, routeur **B**

			Send flags			ACK flags			
Source	Seq.	Age	A	C	F	A	C	F	Data
A	21	60	0	1	1	1	0	0	
F	21	60	1	1	0	0	0	1	
E	21	59	0	1	0	1	0	1	
C	20	60	1	0	1	0	1	0	
D	21	59	1	0	0	0	1	1	

Calcul des Nouvelles Routes

- Récupération d'une image complète du réseau par tous les routeurs
- Deux lignes par lien
- Exécution de l'Algorithme de **Dijkstra**

Plan - *Couche Réseau (1)*

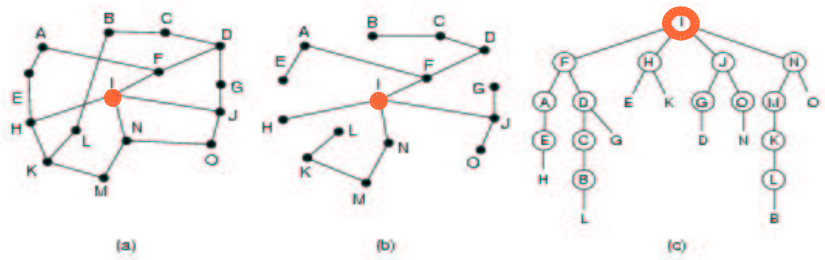
- Présentation
- **Routage**
  - Principes de base
  - Trouver le plus court chemin
  - Inondation
  - Routage par vecteur de distance
  - Routage par état des liaisons
  - **Routage *multicast***
  - Autres routages
  - Exercices
- *Internetworking*

Broadcast Routing

- 1) Envoi multiple
  - inefficace et nécessite la liste des destinataires
- 2) *Flooding*
- 3) Routage multi-destinataires
  - efficace mais avec une liste explicite des destinaires
- 4) Arbres couvrants (*spanning trees*)
  - efficace, pas de boucles mais mémorisation obligatoire
- 5) RPF (*Reverse Path Forwarding*)
  - arbre couvrant sans mémoire...

Reverse Path Forwarding

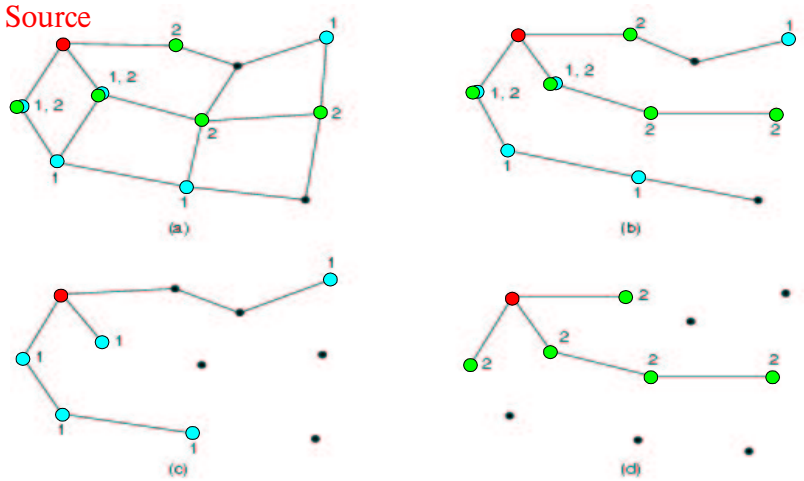
- Algorithme :
  - lorsqu'un paquet diffusé arrive dans un routeur, vérification qu'il arrive par l'interface utilisée pour joindre la source de celui-ci
    - si oui, alors probablement sur l'arbre couvrant et diffusion vers les interfaces de sorties
    - sinon élimination (paquet dupliqué)



Multicast Routing (1)

- Transmission vers un **groupe**
- ***Multicasting*** :
  - gestion de groupe
  - routage vers les membres du groupe
    - calcul d'un arbre couvrant spécifique
    - élagage (*pruning*)
      - exemple avec deux groupes...

Multicast Routing (2)



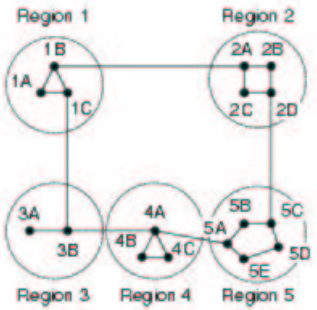
Multicast Routing (3)

- Deux approches :
  - vecteur de distance
    - algorithme RPF
    - envoi progressif de messages *PRUNE* des routeurs terminaux qui n'ont pas de membres
    - peu efficace pour les grands groupes (mémorisation)
  - état des liens
    - connaissance complète de la topologie
    - calcul direct des arbres de distribution
  - arbres centrés (*core based trees*)
    - utilisation d'un « point de rendez-vous »

Plan - *Couche Réseau (1)*

- Présentation
- **Routage**
  - Principes de base
  - Trouver le plus court chemin
  - Inondation
  - Routage par vecteur de distance
  - Routage par état des liaisons
  - Routage *multicast*
  - **Autres routages**
  - Exercices
- *Internetworking*

Hierarchical Routing



(a)

Full table for 1A

Dest.	Line	Hops
1A	–	–
1B	1B	1
1C	1C	1
2A	1B	2
2B	1B	3
2C	1B	3
2D	1B	4
3A	1C	3
3B	1C	2
4A	1C	3
4B	1C	4
4C	1C	4
5A	1C	4
5B	1C	5
5C	1B	5
5D	1C	6
5E	1C	5

(b)

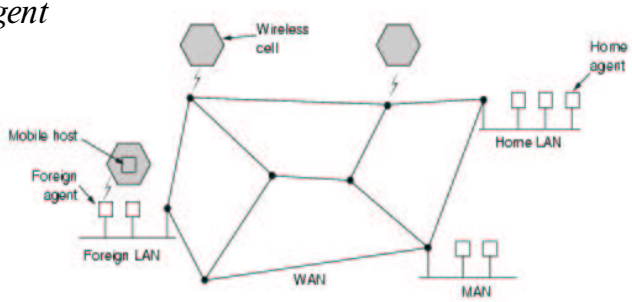
Hierarchical table for 1A

Dest.	Line	Hops
1A	–	–
1B	1B	1
1C	1C	1
2	1B	2
3	1C	2
4	1C	3
5	1C	4

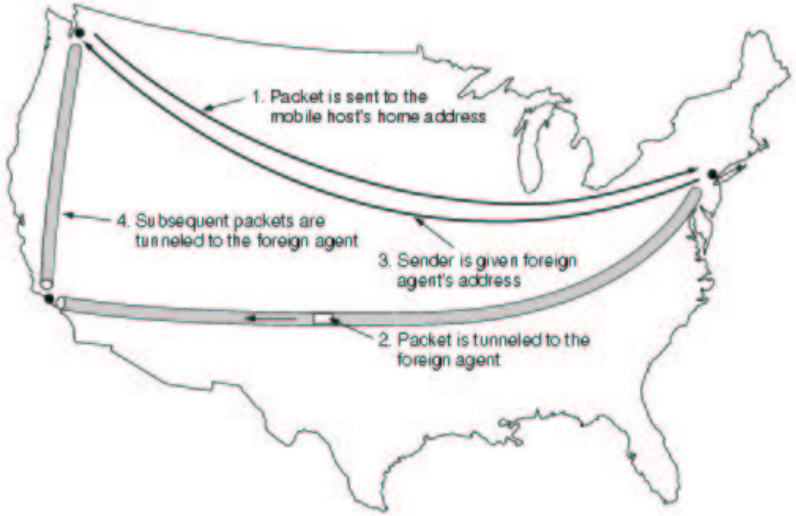
(c)

Mobile Routing (1)

- Utilisateur mobile (*home location* + *home agent*)
- Site distant (*foreign agent*)
  - diffusion info
  - enregistrement du mobile auprès du *foreign agent*
  - *foreign -> home agent*
  - identification
  - enregistrement



Mobile Routing (2)



QoS Routing

- ...

Plan - *Couche Réseau (1)*

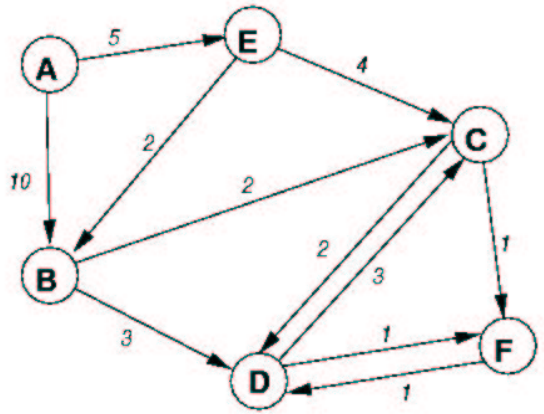
- Présentation
- **Routage**
  - Principes de base
  - Trouver le plus court chemin
  - Inondation
  - Routage par vecteur de distance
  - Routage par état des liaisons
  - Routage *multicast*
  - Autres routages
  - **Exercices**
- *Internetworking*

Exercice 1

- Soit le réseau composé de 4 nœuds (A, B, C et D) avec les liaisons pondérées suivantes : p[AB] = 2, p[AC] = 3, p[BC] = 2, p[CD] = 3 et p[BD] = 3. Les nœuds executent l'algorithme de **Ford-Fulkerson** et les liens sont symétriques.
  - Donnez le vecteur distance ainsi que la table de routage de chacun des nœuds.
  - La Liaison CD est rompue, montrez comment les vecteurs sont mis à jours avec la séquence d'échange suivante :
    - à t1 : D reçoit vB
    - à t2 : B reçoit vA, vC et vD
    - à t3 : C reçoit vA et vB
    - à t4 : A reçoit vB et vC

Exercice 2

- A partir du réseau suivant, construisez la table de routage de A à l'aide de l'algorithme SPF de **Dijkstra**



Plan - *Couche Réseau (1)*

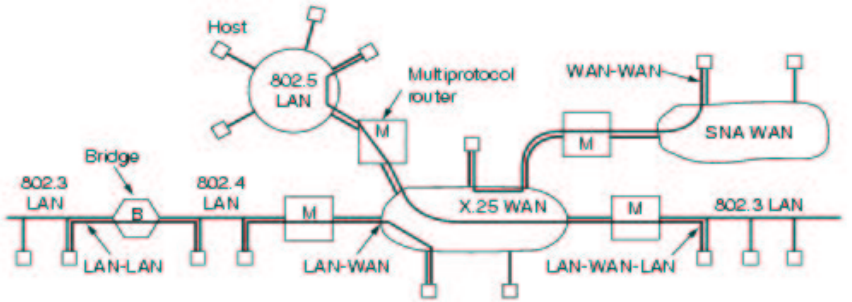
- Présentation
- Routage
- ***Internetworking***
  - **Bases**
  - Hétérogénéité
  - Tunnel
  - Fragmentation
  - Parefeu
  - Exercices

Principes de Bases

- Le monde des réseaux est **hétérogène** et doit **interfonctionner**
- Situation constante :
  - il y a toujours eut des technologies qui allaient dominer...

Types d'interconnexion

- LAN-LAN : inter-labo sur le campus
- LAN-WAN : accès à un super ordinateur distant
- WAN-WAN : méta-computing
- LAN-WAN-LAN : dicussion entre deux universités

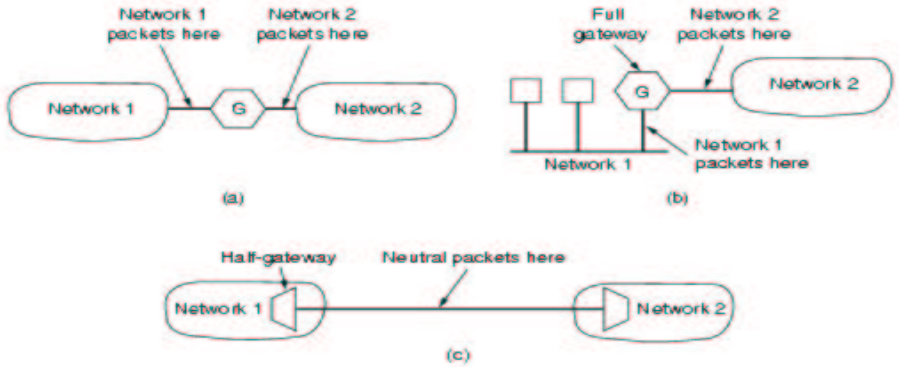


Niveaux d'Interconnexion

- Noms des éléments d'interconnexion :
  - Couche 1 : **répéteur** (*repeater*) [bits]
  - Couche 2 : **pont** (*bridge*) [trames]
  - Couche 3 : **routeur multiprotocole** (*router*) [paquets]
  - Couche 4 : **passerelle niveau transport** (*transport gateway*) [segments]
  - Couche 5 et + : **passerelles applicatives** (*application gateways*) [messages]

Passerelles

- ***Gateway*** pour tout élément reliant deux réseaux différents
  - Utilisation de zones neutres entre opérateurs :



Plan - *Couche Réseau (1)*

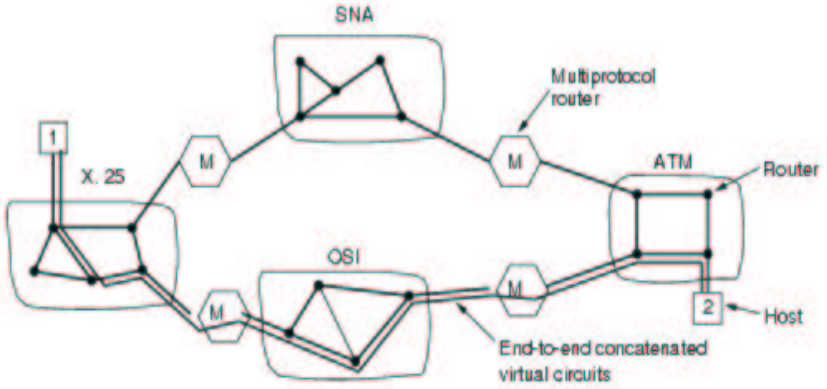
- Présentation
- Routage
- ***Internetworking***
  - Bases
  - **Hétérogénéité**
  - Tunnel
  - Fragmentation
  - Parefeu
  - Exercices

Différence dans les Réseaux

- **Service offert :** avec ou sans connexion
- **Protocoles :** IP, IPX, CLNP, Appletalk, DECnet...
- **Adressage :** hiérarchique (IP, ATM...) ou non (MAC)
- **Multicast :** présent, *broadcast,* émulé...
- **Taille Paquets :** maximum associé à chaque réseau
- **Qualité de Service :** plein de différentes, voir absente
- **Gestion des Erreurs :** fiabilité, ordonnancement...
- **Contrôle de Flux :** contrôle de débit, fenêtre glissante...
- **Contrôle de Congestion :** *leaky bucket, choke paquet...*
- **Sécurité :** cryptage, authentification...
- **Paramètres :** temporisations...
- **Facturation :** durée, quantité... absente !

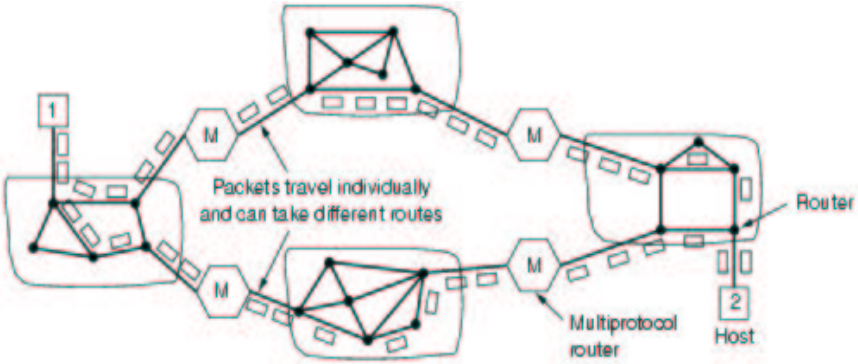
Association de Circuits Virtuels

- Interfonctionnement utilisant des concaténations de circuits virtuels :



Réseaux Sans Connexions

- Interfonctionnement utilisant plusieurs réseaux basés sur des datagrammes

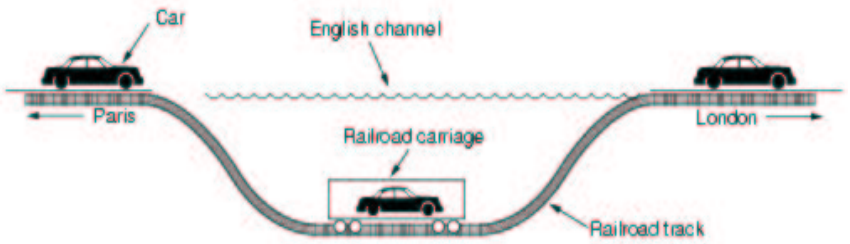


Plan - *Couche Réseau (1)*

- Présentation
- Routage
- ***Internetworking***
  - Bases
  - Hétérogénéité
  - **Tunnel**
  - Fragmentation
  - Parefeu
  - Exercices

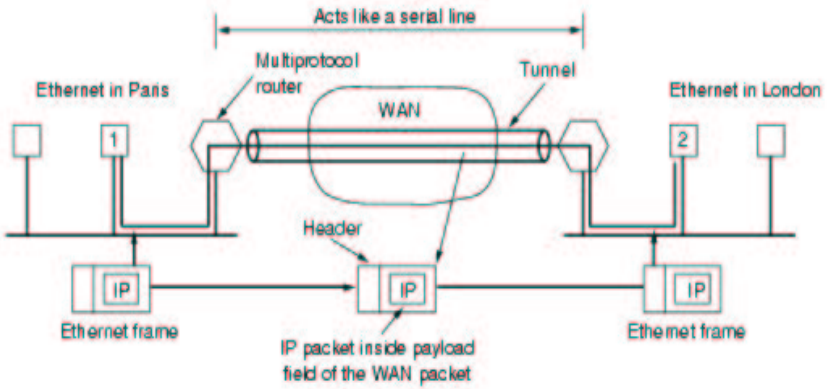
Principe du Tunnel

- Evitement possible de l'interfonctionnement lorsque la source et le destinataire utilise la même technologie : **encapsulation** (*tunneling*)



Exemple de Tunnel IP

- Conservation du paquet IP de la source à la destination, même si l'on passe sur d'autres technologies réseaux :



Plan - *Couche Réseau (1)*

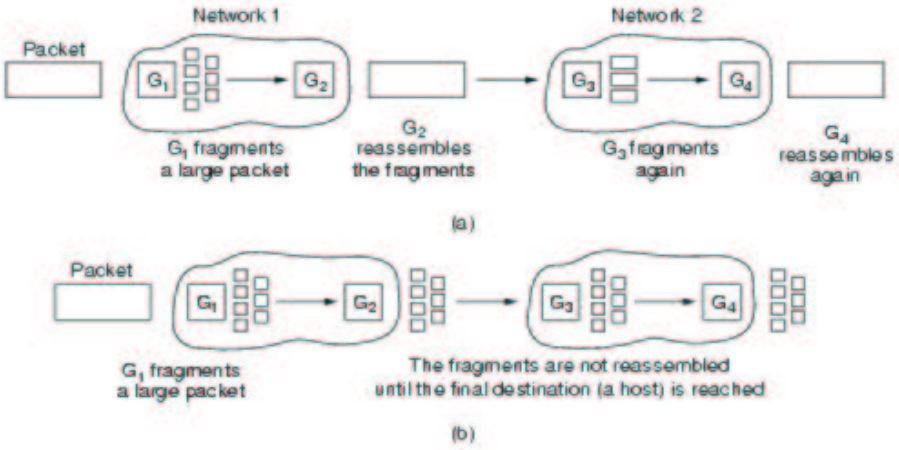
- Présentation
- Routage
- ***Internetworking***
  - Bases
  - Hétérogénéité
  - Tunnel
  - **Fragmentation**
  - Parefeu
  - Exercices

Limitation de la Taille des Paquets

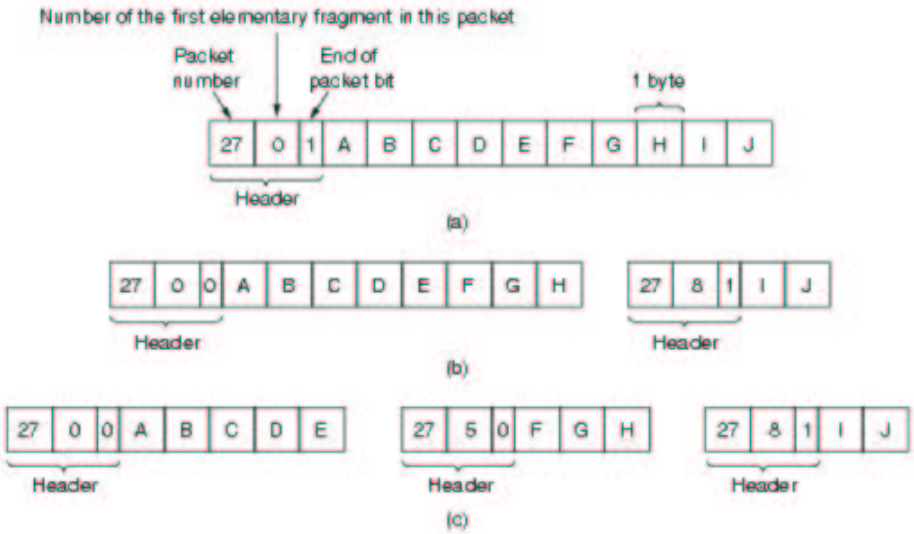
- Chaque réseau impose ses contraintes en terme de taille maximum de paquets
  - limitations liées :
    - au matériel ou à l'OS
    - au protocole (champs longueur du paquet)
    - réduction des erreur, de la congestion (retransmission) et de l'occupation du canal
  - taille utile (***payload***) de 48-octets (ATM) à 65,515-octets (IP)

Type de Fragmentation

- Fragmentation transparente (a) ou non (b)



Numérotation des Fragments

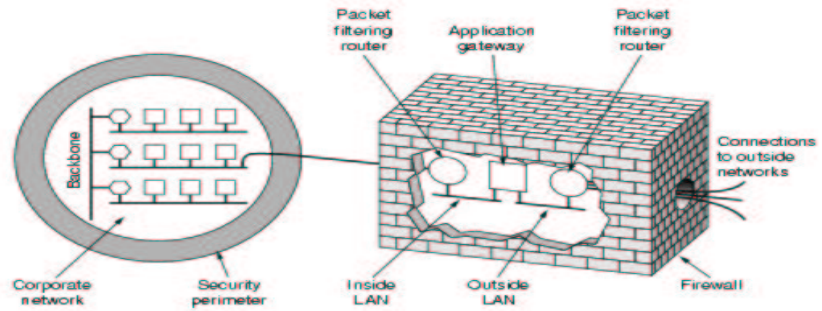


Plan - *Couche Réseau (1)*

- Présentation
- Routage
- ***Internetworking***
  - Bases
  - Hétérogénéité
  - Tunnel
  - Fragmentation
  - **Parefeu**
  - Exercices

Principe du Parefeu

- Besoin de filtrer les informations entrantes **et** sortantes :
  - Architecture en château fort
    - Seul accès : le pont-levi (Parefeu, Garde Barrière, ***Firewall***)



Réalisation de Parefeu

- Généralement deux routeurs filtrants (***Packet Filters***) et une passerelle applicative
- *Packet Filter*
  - filtre les adresses sources ou destinations
  - sous Unix, les numéros de port (correspondant à des services)
- *Application Gateway*
  - inspection de la sémantique des données...

Plan - *Couche Réseau (2)*

- Présentation
- Routage
- ***Internetworking***
  - Bases
  - Hétérogénéité
  - Tunnel
  - Fragmentation
  - Parefeu
  - **Exercices**

Exercice 3

- Un message de 2200-octets est passé dans *Internetwork* composé des réseau suivant :
  - (1) Ethernet (1500-octets max.)
  - (2) FDDI (4096-octets max.)
  - (3) RadioXDV (600-octets max.)
  - (4) Ethernet (1500-octets max.)
- Avec une fragmentation non transparente qu'en résulte-t-il sur la numérotation des fragments ??

Exercice 4

- L'utilisation de **tunnels** en mode connecté est évidente, il suffit de créer un circuit virtuel du routeur associé à l'entrée à celui de la sortie puis d'y faire passer les paquets dedans.
- Peut on aussi utiliser les **tunnels** en mode sans connexion ???