Cours Réseaux et télécoms L3/Info + Miage

Internet : comment ça marche ?

Enseignants :

Claude Jard : CM, TD, TP
Charles Bouvet : CM (Orange)
Salima Hamma : TD, TP
Marie Pelleau : TD
Benjamin Martin : TP
Thomas Vincent : TP

Une vue "système" des réseaux

Des applications vers le réseau : quelques protocoles clés de l'Internet

- *Cours 0 : état, enjeux*
- *Cours 1 : les applications pratiques de l'Internet*
- *Cours 2 : la couche de transport, étude de TCP*
- *Cours 3 : la couche réseau et le routage*
- *Cours 4 : la couche liaison et l'accès au canal de transmission*
- *Cours 5 : boucle locale, ADSL et Wifi : vision opérateur*
- *Cours 6 : la sécurité dans les réseaux*

Objectifs :

- Comprendre les mécanismes de base sous-jacents aux applications communicantes
- Connaître les enjeux et les difficultés

Cours 0 : le contexte …

- le Web comme une gigantesque base de données construite collectivement
- une myriade d'objets communicants (entre eux et avec les personnes)
- des défis techniques pour l'autogestion des infrastructures et l'adaptation des protocoles
- des défis sociétaux : nouveaux usages, sécurité, vie privée, …

Pourquoi l'Internet grandit ?

"Connectivity is its own reward" [Rutkowski, Internet Society]

- Effet de famille : un réseau de messagerie est d'autant plus utile qu'on peut y joindre d'avantage d'utilisateurs. L'attraction détermine le volume des nouveaux entrants (dérivée proportionnelle à la taille -> croissance exponentielle de l'effet boule de neige)
- Effet de marché : plus il y a d'utilisateurs, plus il y a de services offerts

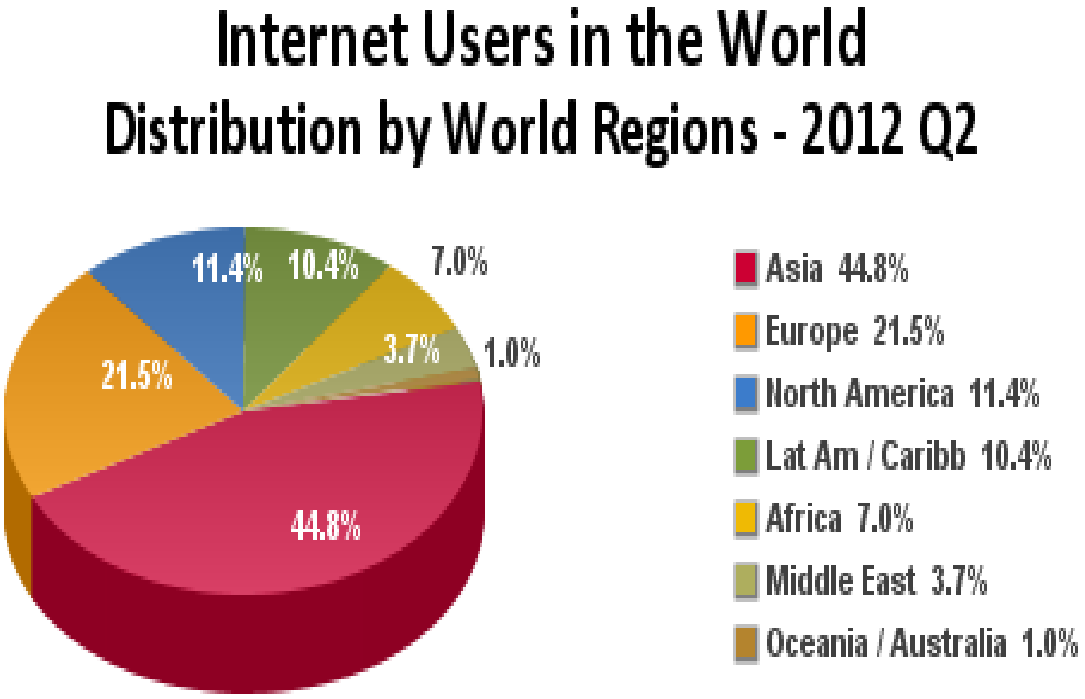
Pourquoi l'Internet grandit ?

"Connectivity is its own reward" [Rutkowski, Internet Society]

- Choix techniques : les architectes se sont attachés à construire le réseau le plus solidement possible, afin de garantir qu'il puisse croître

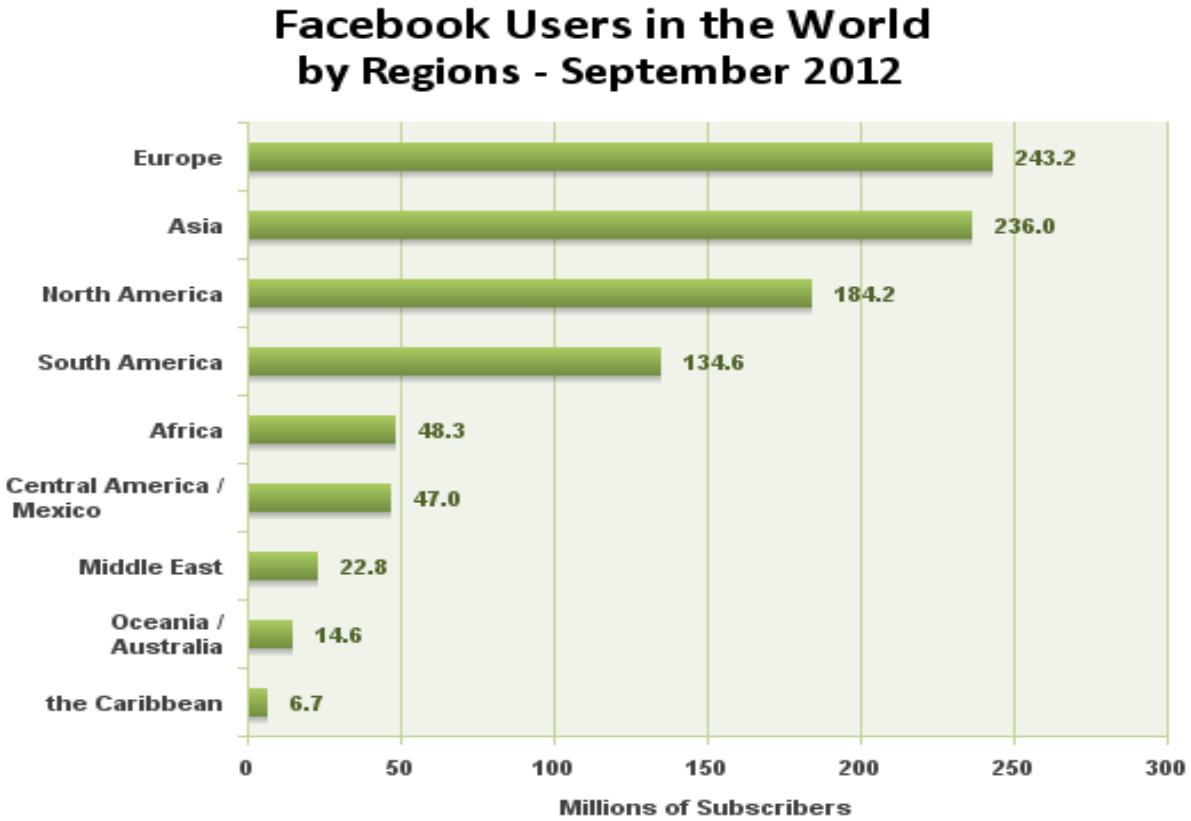
  - Élimination de tout point central
  - Les routeurs sont les briques de base (ordinateurs spécialisés chargés de relayer les messages de proche en proche)
  - Le calcul des chemins les plus courts est réalisé de façon répartie en fonction des informations disponibles chez les voisins

Internet dans le monde



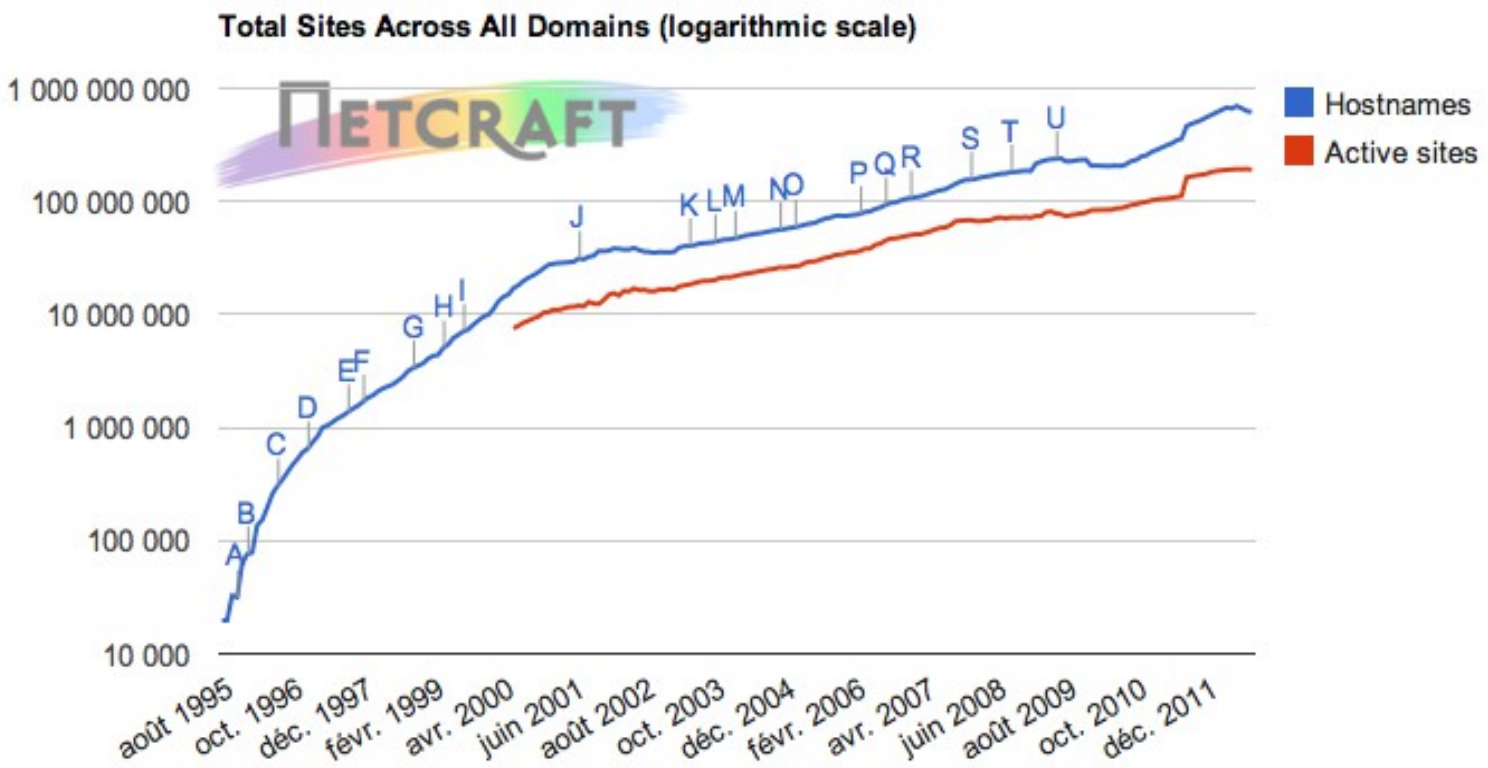
Source: Internet World Stats - www.internetworldstats.com/stats.htm
Basis: 2,405,518,376 Internet users on June 30, 2012

Facebook dans le monde



Source: Internet World Stats - www.internetworldstats.com/facebook.htm
The total number of Facebook subscribers in the world is estimated to be 937,407,180 on September 30, 2012.

Les sites Web dans le monde



Quelques chiffres

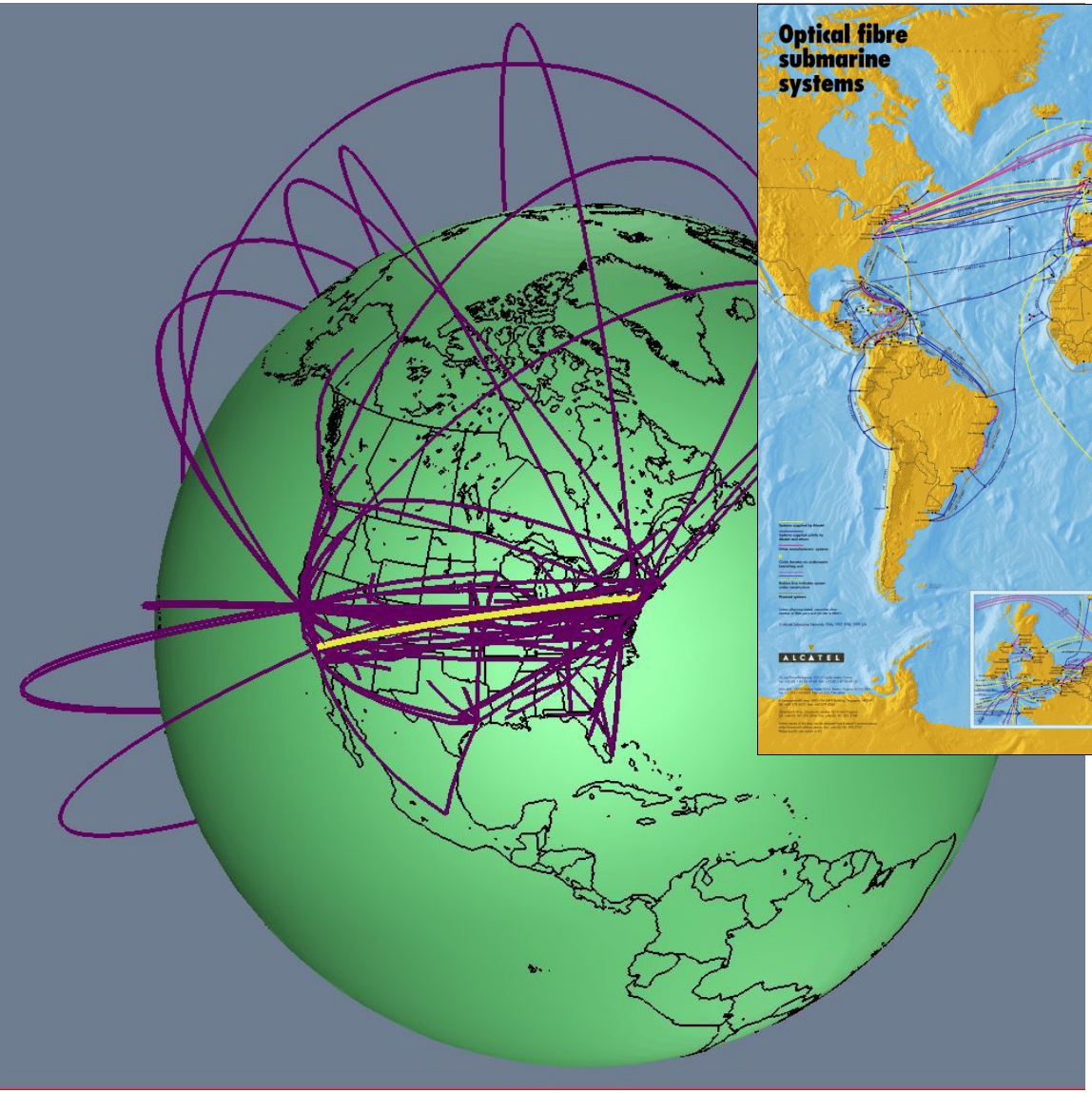
- Google est le site le plus visité au monde (gogol=10^{100}), 40000 requêtes par seconde
- 30000 milliards de documents indexés
- Plus d'un petaoctets (10^{15} octets) nécessaires pour stocker le cache des pages Web référencées (de l'ordre du nombre de grains de sable de la plage de la Baule)
- Google posséderait plus de 2 millions d'ordinateurs dans plus de 100 fermes de serveurs autour de la planète. La puissance de calcul requise pour l'indexation dépasse désormais celle pour le calcul scientifique…
- Avènement du « green computing »

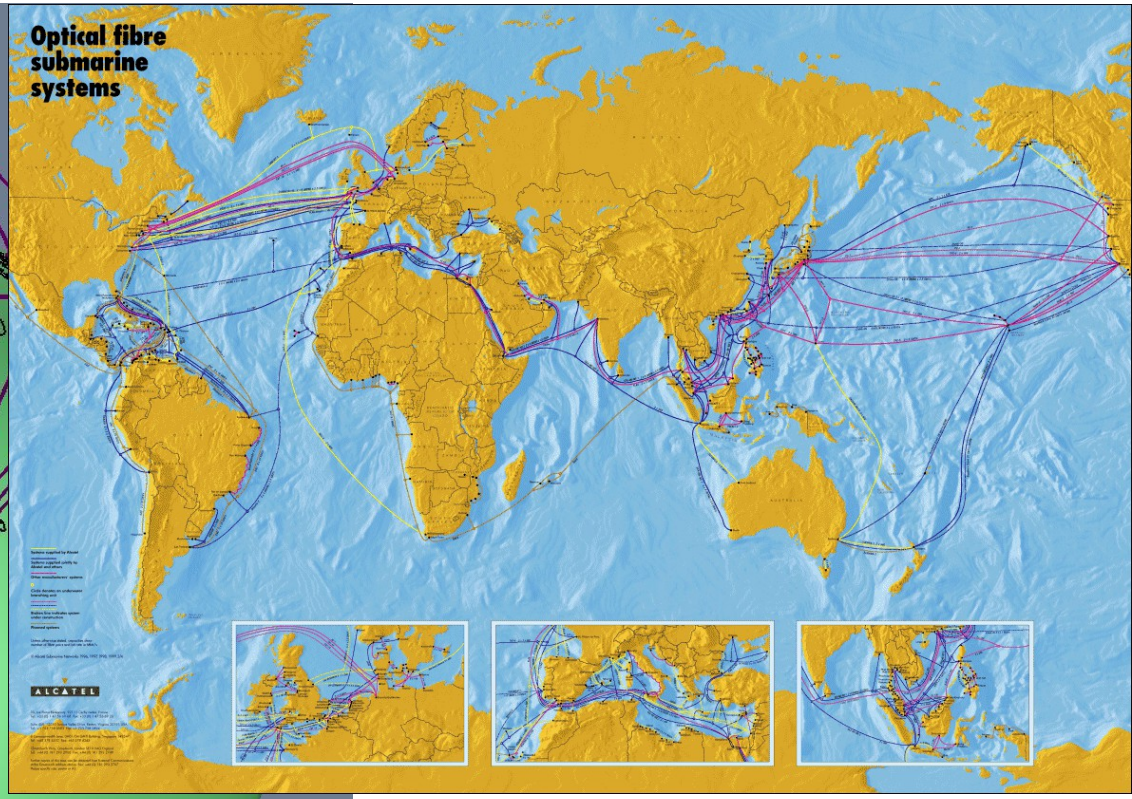
Quelques chiffres

- La population d'internautes dans le monde a dépassé 2,5 milliards en 2012 (1,5 milliard en 2011)
- Pénétration : Etats-Unis, Océanie, Europe, Amérique du sud, Moyen-orient, Asie, Afrique
- 1 milliard de noms de domaines
- Nombre moyen de pages Web par site : quelques centaines
- Ipcalyspe actuel…
- Vertige : Cerveau humain : 100 milliards de neurones, 10^{15} connexions. Vitesse influx nerveux : 100m/s

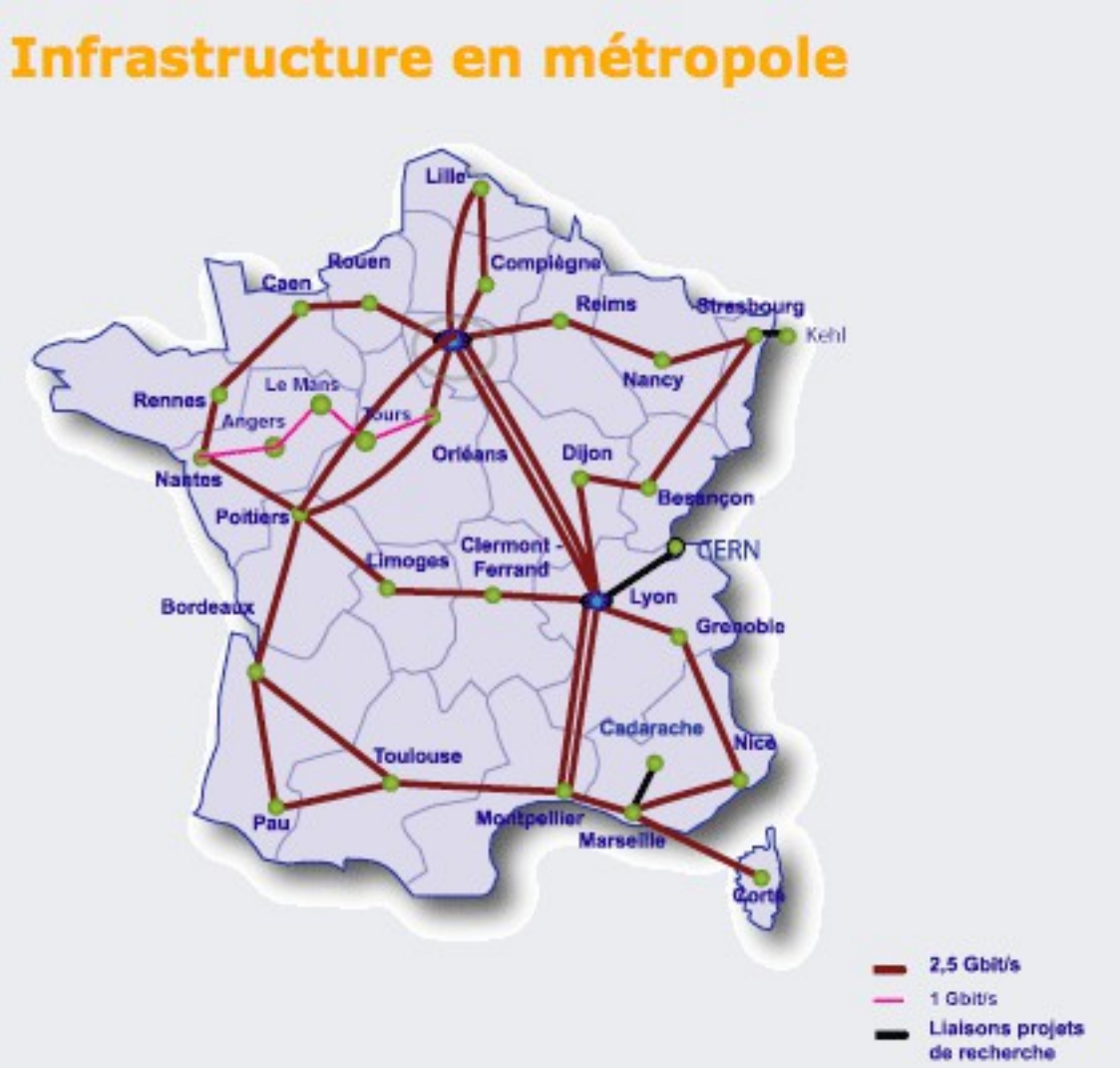
Les cartes de l'Internet : un nouveau territoire à découvrir…

Structure physique de l'Internet :
l'épine dorsale "Mbone"



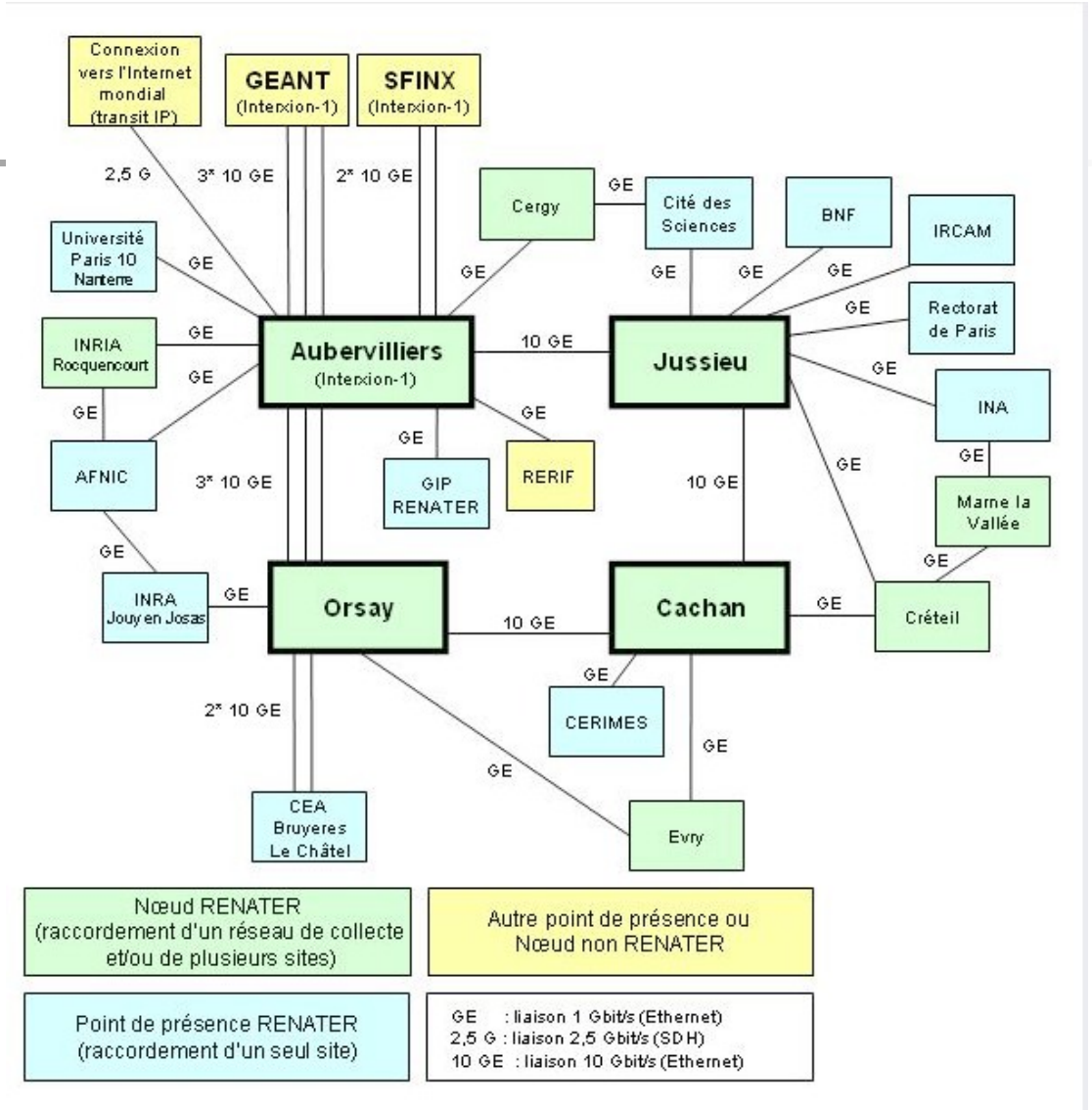


Le réseau de la recherche en France (RENATER)

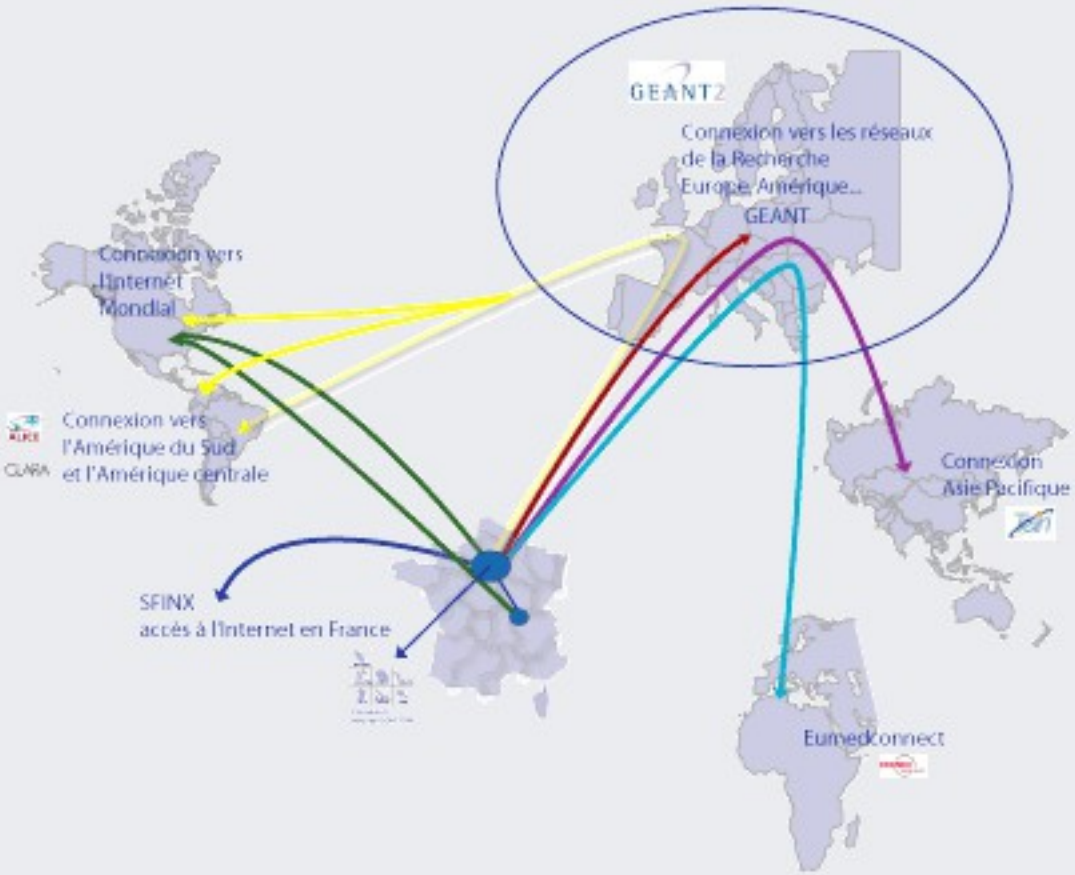


Les entreprises qui contribuent à l'évolution du réseau sont NeufCegetel pour la fourniture de l'ensemble du réseau de production national métropolitain (à l'exception de l'Ile de France et de la Corse), France Telecom pour la liaison avec la Corse, CISCO pour les équipements de routage et de commutation, Cegetel, Level3 et neuf telecom pour le réseau optique dédié aux grands projets de recherche et ALCATEL pour les équipements optiques. C'est Communication & Systèmes qui assure le déploiement, la gestion et l'administration de la totalité du réseau RENATER-4.

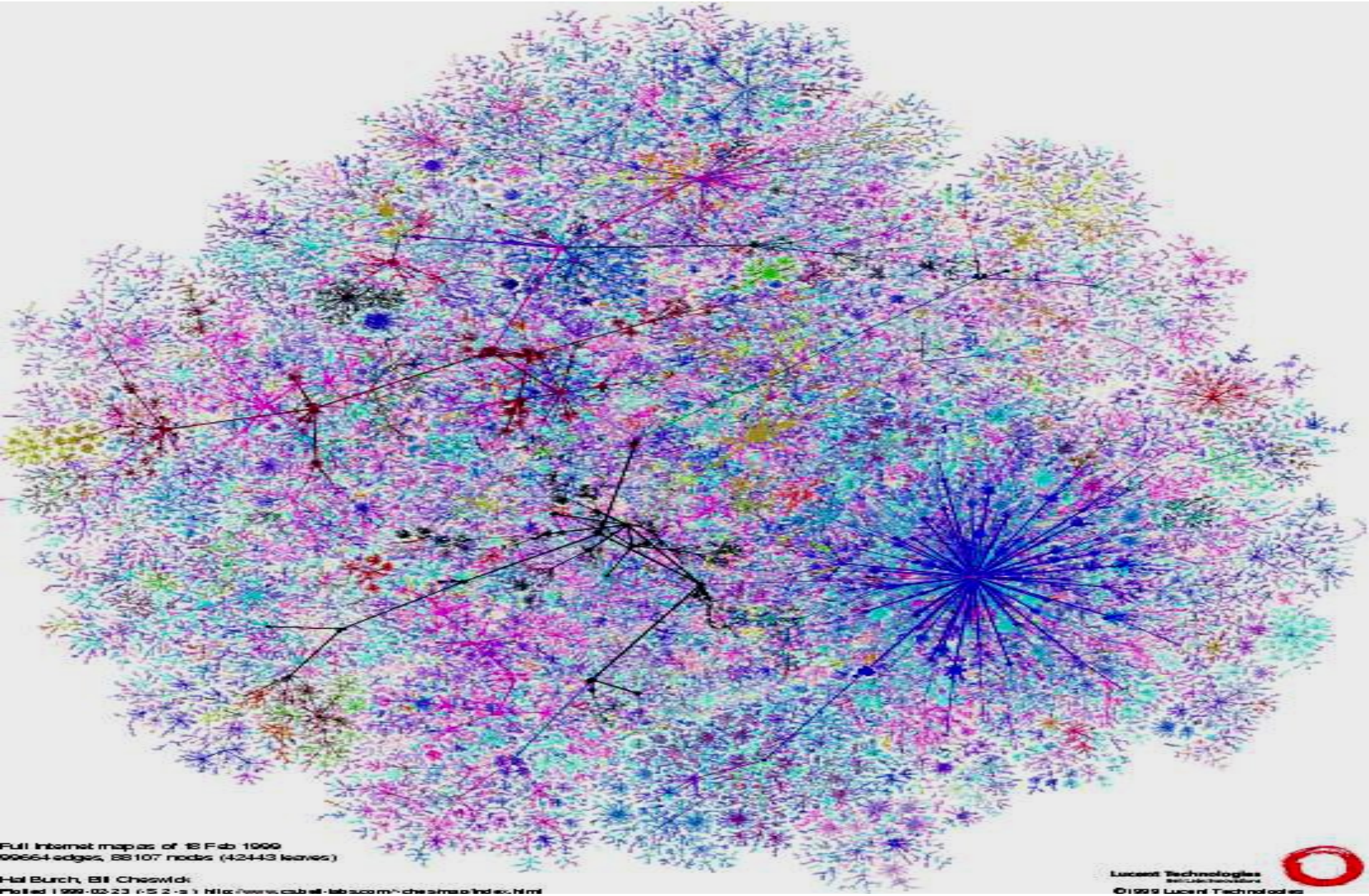
Le réseau de la recherche en Ile de France



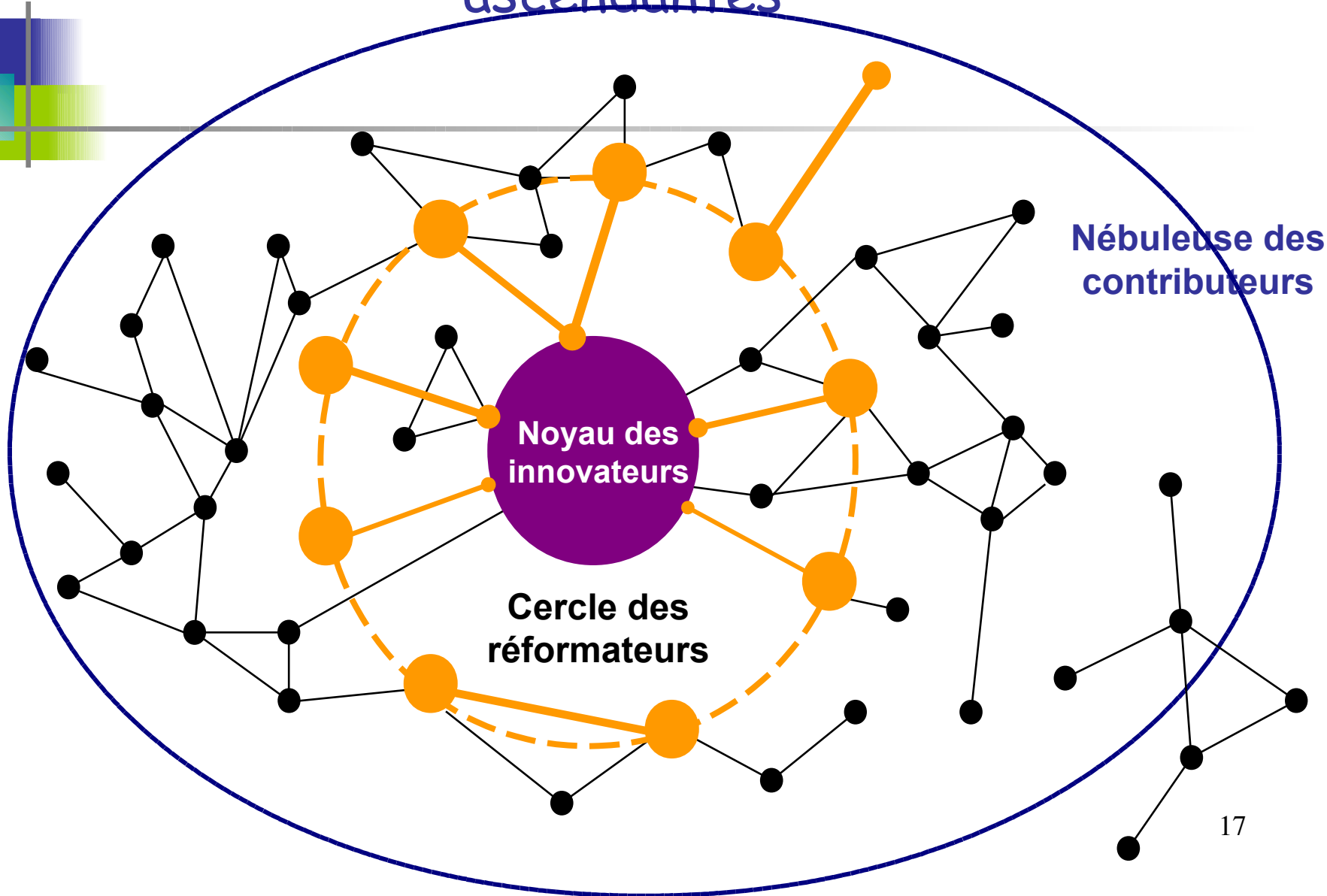
Interconnexion vers les réseaux de recherche dans le monde

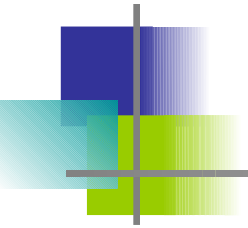


L'interconnexion Internet



L'organisation des grands collectifs sur Internet : trois cercles des innovations ascendantes

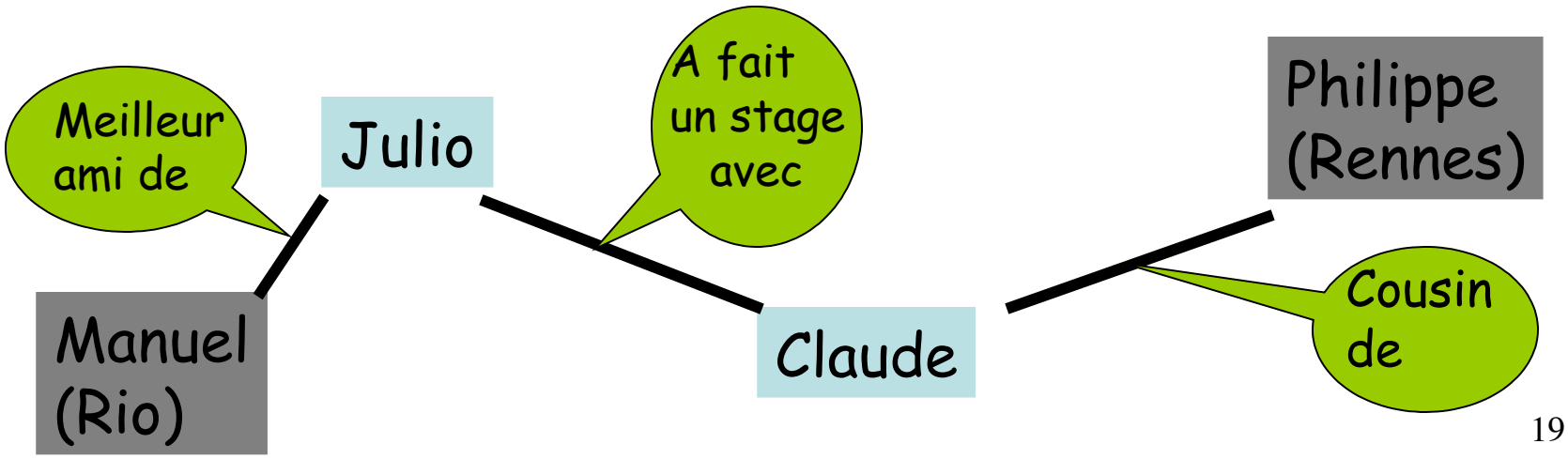




La surprise des graphes petits mondes …

Le monde est petit…

- Expérience du psychologue Stanley Milgram en 1967
- 300 personnes sélectionnées aléatoirement doivent faire parvenir une lettre à une personne donnée de Boston
- Règle : on ne peut envoyer une lettre qu'à quelqu'un que l'on connaît



Résultats

- 20% des lettres sont arrivées
- Elles sont arrivées en 6 étapes max seulement !

Caractéristiques des petits mondes :

1. Des chemins courts existent,
2. Les gens sont capables de les trouver

Les messages se déplacent sur un réseau social

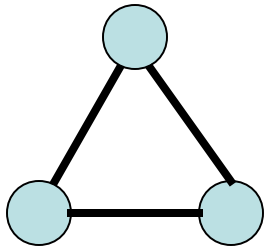
Grande connexité des petits mondes

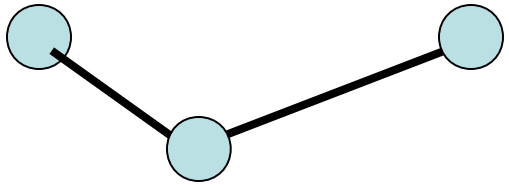
Distance moyenne entre 2 noeuds

Réseau	Nombre de noeuds	Nombre de liens	Distance moyenne (L) (<< #noeuds)
Co-signature des articles en biologie	1 520 251	11 803 064	4.9
Le Web Altavista (2000)	203 549 046	2 130 000 000	16.1

Phénomène de groupes

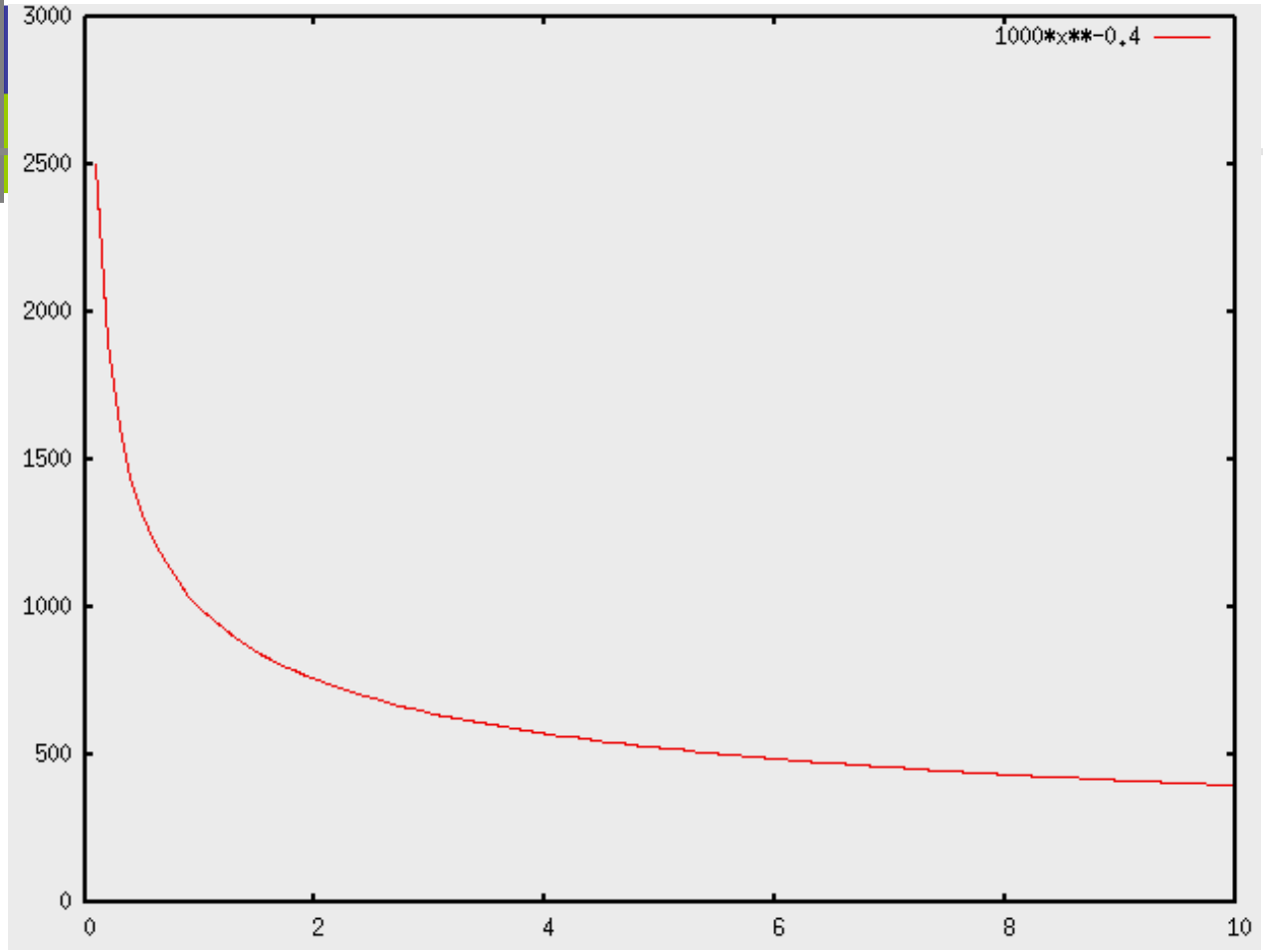
- "Les amis de mes amis sont mes amis" : beaucoup de triangles
- Mesure possible de connectivité ("Clustering") :
  C = 3 #triangles / #triplets connectés

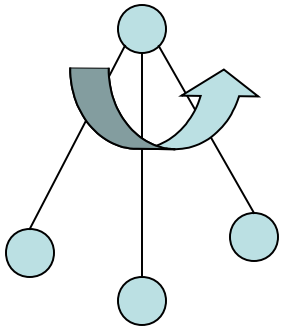




Parties du graphe de l'Internet : C = 0.11
(sur un graphe aléatoire pur : 0.0001)

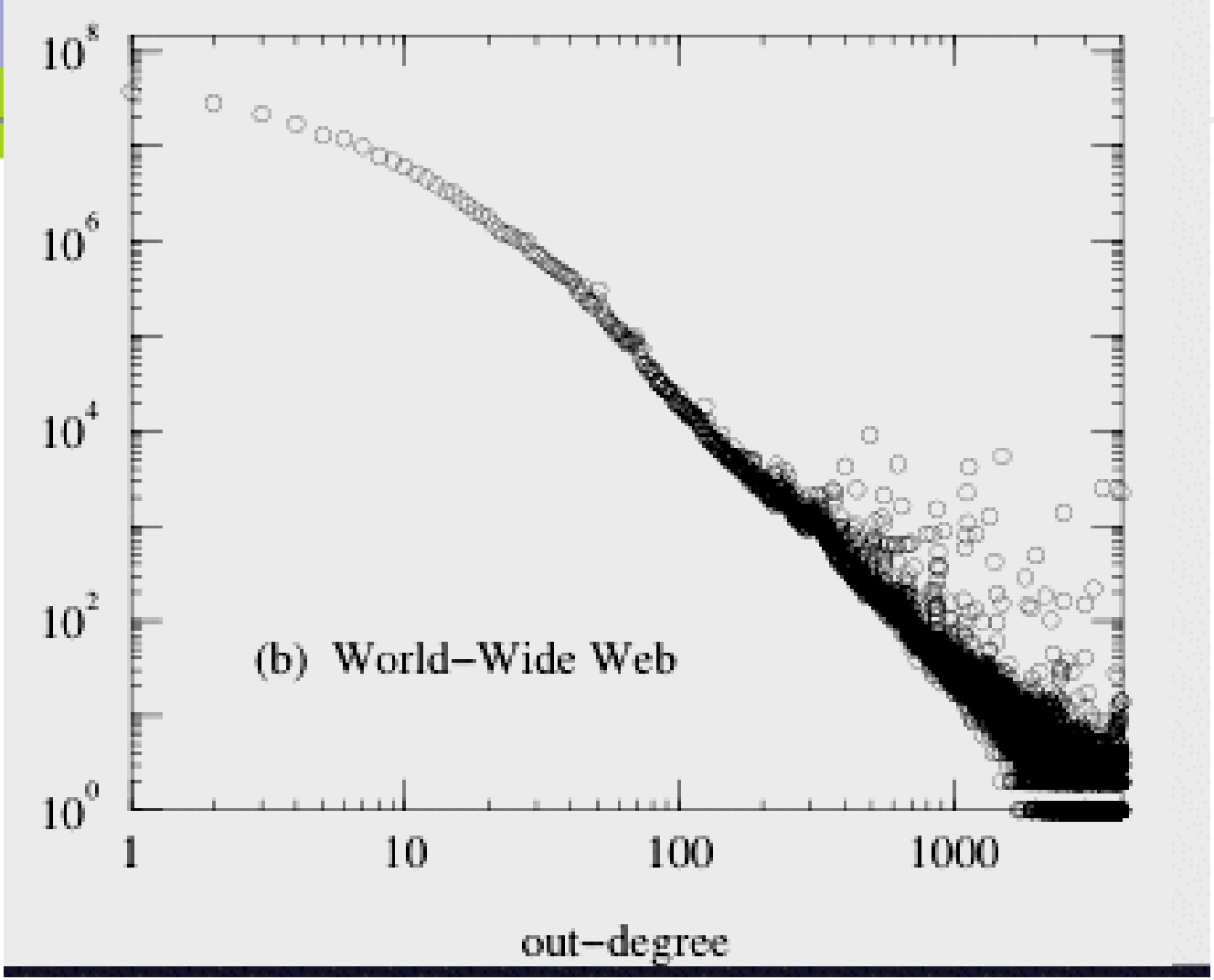
La distribution des degrés





- Loi de "puissance" : le nombre de noeuds de degré k est en k^{-a}

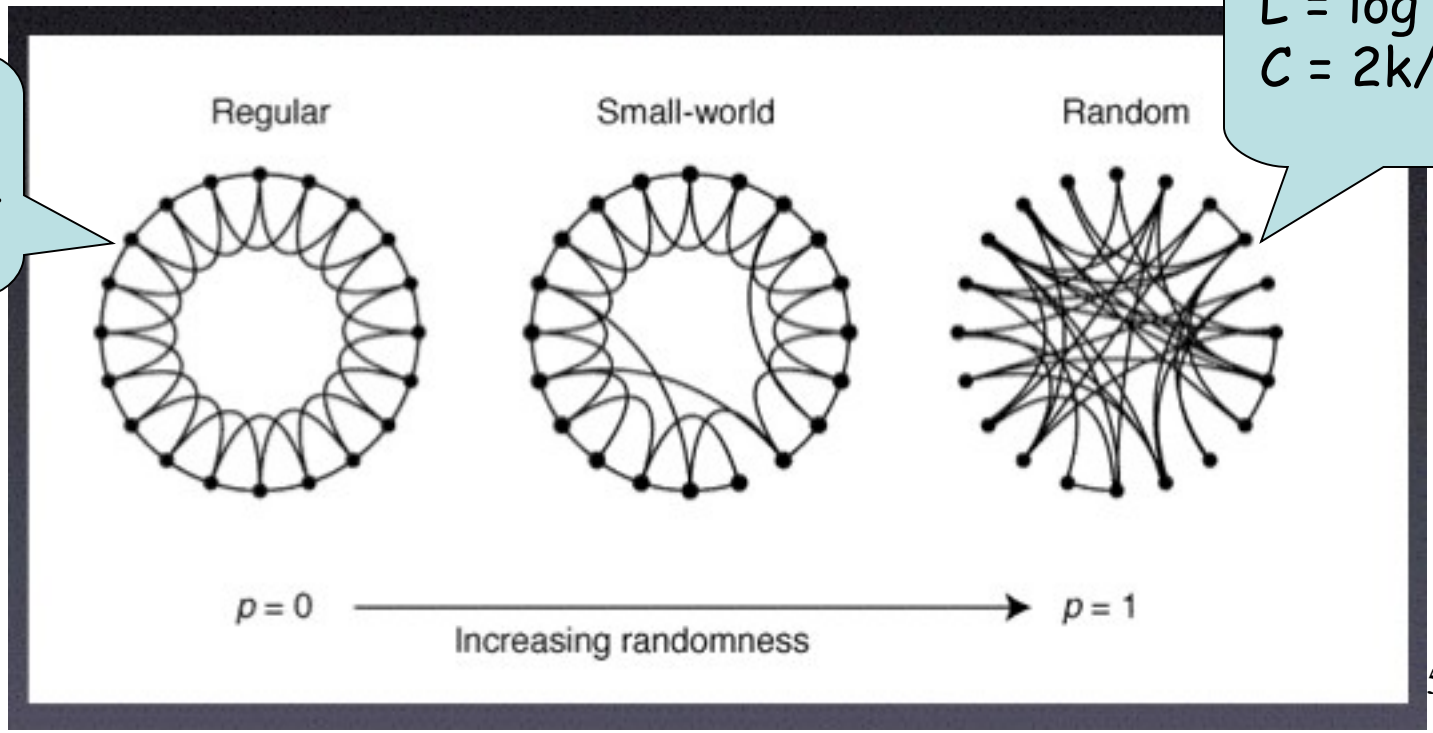
Un sous-graphe du Web
[Broder et al. 2000]



Modèle de Watts et Strogatz [Nature 1998]

1. Un anneau de n noeuds
2. Chaque noeud est en plus connecté à ses k voisins les plus proches
3. Recablage aléatoire

L = n/k
C = 3/4



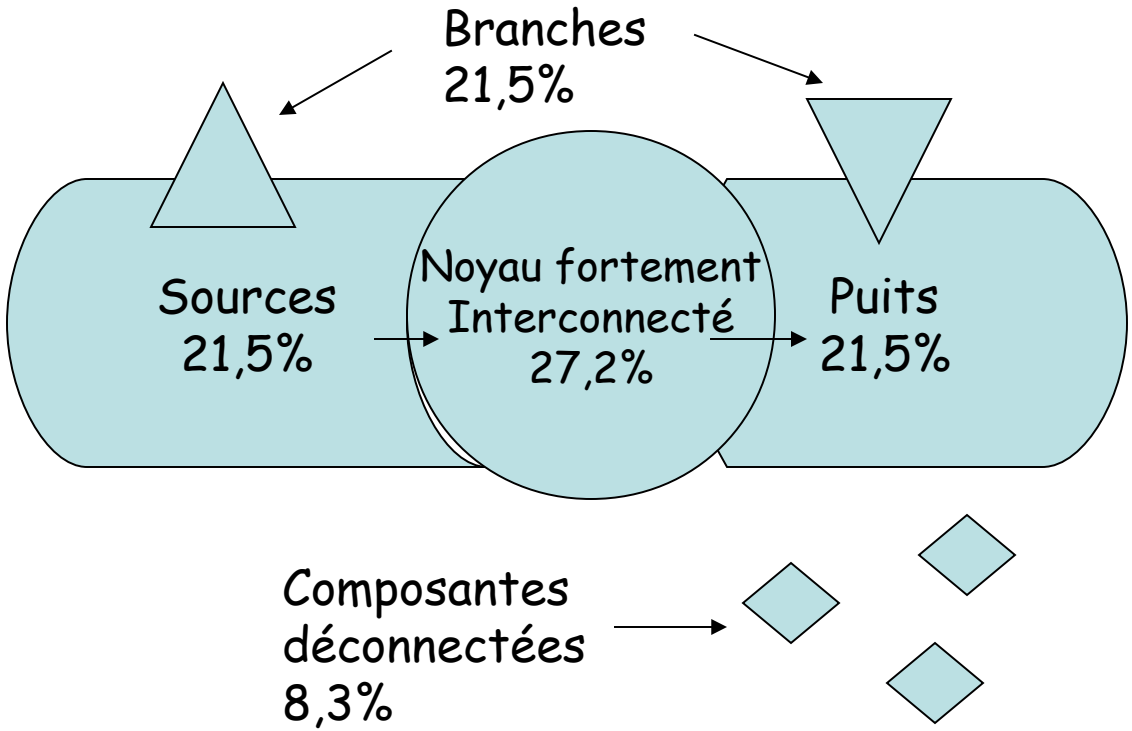
L = log n/log k
C = 2k/n

Modèle dynamique

- Le modèle de Watts et Strogatz ne retrouve pas la loi de puissance des degrés
- Plus de réalisme peut être obtenu par la simulation de la croissance du réseau (modèle dynamique)
- Stratégie de "l'attachement préférentiel" [Albert et Barabasi 1999] :
  - Le réseau croît continuellement,
  - Les nouveaux noeuds s'attachent préférentiellement aux sites déjà bien connectés ("on ne prête qu'aux riches")
- Le phénomène de groupe reste un peu trop faible…

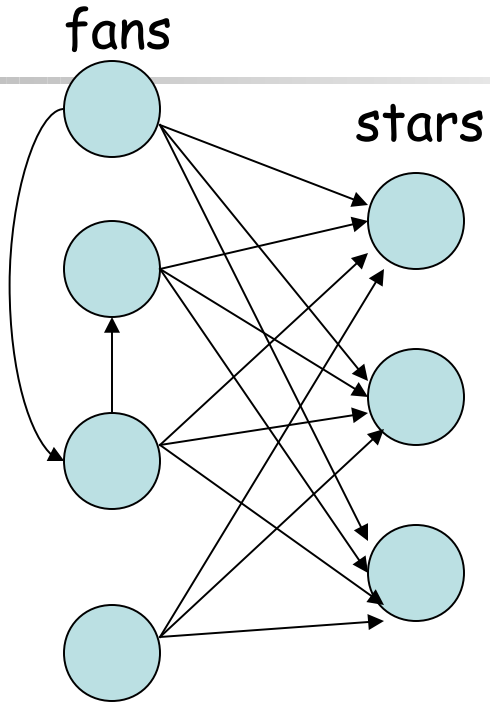
A quoi ressemble le graphe du Web ?

- De façon macroscopique : c'est un "nœud papillon". Avant cette découverte, on pensait pouvoir aller de n'importe quelle page à n'importe quelle autre. En fait, à partir du cœur, on n'accède qu'à la moitié du Web. Les liens peuvent aussi être très longs (900 clicks).



A quoi ressemble le graphe du Web ?

- Des structures microscopiques :

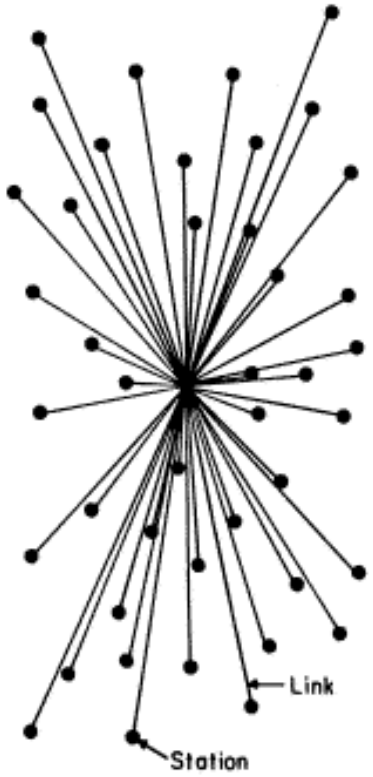


- Une page contient en moyenne 11 liens
- La probabilité qu'une page contienne *i* liens est proportionnelle à $i^{-2.1}$ (5 fois de moins de pages avec 9 liens qu'avec 1)

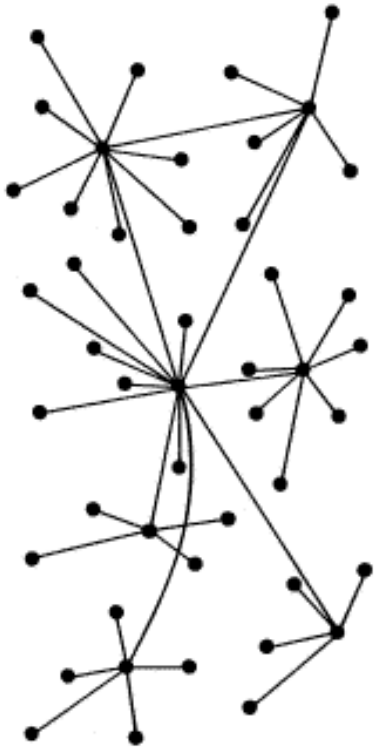
L'histoire…

Au commencement...

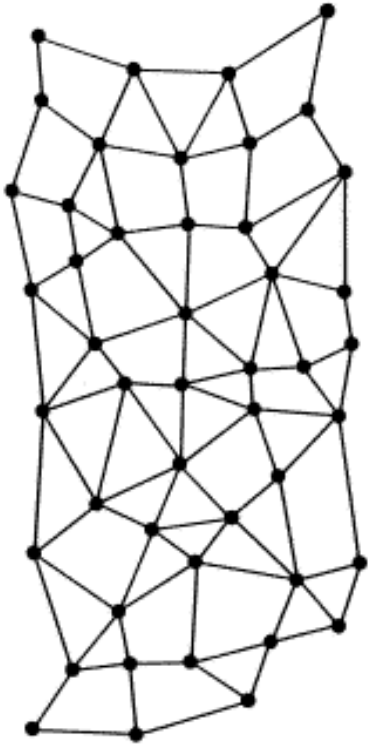
- En 1957, l'URSS est la première des deux super-puissances à envoyer un satellite artificiel dans l'espace : c'est le fameux Spoutnik. Traumatisés, les états-Unis forment au sein du Département de la défense un groupe appelé ARPA ("Advanced Research Projects Agency"), constitué de scientifiques, chargé de concevoir des innovations technologiques appliquées à l'armée.
- En 1962, l'US Air Force demande à un groupe de chercheurs de RAND (de "Research ANd Development", association non lucrative visant à développer les sciences et l'éducation aux états-Unis) de concevoir un réseau capable résister à une frappe nucléaire massive, afin de pouvoir riposter à son tour.
- La solution est un système décentralisé, qui permet au réseau de continuer à fonctionner même si une ou plusieurs machines est touchée. L'idée de décentralisation est due à Paul Baran. Plus précisément, c'est lui qui pensa à un système où chaque machine, maillon d'un réseau en toile d'araignée, chercherait, à l'aide de paquets de données dynamiques, la route la plus courte possible d'elle-même à une autre machine, et où elle patienterait en cas de "bouchons". Le projet de Paul Baran est refusé par les militaires et ce n'est que 6 ans plus tard qu'il se concrétise.



CENTRALIZED
(A)



DECENTRALIZED
(B)

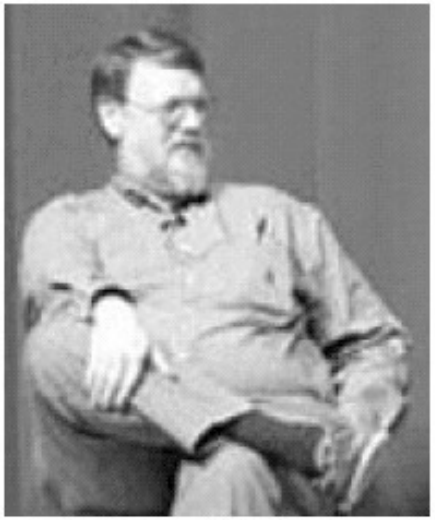


DISTRIBUTED
(C)

FIG. 1 — Centralized, Decentralized and Distributed Networks

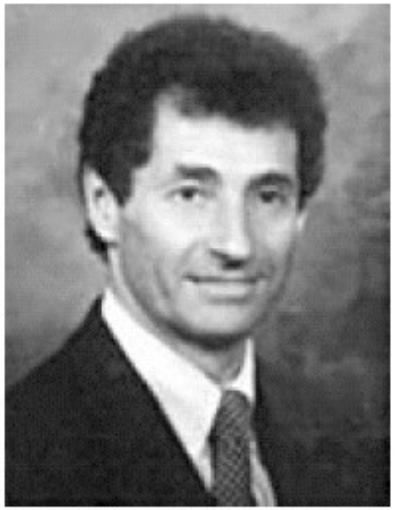
Premières briques…
1969 : ARPANET

- L'Arpanet, un réseau décentralisé se met en place sur commande de l'ARPA à BBN (Bolt Beranek and Newman Inc., une SSII de Cambridge, Mass.). Il comprend quatre grands centres universitaires américains :
  - UCLA (Université de Californie à Los Angeles)
  - SRI (Institut de recherche de Stanford)
  - UCSB (Université de Californie à Santa Barbara)
  - l'Université de l'Utah
- Ces quatre centres étaient reliés par des câbles 50Kbps, et utilisaient le NCP ("Network Control Protocol").
- La date conventionnelle pour la "naissance d'Internet", c'est la date de publication de la première RFC ("Request For Comments of the Internet Engineering Task Force"), le 7 avril 1969.



Ray Tomlinson, écoutant

Le labo de Ray Tomlinson, son mauvais éclairage et ses machines bruyantes



Le professeur Kleinrock

UCLA

Premières briques…
1971 : le courrier électronique

- 1971 : le courrier électronique ("killer application"). C'est Ray Tomlinson, de BBN, qui en est l'inventeur. A l'époque, Tomlinson travaille sur un système permettant à un utilisateur d'une machine de laisser un message à un autre utilisateur de la même machine (Post-It sur l'écran). En même temps, il teste un logiciel de transfert de fichiers via l'Arpanet. C'est en réunissant les deux concepts qu'il invente le courrier Électronique. C'est également lui qui choisit l'arrobe (le fameux glyphe "@") comme séparateur pour les adresses électroniques.

- Anecdote : que contenait le premier courrier électronique jamais envoyé ? Le premier message télégraphique de Samuel Morse était "What hath God wrought?", le premier message téléphonique d'Alexander Bell, "Mr. Watson, come here; I want you". Ray Tomlinson ne s'en souvient plus bien, mais il pense que c'était : "QWERTYUIOP", la première rangée de lettres d'un clavier qwerty…

Premières briques…
1973 : TCP/IP

- En 1973 se développe se que l'on appellera plus tard le protocole TCP/IP, l'une des pierres d'angle de l'Internet actuel, sous la houlette de Vinton Cerf, de Stanford, et de Robert Kahn, de la DARPA (nouveau nom de l'ARPA), et de Louis Pouzin (INRIA).
- Ce sont ces hommes qui, en 1974, parlèrent pour la première fois "d'Internet". Le protocole TCP/IP sera adopté par le Département de la défense pour l'Arpanet en 1976.

L. Pouzin

R. Kahn

V. Cerf

Premières briques…
1980 : le cas du Minitel français

- La direction générale des télécommunications vient de confier à un jeune polytechnicien, Jacques Dondoux, la responsabilité de lancer le Minitel, un terminal familial de réservation de billets de train dans la gare… de Rennes.
- Dès 1981, la France entière découvre les "autoroutes de l'information", l'indicatif 3615, et les prénoms féminins à consonance scandinave

Jacques Dondoux, un temps ministre

Premières briques…
1983 : le DNS

- Au début de l'Arpanet, les informations nécessaires à la connexion des machines entre elles (conversion nom <-> adresse) sont contenues dans un fichier nommé "hosts.txt". Ce fichier est maintenu par le NIC ("Network Information Center") de l'Institut de recherche de Stanford. Chaque administrateur d'une machine reliée à l'Arpanet doit envoyer ses modifications au NIC qui les centralise et redistribue périodiquement le hosts.txt mis à jour. Au fur et à mesure que l'Arpanet se développe, le système devient trop lourd à gérer.
- En 1983, pour résoudre ce problème, un groupe constitué de Jon Postel, Paul Mockapetris et Craig Partrige rédige les RFC 882 et 883 : le DNS ("Domain Name System") est inventé. C'est une base de données distribuée qui permet une gestion locale des noms de domaine, tout en rendant l'information disponible à tous. La base de données est divisée en zones. Pour chaque zone, un ou plusieurs serveurs de noms ("name servers" en anglais) répond aux requêtes des résolveurs. Les résolveurs permettent de faire le lien entre nom d'une machine et une adresse IP.
- En 1984 se mettent en place les "top level domains", c'est-à-dire les suffixes comme .com, .gov, .net ou encore .org.

Le boom…
1989 : le World Wide Web

- C'est avec la naissance du Web qu'Internet s'étend au grand public. On fait généralement remonter la date de naissance du Web au texte "Information Management: A Proposal" de T. Berners-Lee en mars 1989, alors chercheur au CERN de Genève.
- Berners-Lee est l'inventeur du premier serveur Web, qu'il appelle "httpd", et du premier client Web, qu'il appelle "WWW". Tim Berners-Lee travaille au MIT et a dirigé le W3C, un consortium chargé de mettre au point les standards du Web. Aujourd'hui le W3C est co-dirigé en plus par l'INRIA et KEIO (Japon).
- C'est en mars 1993 qu'est inventé Mosaic, le premier des navigateurs grand public, doté d'une interface graphique. Son auteur est Marc Andreessen, étudiant à l'université de l'Illinois, et assistant au NCSA ("National Center for Supercomputing Applications"). La première version de Mosaic est pour Unix, mais rapidement sortent des versions pour Windows et Mac OS. Mosaic connaît un succés et immédiat. C'est le premier navigateur a avoir reconnu la balise IMG, autorisant ainsi l'emploi d'images sur des pages Web…

Le boom…
1995-2003 : la guerre des navigateurs

- Marc Andreessen a continué de faire parler de lui en créant, en 1994, Netscape, un navigateur qui supplante rapidement Mosaic et règne en maître sur le marché des navigateurs, jusqu'en 1995, quand Microsoft lance Windows 95 et son propre navigateur, Internet Explorer. Menacé par ce puissant rival, Netscape, qui en est à Netscape 4, lance en 1998 le groupe Mozilla (d'aprés le nom de code de Netscape Communicator). Le groupe Mozilla est chargé de produire un navigateur libre et gratuit, que Netscape pourrait récupérer ensuite. Mais le groupe Mozilla décide de tout réécrire de fond en comble et tarde énormément à rendre un produit fini.
- Quand le groupe Mozilla sort enfin Mozilla 1.0 en 2002, Netscape (racheté entre temps par AOL) peut lancer Netscape 7, mais c'est déjà trop tard, Internet Explorer détient plus de 90% du marché. En 2003, AOL prend acte de son échec et cesse de développer Netscape. Aujourd'hui, Mozilla et ses dérivés sont des navigateurs de choix pour tous ceux qui n'utilisent pas Windows. Quant aux autres, il ne tient qu'à eux d'essayer.

2004- : l'explosion des réseaux sociaux

- Facebook, Youtube, …
- Les bibliothèques numériques sont au cœur de l'Internet
- Web 2.0.
- Services mobiles

<u>Aujourd'hui :</u>

- L'Internet des objets
- Le Web social
- Cloud computing. Virtualisation ->

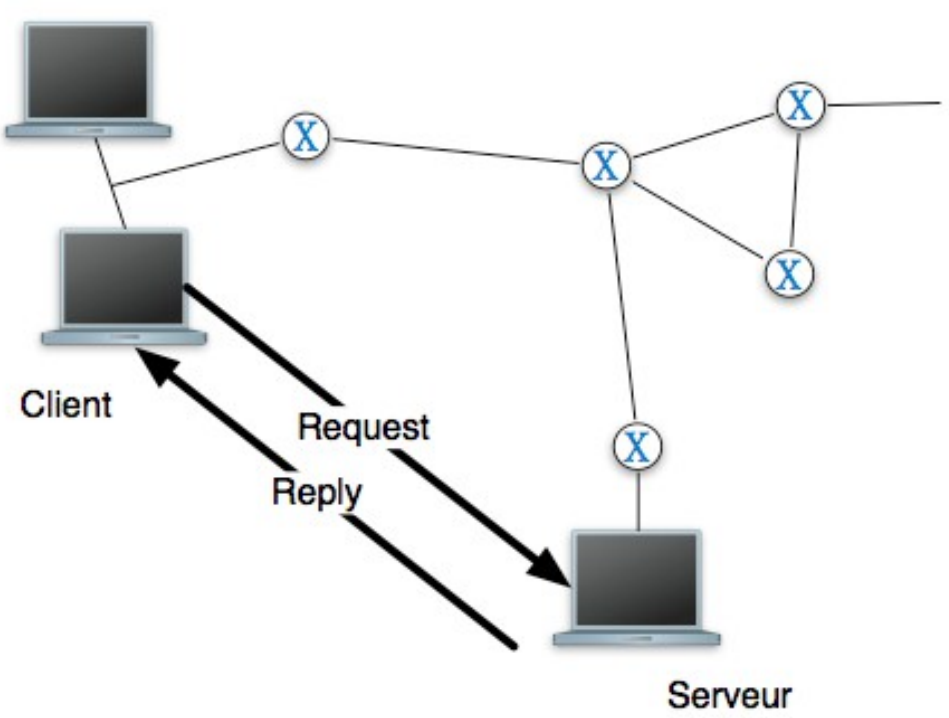
Le monde devient numérique…

Cours 1 : des applications de l'Internet

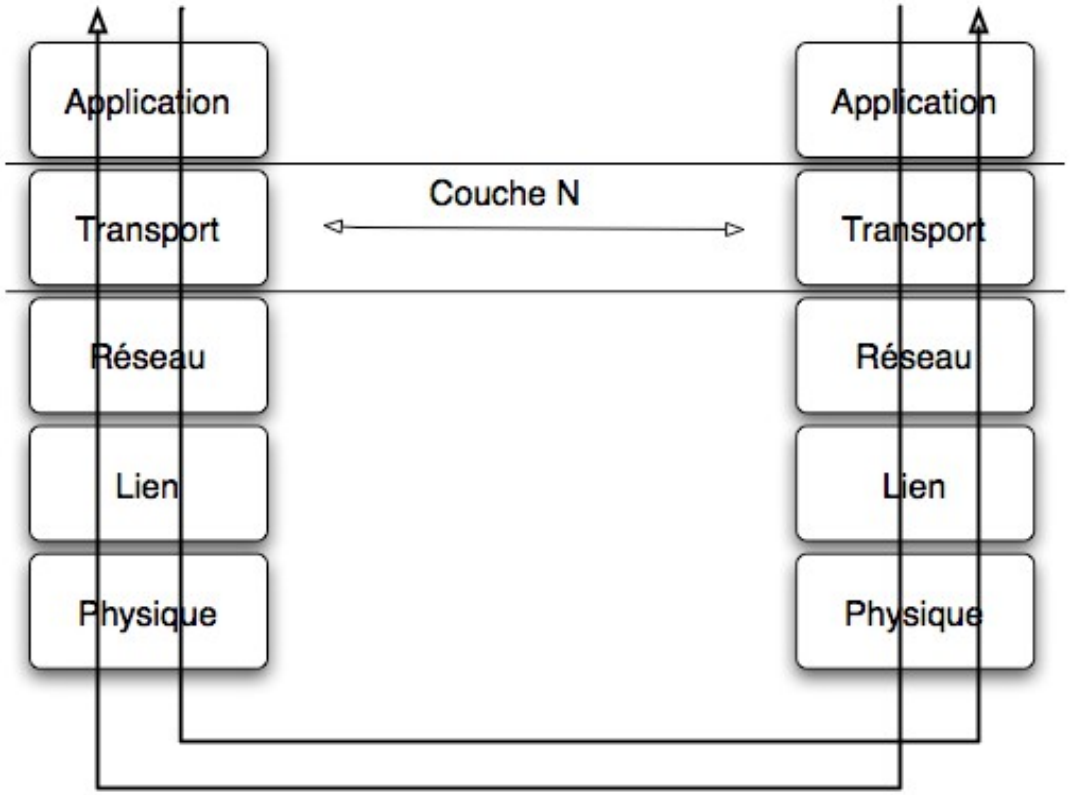
- 1.1 Le protocole HTTP et le Web
- 1.2 La messagerie
- 1.3 DNS
- 1.4 Cache Web
- 1.5 Services Web
- 1.6 Outil de validation SPIN

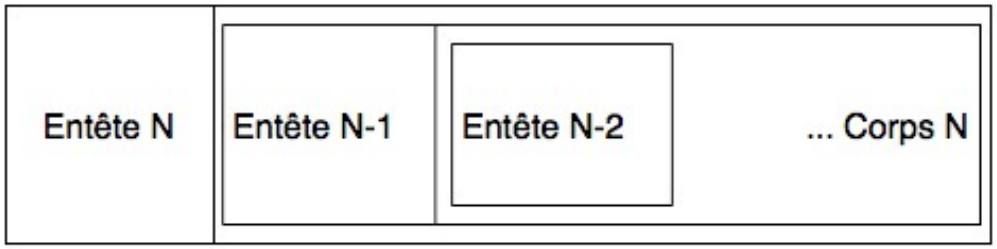
1.1 Le protocole HTTP et le Web
Client-serveur

- "HyperText Transfer Protocol" au coeur du Web actuellement
- Interaction client-serveur à travers un réseau : protocole de bout-en-bout. Ex. serveur Web Apache pour Unix. Ex. client Web brouteur Netscape ou Internet Explorer



La notion d'architecture en couches





- Message http = document (ou page Web)
- Page Web = { objets }
- Objet = fichier (html : généralement le document de base, jpeg, gif, applet java, clip audio, …)
  Les fichiers sont repérés par une adresse = URL ("Uniform Resource Locator")
- URL = hostname/pathname
  (www.bretagne.ens-cachan.fr/DIT/People/Claude.Jard)
- Communication sous-jacente fiable = TCP (à travers les "sockets"). Déclenchement d'une requête http par exemple par un click sur un hyper-lien.

- Interopérabilité

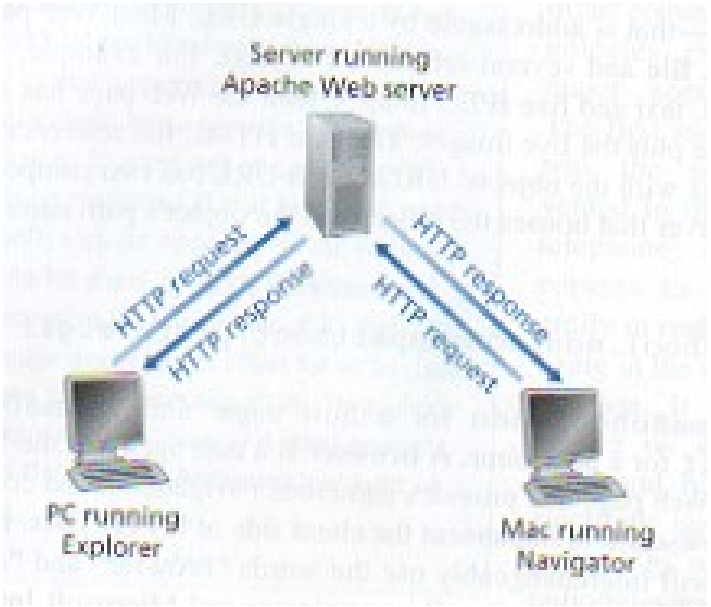
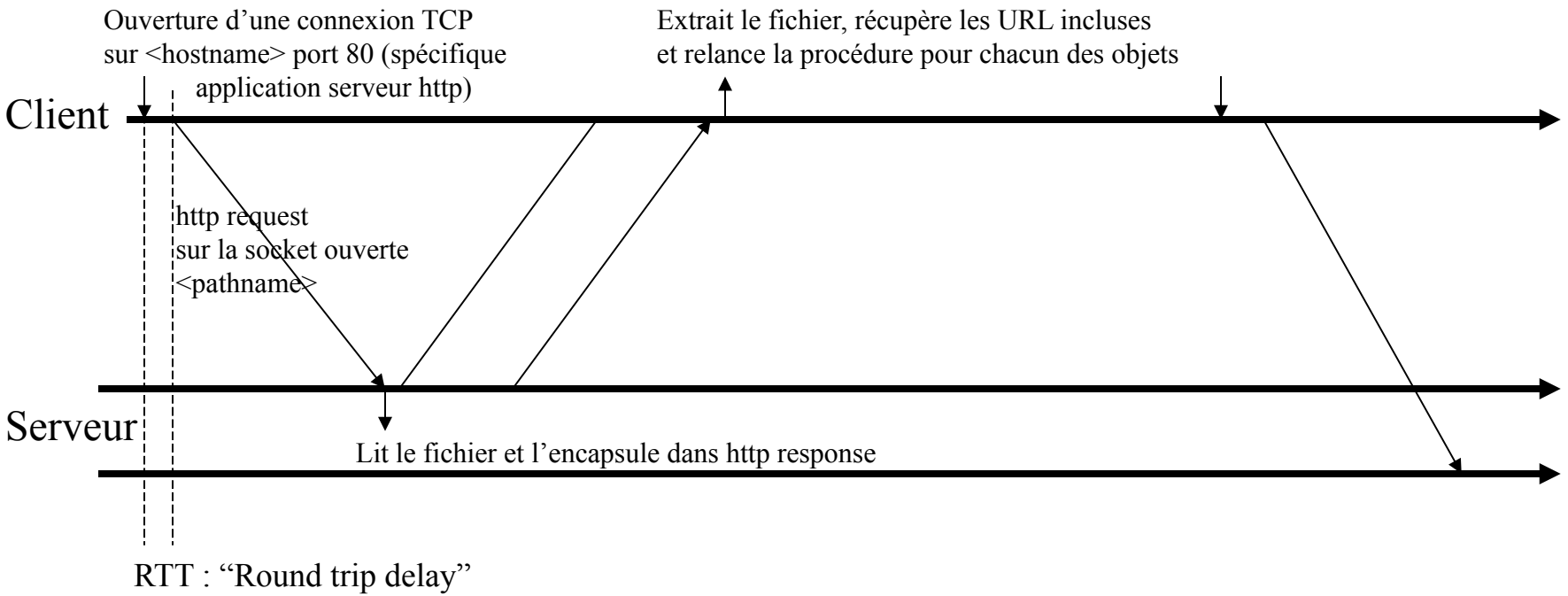


Schéma de connexion

- http est un protocole sans-mémoire : les différentes requêtes sont traitées indépendamment
- Connexion non-persistante :



- Ouverture possible de plusieurs connexions TCP en // (5 à 10 en pratique actuellement)
- Durée min d'une requête http : 2*RTT + taille_fichier/débit

- Connexion persistante :
  - Pb non-persistance = charge le serveur de la gestion de très nombreuses connexions TCP. Délais RTT importants.
  - En mode persistant, la même connexion peut être utilisée pour envoyer tous les objets de la page résidant sur le même serveur
  - Dans ce mode en général, le client peut anticiper les requêtes (pipeline des requêtes à destination d'un même serveur) -> réduction des délais
  - C'est le mode par défaut

- http://wps.aw.com/aw_kurose_network_2/0,7240,227091-,00.html

Format

- Texte ASCII
- message ::= method <sp> url <sp> version <cr> <lf>
  (header <sp> value <cr> <lf>)$^{+}$ <cr> <lf>
  body
- method ::= (get | post | head)

```
GET /somedir/page.html HTTP/1.1
Host: www.bretagne.ens-cachan.fr
Connection: close              /* non persistance */
User-agent: Mozilla/4.0        /* type du browser */
Accept-language: fr            /* si il existe plusieurs
                                  versions de l'objet */
```

- Body vide avec la méthode GET
- POST sert à passer des infos pour sélectionner la page Web (mots-clés par exemple); peut aussi passer avec GET : www.google.fr/…?clé1&clé2
- HEAD : idem GET mais ne renvoit pas l'objet (débogage)
- http 1.1 (1998) -> PUT/DELETE (outils de publication sur le serveur)

- Réponse HTTP :

```
HTTP/1.1 200 OK            /* statut */
Connection: close          /* non persistance */
Date: …                    /* date de la réponse */
Server: Apache/1.3.0 (Unix)        /* type du serveur */
Last-Modified: …           /* date de modif de l'objet (pour cache) */
Content-Length: 6821
Content-Type: text/html

(data data …)
```

- 200 OK
- 301 Moved Permanently /* nouvel URL dans le header Location: de la réponse */
- 400 Bad Request /* requête non comprise */
- 404 Not Found /* document non trouvé */
- 505 HTTP Version Not Supported

telnet www.bretagne.ens-cachan.fr 80

GET <pathname> HTTP/1.0

<lf>

Personnalisation

- "Sans mémoire" = performance de gestion simultanée de nombreuses connexions
- Nécessité d'aller au delà : protection par mot de passe

1. -> http request
2. -> http response (401 authorization required)
3. -> http request (username)
4. -> http request (password)

- Demandé pour chaque objet, mais l'identifiant et le mot de passe restent dans le cache jusqu'à fermeture (fragile)

Cookies

- Un mécanisme pour tracer le client
  - Entêtes cookie dans les requetes et réponses
  - un fichier cookie dans l'espace du client
  - une base de données associée sur le serveur

1. <- http request (cookie) : création d'une entrée dans la BD et allocation d'un ID.
2. -> http response (Set-cookie: ID)
3. <- http response : ajout d'une ligne dans le fichier cookie contenant l'adresse du serveur et l'ID

- A chaque prochaine requête, l'ID sera transmis (Cookie: ID). Le serveur peut donc tracer toute l'activité du client! Et proposer de la publicité personnalisée…

Le Web en tant que mémoire

- Cache client. Comment savoir si la copie est à jour ?

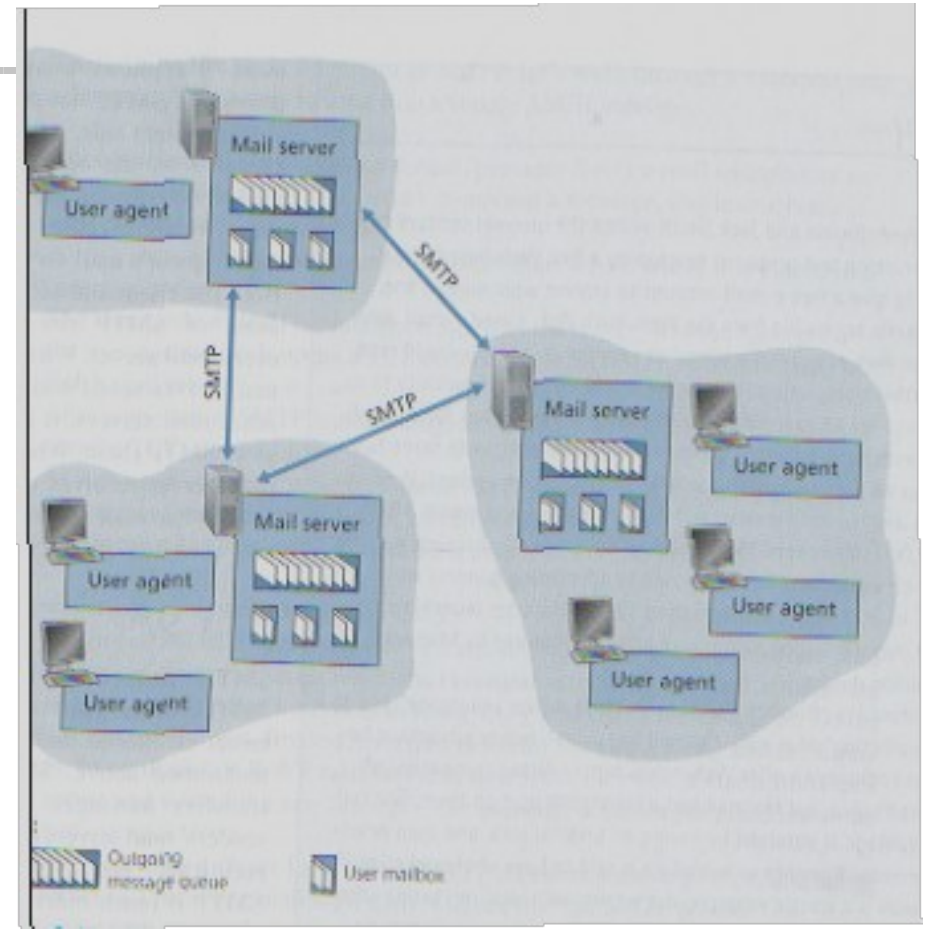
1. -> http request (GET <object>)
2. <- http response (Last-modified: <date>) : cette date est mise aussi dans le cache avec l'objet
3. -> http request (GET <object>, If-modified-since: <date>)
4. <- http response (304: Not modified)

FTP "File Transfer Protocol" : comme HTTP, il s'agit d'un protocole de transfert de document au dessus de TCP. La différence majeure est que FTP est "out-of-band" : il utilise deux connexions, une pour les données et une pour le contrôle.

-> sera vu en TP.

1.2 La messagerie

- Utilisateurs (read, reply-to, forward, save, compose)
- Serveurs de messagerie (stockage, retransmission, …)
- Protocole SMTP "Simple Mail Transfer Protocol" au dessus de TCP
- Assez ancien (norme 1982), quelques archaismes (données ASCII 7 bits -> codage/décodage multi-media)



Scenario typique

1. Composition d'un message par l'intermédiaire de l'agent utilisateur. Fourniture de l'adresse Mail du destinataire. Ordre d'émission du message.
2. L'agent envoit le message au serveur de messagerie. Le message est placé dans la file de sortie.
3. Le serveur SMTP (côté client) voit un message dans la file. Il ouvre une connexion TCP avec le serveur SMTP du destinataire (côté serveur), (port 25).
4. Après une négociation SMTP, le client SMTP envoit le message sur TCP.
5. A la réception, le serveur SMTP place le message dans la mailbox de destination.
6. Le destinataire pourra venir lire ce message.

- Il n'y a pas de stockage intermédiaire : il s'agit d'un protocole de bout-en-bout. La non disponibilité du serveur destinataire implique que le message attend dans le serveur de départ.

-> telnet <serverName> 25

<- 220 <serverName>
-> HELO <hostName serveur client>
<- 250 <serverName>
-> MAIL FROM: <sender address>
<- 250 ok
-> RCPT TO: <receiver address>
<- 250 ok
-> DATA
<- 354 End data with <CR><LF>.<CR><LF
(ASCII …)
.
<- 250 Ok: queued as 3DB75119A7C
-> QUIT
<- 221 Bye

Extension MIME pour les données non ASCII

- "Multipurpose Internet Mail Extensions" (dans la partie données)

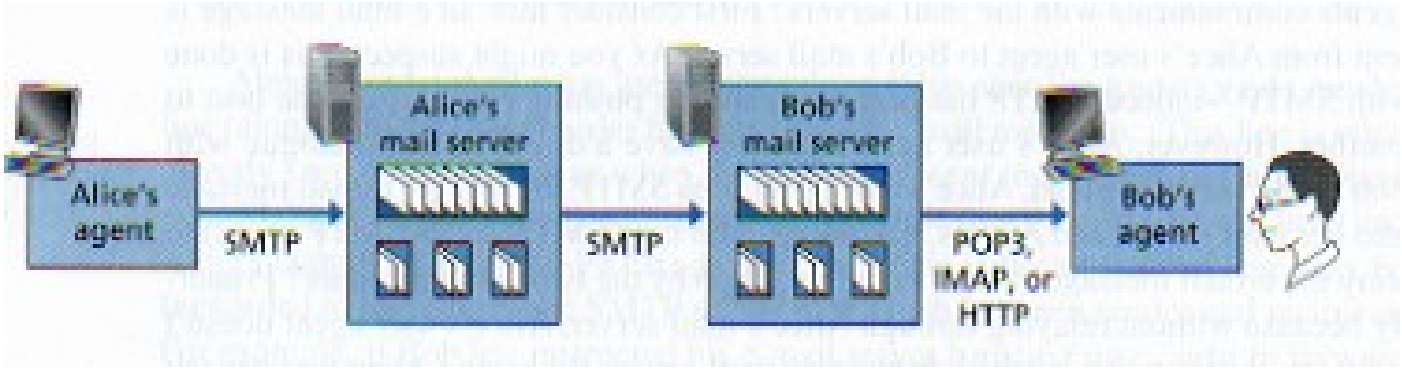
From: …
To: …
Subject: …
MIME-Version: 1.0
Content-Transfer-Encoding: base64
Content-Type: image/jpeg

(données codées en base64)

- "Quoted-printable content-transfer-encoding" /* 8bits->7bits */
- "Content-Type": text/plain text/html image/jpeg image/gif application/msword, permet de lancer l'application d'affichage à la réception
- "Multipart/mixed; Boundary=StartofNextPart" pièces attachées
- Entête "Received: from <émetteur SMTP> by <récepteur SMTP>; <date>, permet de tracer le chemin en cas de serveurs intermédiaires

Protocoles d'accès à la messagerie

- "Mail Access Protocols"
- Le serveur SMTP ne tourne pas forcément sur la machine du client récepteur (ne serait ce que pour ne pas obliger la connexion permanente de la machine)



- POP3 : "Post Office Protocol - Version 3"
- IMAP : "Internet Mail Access Protocol"
- HTTP

POP3

- telnet <mailServer> 110
- <- +OK POP3 server ready /* ou -ERR */
- -> user <userName>
- <- +OK
- -> pass <password>
- <- +OK user successfully logged on
- -> list
- <- 1 <size> /* les messages disponibles */
  2 <size>
  .
- -> retr 1 /* l'utilisateur a ses propres folders locaux */
- <- (…)
- -> dele 1
- -> retr 2
- <- (…)
- -> quit
- <- +OK POP3 server signing off

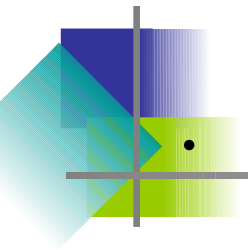
IMAP

- Le client nomade ou possédant plusieurs ordinateurs préfère que les messages soient gérés par le serveur SMTP. POP3 ne permet pas de créer des folders distants
- Chaque message est associé à un folder (les messages arrivants sont associés au folder INBOX)
- Les commandes de gestion des folders distants sont véhiculées par IMAP
- Permet aussi de ne récupérer qu'une partie du message (l'entête ou une pièce attachée seulement)

"Web-based E-mail"

- L'agent de l'utilisateur est le brouteur Web
- Utilisation de http
- Mais utilise les facilités de gestion des folders des serveurs POP3 ou IMAP (les contenus des messages http sont des scripts à destination des serveurs IMAP par exemple)

1.3 DNS "Domain Name System"

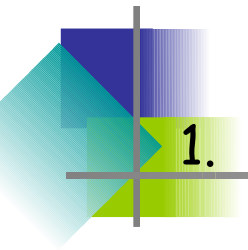


- 2 façons d'identifier une machine sur le réseau : son nom ("hostname") et son adresse IP. Le premier cas est plus explicite mais de longueur variable. Le second est de longueur fixe et est utilisé par les routeurs (adresse IP).
- Le rôle de l'annuaire DNS est de maintenir une correspondance
- Cet annuaire est réparti
- Le protocole DNS tourne au dessus de UDP sur le port 53. Il offre un service (enfoui) aux autres applications
- Offre aussi un service d'alias
- Offre aussi un service de distribution de charge dans la mesure ou un grand nombre de serveurs Web sont répliqués
- gethostbyname() /* Unix */

DNS est un excellent exemple de BD répartie sur Internet

- Serveurs locaux proches des machines clients. Typiquement configurés à la main, ils transmettent aux serveurs racines si ils ne possèdent pas l'entrée demandée (niveau établissement)
- Serveurs racines (quelques dizaines dans le monde, principalement aux US: http://netmon.grnet.gr/stathost/rootns/). Si il a la réponse, il transmet au serveur local demandeur qui transmet à son tour au client. Sinon, il demande à un serveur de référence (typiquement le serveur local de la machine demandée).
- Chaque machine a 2 serveurs de référence (redondance). Le problème est de le trouver dans le réseau. Il existe des serveurs DNS intermédiaires (routeurs): le routage s'effectue de façon hiérarchisé en utilisant la structure des noms.
- Requete récursive, requete itérative
- Notion de cache DNS. "Time To Live record"

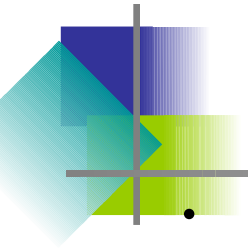
1.4 Cache Web



1. Le brouteur établit une connexion TCP avec le cache et envoit la requête http
2. Le cache renvoit l'objet si il le possède
3. Sinon le cache ouvre une connexion avec le serveur demandé
4. A la réception de l'objet, le cache stocke une copie et transmet l'objet sur la connexion ouverte précédemment

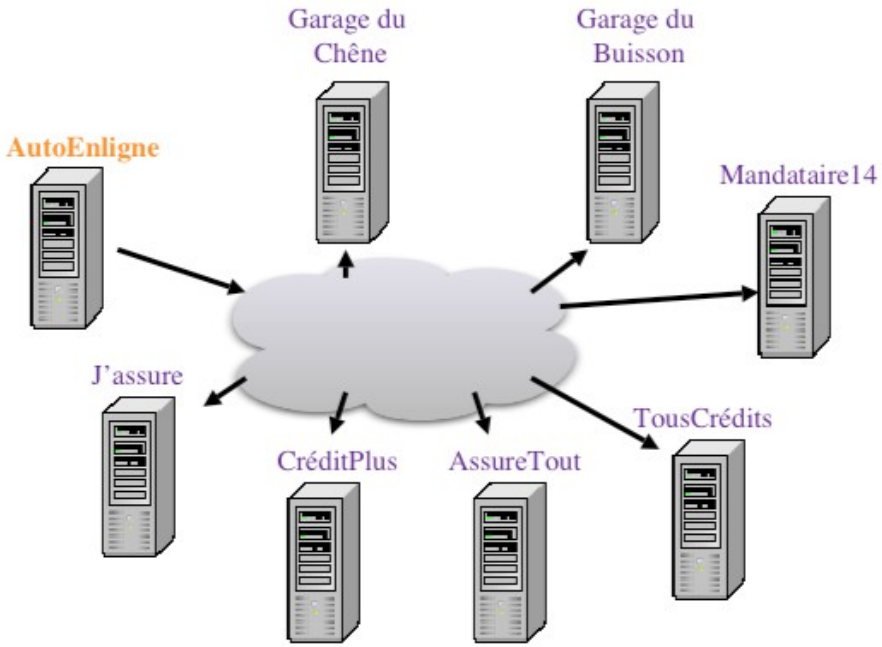
- Il y a typiquement un cache Web par établissement. S'avère efficace pour la distribution massive de contenu (pour l'ensemble des machines qui partagent le cache)
- Notion de cache coopératif (cache coeur du réseau, cache de l'établissement, …)

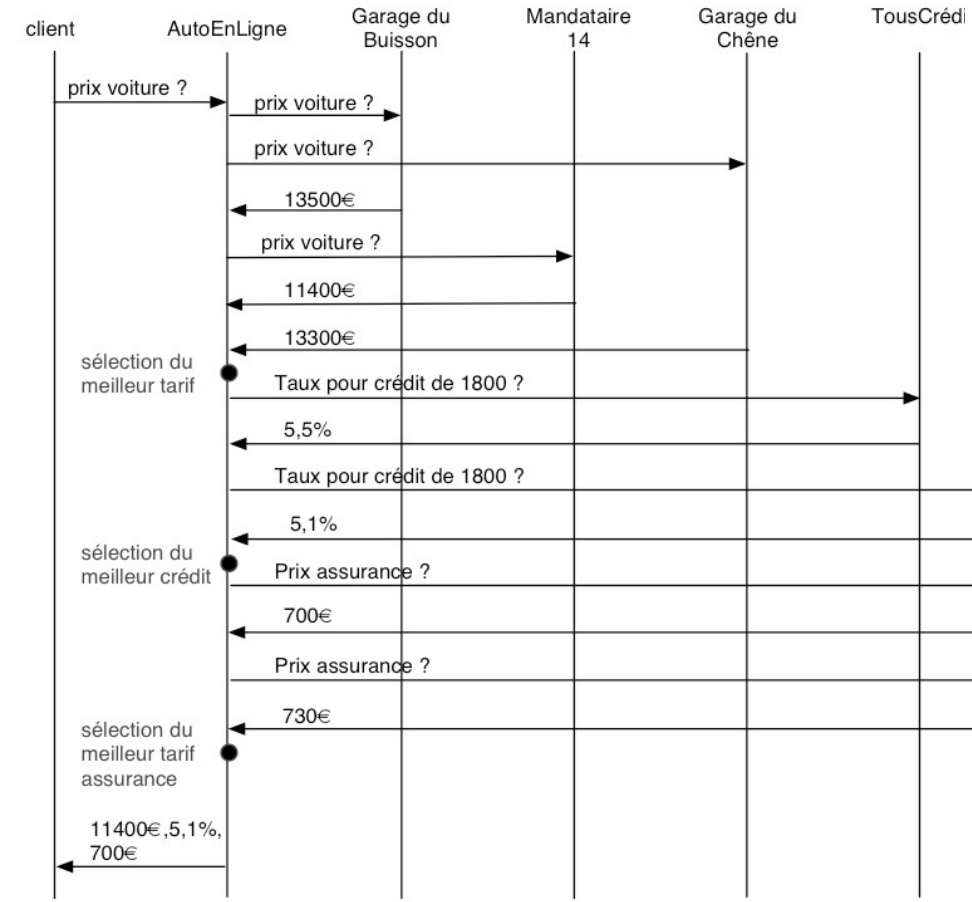
1.5 Services Web



- SOAP (“Simple Object Access Protocol”) : effectue de l’appel de procédures distantes (RPC) au-dessus de HTTP et en échangeant des messages XML.
- L'application cliente encode dans un fichier XML, la fonction qu'elle désire appeler, ainsi que les paramètres. Ce fichier est envoyé au serveur. Le serveur le décode , éxécute le traitement demandé, encode le résultat dans un fichier XML qui est renvoyé au client.
- WSDL (“Web Service Description Language”)

Autoenligne





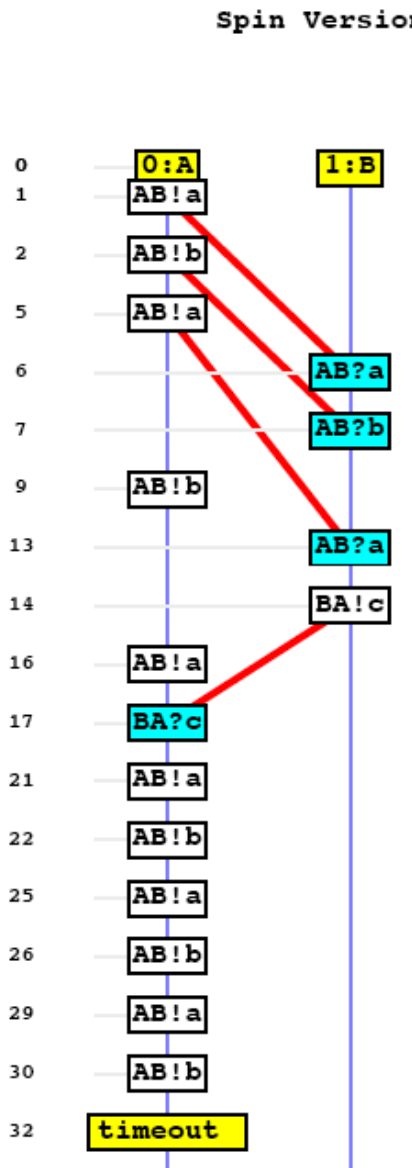
1.6 Outil de validation SPIN

Exemple du protocole de connexion-déconnexion

```
/* My favorite first example of
   complex behaviors in distributed systems */

mtype = {a,b,c};

chan AB = [8] of {mtype};
chan BA = [1] of {mtype};

active proctype A()
{
        do
          :: AB!a;
             if
             :: BA?c;
             :: AB!b;
             fi
        od
}
```

```
active proctype B()
{
        do
          :: AB?a;
             if
             :: AB?b;
             :: BA!c;
             fi
        od
}
```

Exemple de scénario



spin -M Codec0

spin -a codec0
cc -o pan pan.c
./pan -E

(Spin Version 4.0.7 -- 1 August 2003)
+ Partial Order Reduction

Full statespace search for:
never claim - (none specified)
assertion violations +
acceptance cycles - (not selected)
invalid end states - (disabled by -E flag)

State-vector 36 byte, depth reached 16, errors: 0
43 states, stored
24 states, matched
67 transitions (= stored+matched)
0 atomic steps
hash conflicts: 0 (resolved)
(max size 2^18 states)

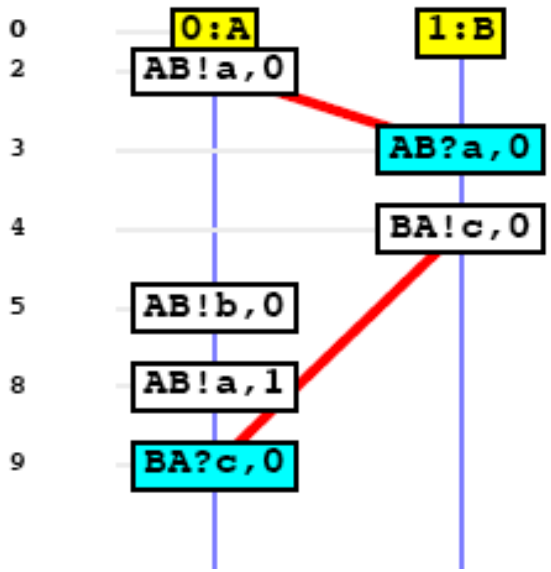
1.573 memory usage (Mbyte)

Vérification

```
#define MaxNumberCn 10

mtype = {a,b,c};

chan AB = [8] of {mtype,int};
chan BA = [1] of {mtype,int};

active proctype A()
{
 int n = 0; /* sequence number, to test a possible intertwining of connections
      */
 int m;
      do
        :: (n < MaxNumberCn) -> AB!a,n;
           if
           :: BA?c,m; assert (m == n); n++;
           :: AB!b,n; n++;
           fi
       od
}
```

```
active proctype B()
{
 int m1,m2;
       do
         :: AB?a,m1;
            if
            :: AB?b,m2; assert (m1 == m2);
            :: BA!c,m1;
            fi
       od
}
```

Découverte d'un bug



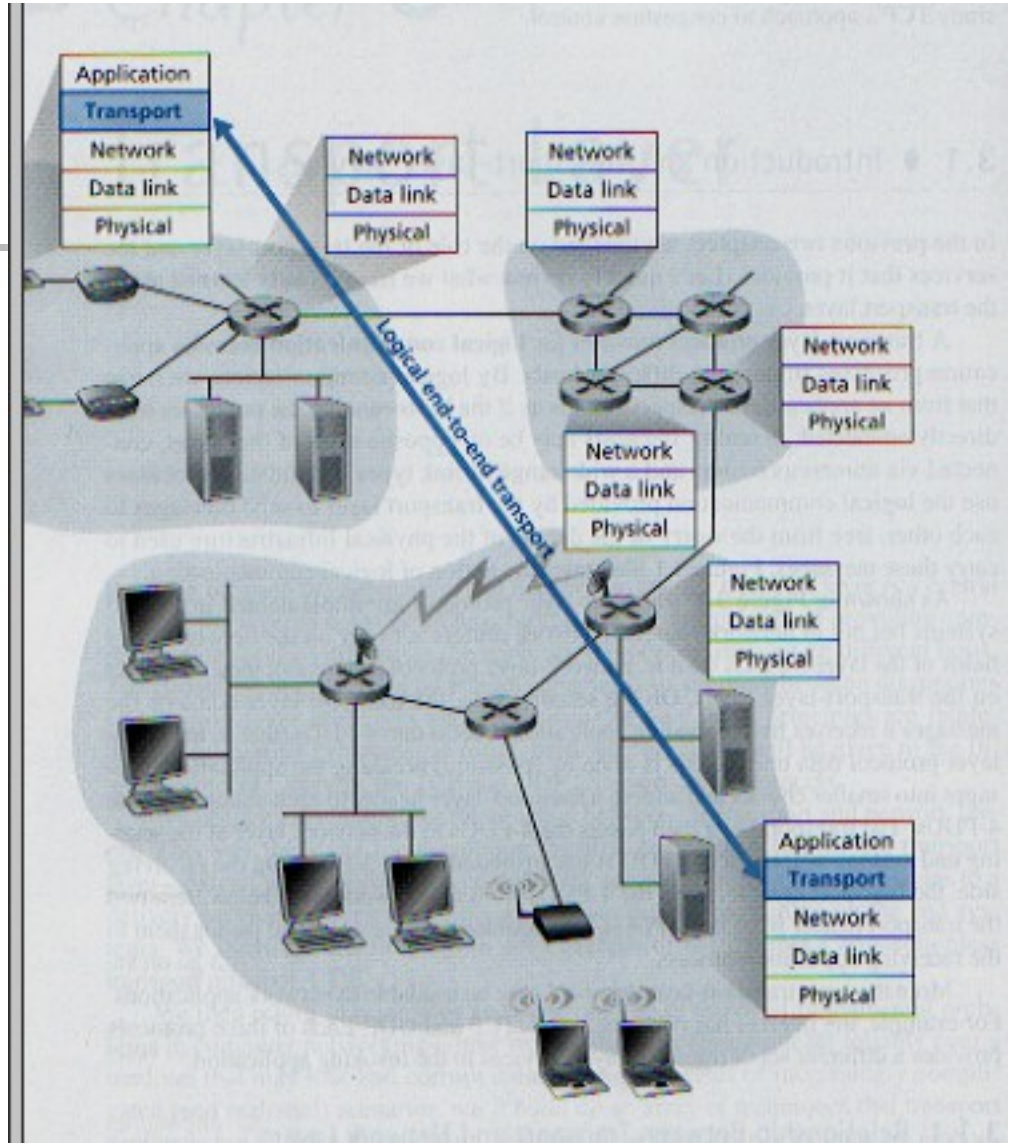

```
spin -a codec0v
cc -o pan pan.c
./pan -E -i
pan: assertion violated (m==n) (at depth 57)
pan: wrote Codec0v.trail
pan: reducing search depth to 57
pan: wrote Codec0v.trail
pan: reducing search depth to 57
pan: wrote Codec0v.trail
pan: reducing search depth to 51
pan: wrote Codec0v.trail
pan: reducing search depth to 51
pan: wrote Codec0v.trail
pan: reducing search depth to 45
pan: wrote Codec0v.trail
pan: reducing search depth to 45
pan: wrote Codec0v.trail
pan: reducing search depth to 39
pan: wrote Codec0v.trail
pan: reducing search depth to 39
pan: wrote Codec0v.trail
pan: reducing search depth to 33
pan: wrote Codec0v.trail
pan: reducing search depth to 33
pan: wrote Codec0v.trail
pan: reducing search depth to 27
pan: wrote Codec0v.trail
pan: reducing search depth to 27
pan: wrote Codec0v.trail
```

```
pan: reducing search depth to 21
pan: wrote Codec0v.trail
pan: reducing search depth to 21
pan: wrote Codec0v.trail
pan: reducing search depth to 15
pan: wrote Codec0v.trail
pan: reducing search depth to 15
pan: wrote Codec0v.trail
pan: reducing search depth to 9
pan: wrote Codec0v.trail
pan: reducing search depth to 9
(Spin Version 4.0.7 -- 1 August 2003)
        + Partial Order Reduction

Full statespace search for:
        never claim             - (none specified)
        assertion violations    +
        acceptance   cycles     - (not selected)
        invalid end states      - (disabled by -E flag)

State-vector 116 byte, depth reached 61, errors: 18
     239 states, stored
     162 states, matched
     401 transitions (= stored+matched)
       0 atomic steps
hash conflicts: 0 (resolved)
(max size 2^18 states)

spin -M -t Codec0v
```

Cours 2 : la couche de transport, étude de TCP (UDP)

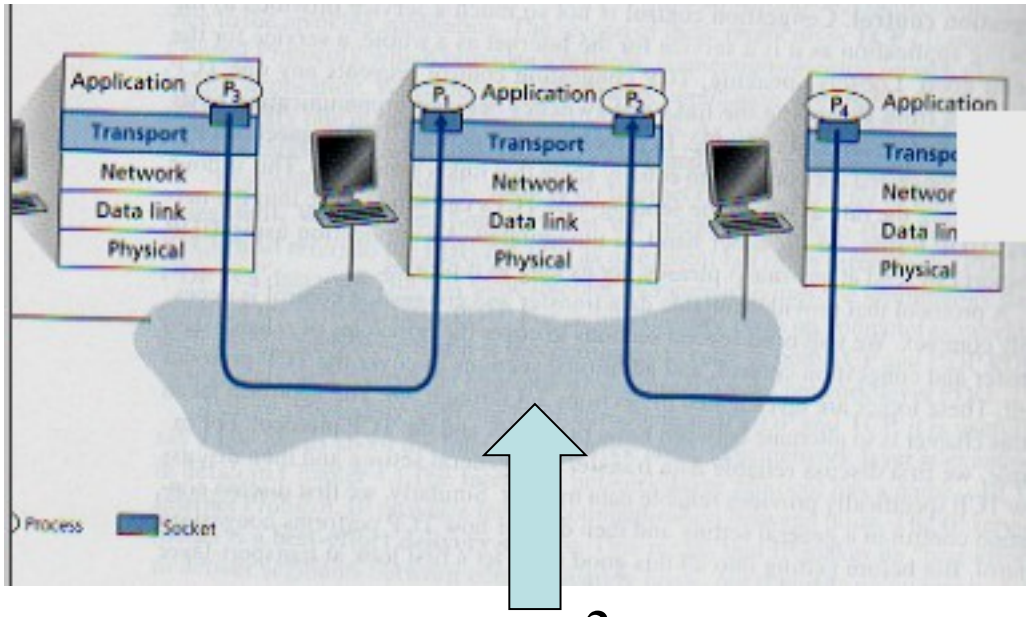
- 2.1 Multiplexage et démultiplexage
- 2.2 Principes du transfert fiable
- 2.3 Go-Back-N
- 2.4 Selective Repeat
- 2.5 Transmission Control Protocol
- 2.6 Estimation des temps
- 2.7 Connexion TCP
- 2.8 Contrôle de flux
- 2.9 Contrôle de congestion

Le "tuyau" logique de bout-en-bout



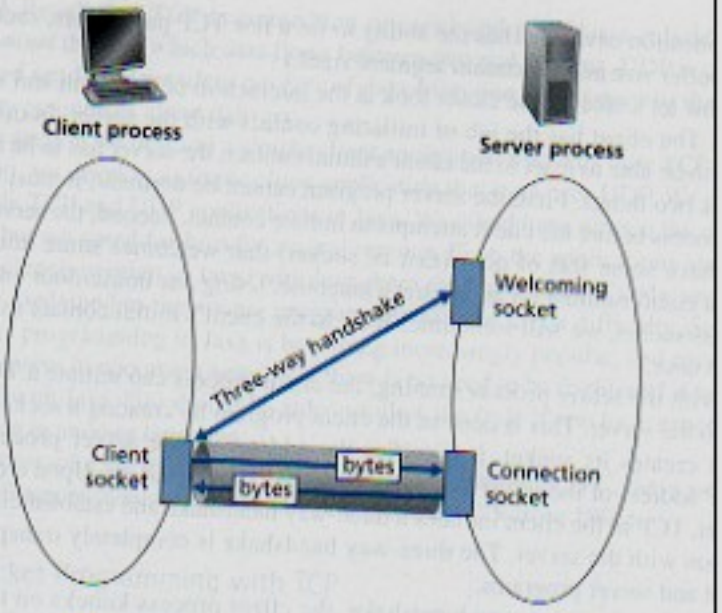
2.1 Multiplexage et démultiplexage

- La couche Transport reçoit des données qu'elle doit acheminer sur le bon processus de l'application (connu par sa "socket")
- L'identificateur (unique sur un élément de réseau) de la socket doit être encapsulé à l'émission et interprété à la réception
- Les données émises sur plusieurs sockets doivent être agglomérées (avec l'id associé) pour passer dans le tuyau (multiplexage). A l'arrivée, la distribution sur les sockets est le démultiplexage



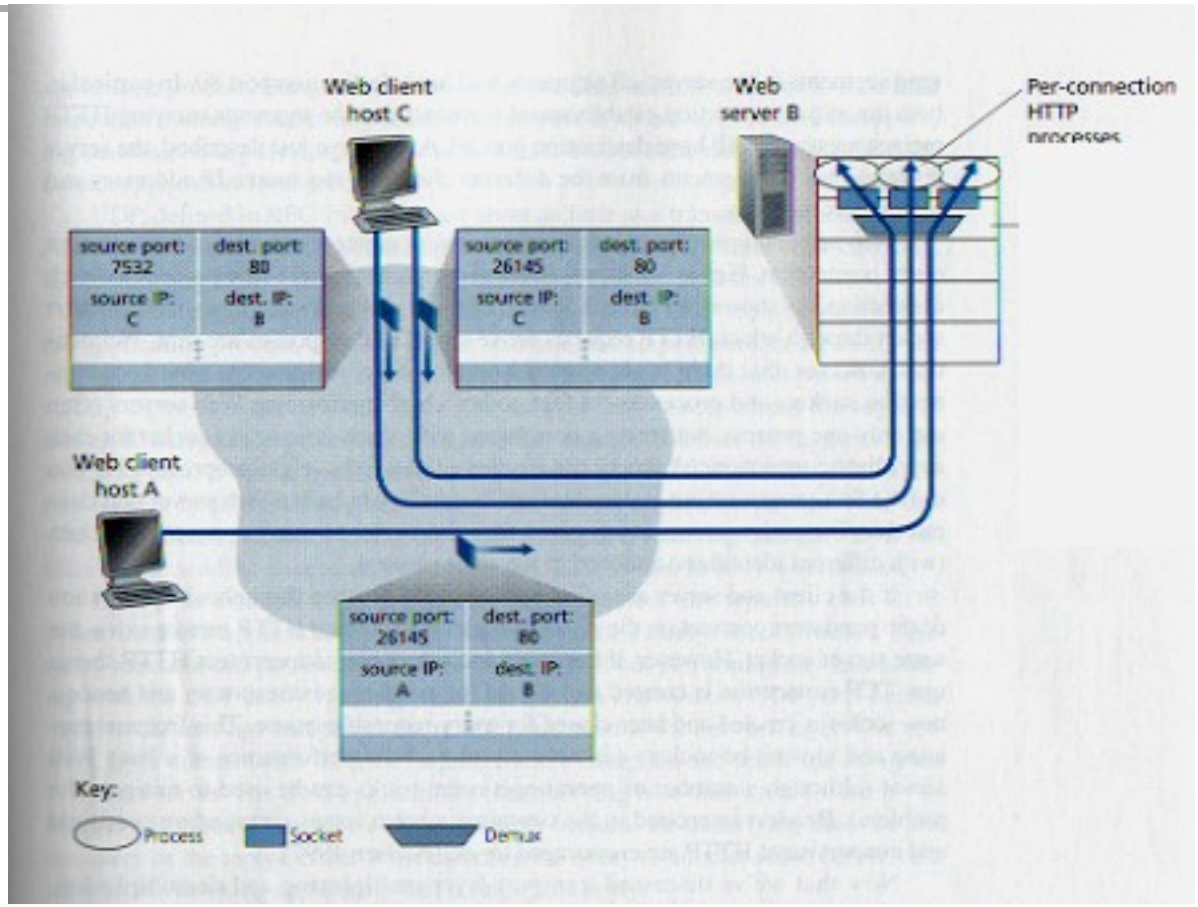
Les ports

- Chaque segment de données comprend une entête de 2x16 bits: *(source_port_number_field, destination_port_number_field)*
- 0..1023 "well-known port numbers" sont réservés pour les applications standards (http, ftp, …) -> disponibles à http://www.iana.org (/etc/services)
- Le mécanisme de création de sockets à distance

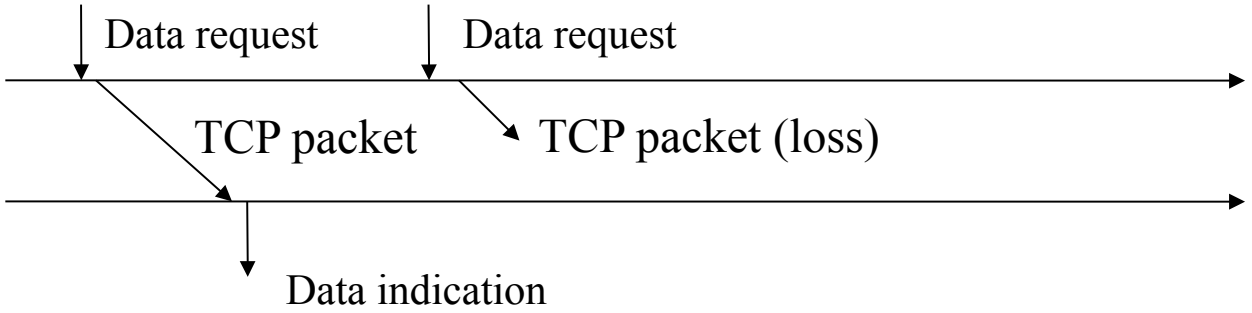


Identité = port id + adresse ip

- Collision possible entre ports sur des machines différentes ->
- Une connexion TCP est identifiée par un quadruplet (Sport,Sip, Dport,Dip)



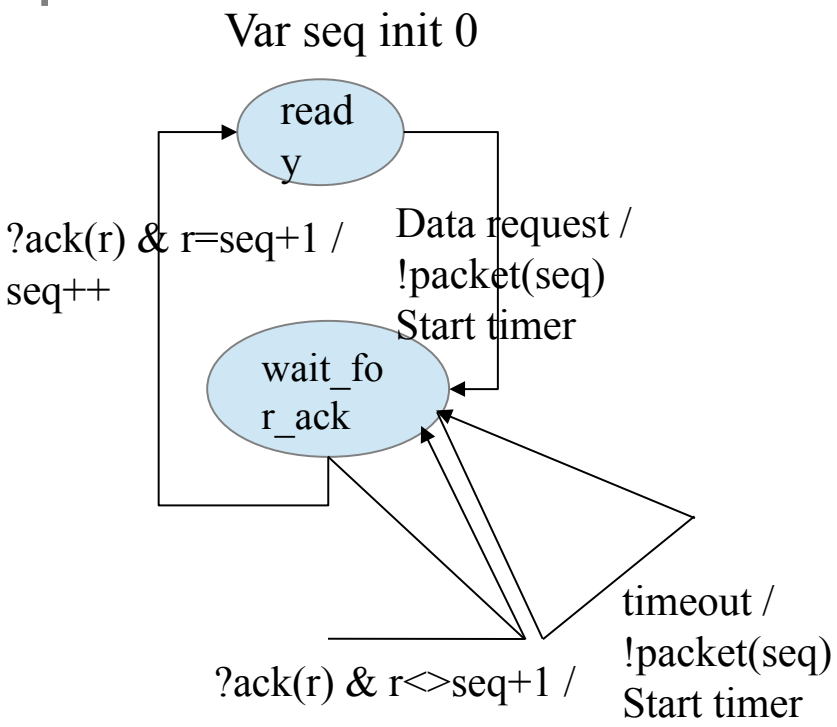
2.2 Principes du transfert fiable au dessus d'une couche non fiable

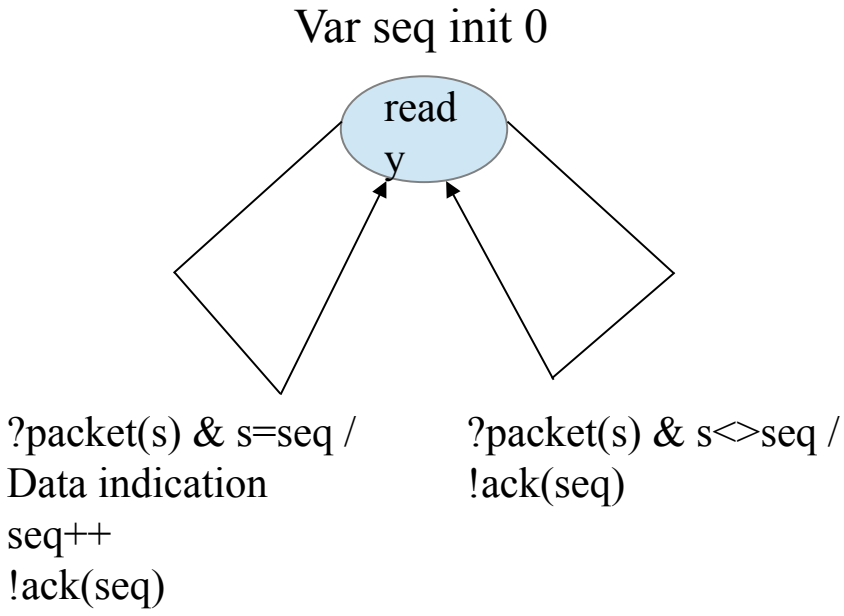


Avec erreurs de transmission

- Détection d'erreurs bit : ex. Checksum → provoque la perte en pratique
- Retransmission ("Stop and Wait Protocols")
- Retour de récepteur : ACK (accusé)

Avec pertes
'stop and wait protocol'

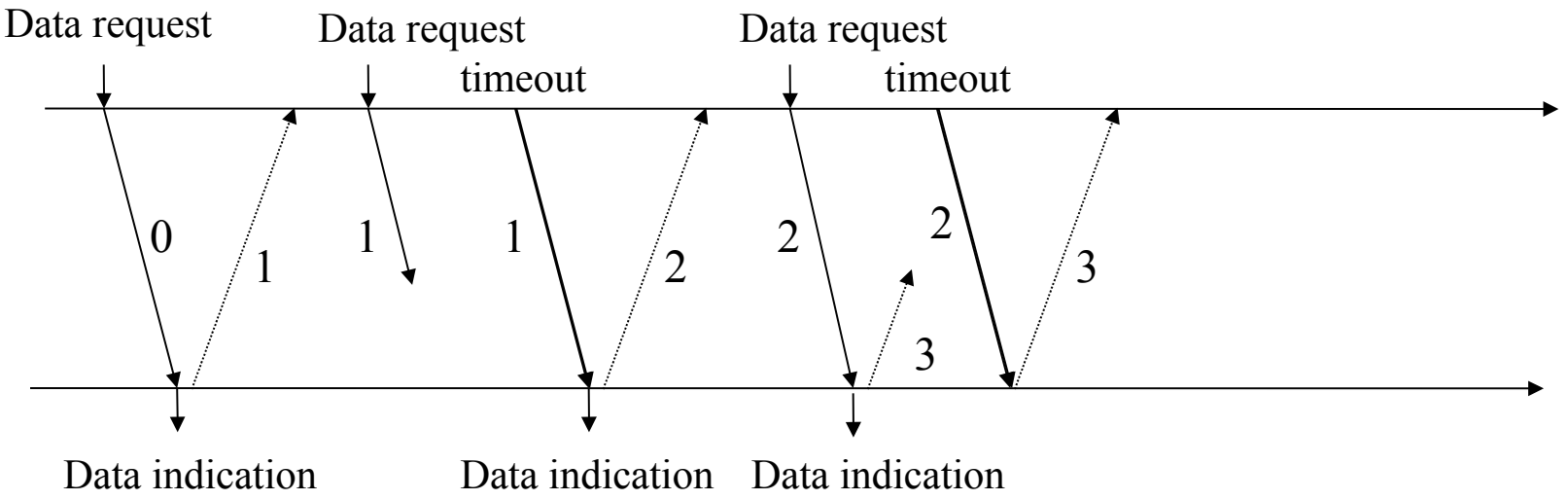




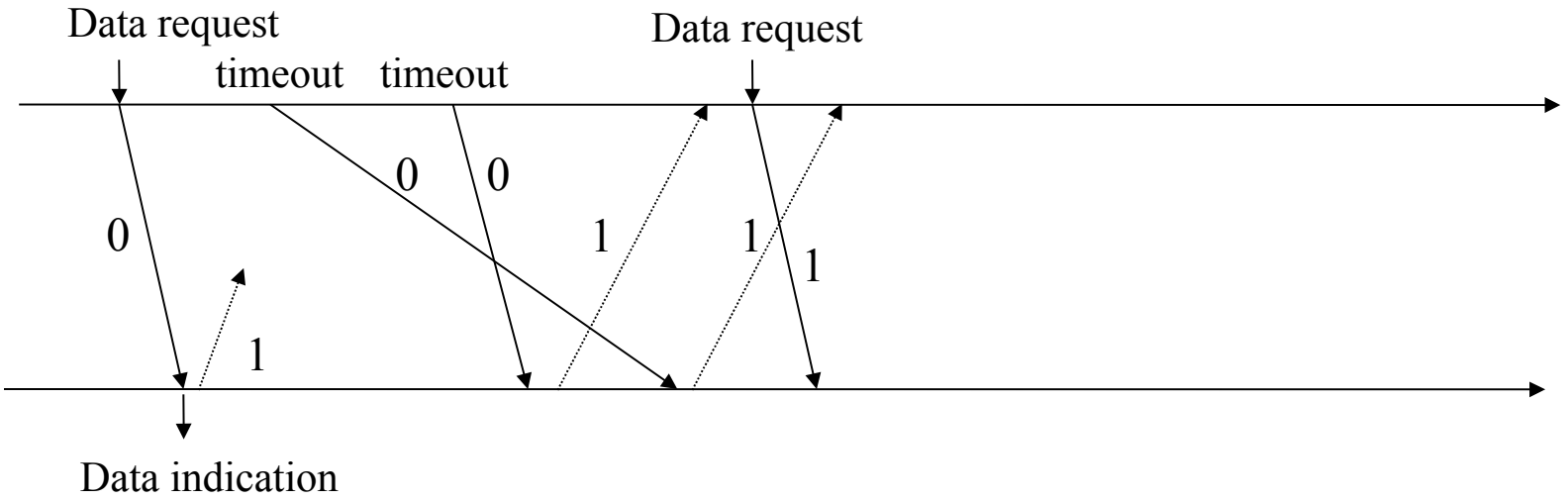
Côté émetteur

Côté récepteur

Scénario possible



Scénario possible compliqué



Quid du problème de l'utilisation (nécessaire) de l'arithmétique modulo ? →
Il y a des cas (peu probables) où TCP peut être mis en défaut...

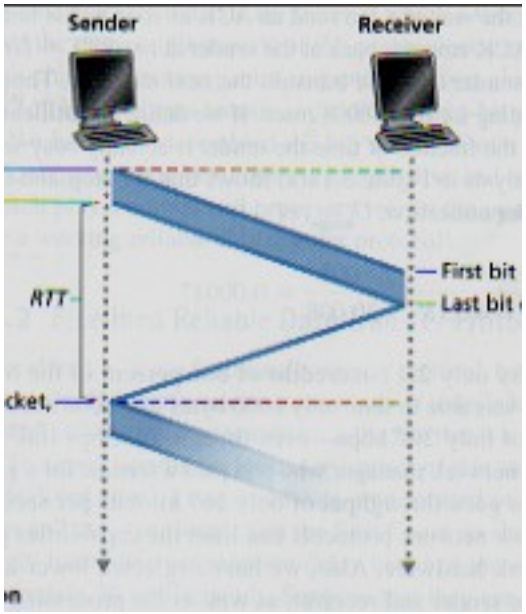
La nécessité de d'anticiper (pipeline)

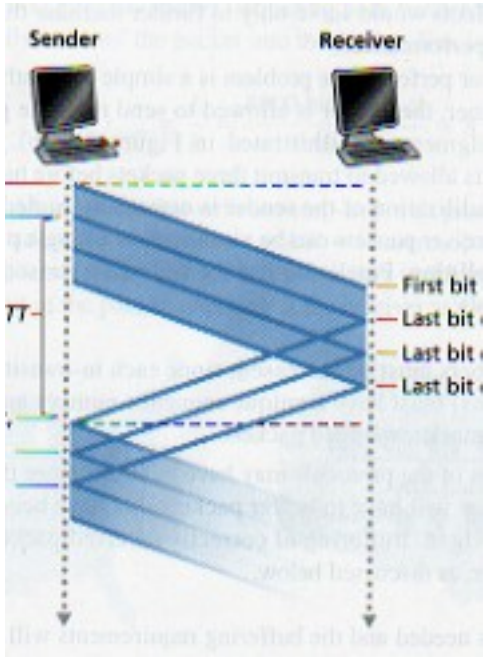
- Un RTT est nécessaire à chaque transmission de paquet : à la vitesse de la lumière pour 1000km -> $6x10^{-3}s$
- Pour un paquet de 1000 octets (L) et un canal de 1Gbps (R), temps de transmission L/R = $8x10^{-6}s$
- Temps total -> RTT/2 + L/R = $3,008x10^{-3}s$ récepteur
- Ack court : temps total émetteur -> RTT + L/R = $6.008x10^{-3}s$
- Pourcentage d'utilisation -> (L/R)/(RTT+L/R) = 0,00013
- L'émetteur est seulement occupé 1,3 millième du temps, ou débit effectif ("throughput") = $8x1000/3,008x10^{-3}$ = 266kbps (pour un lien disponible à 1Gbps !)

-> c'est un bon exemple qui montre comment l'intercalement d'un protocole dégrade les performances.

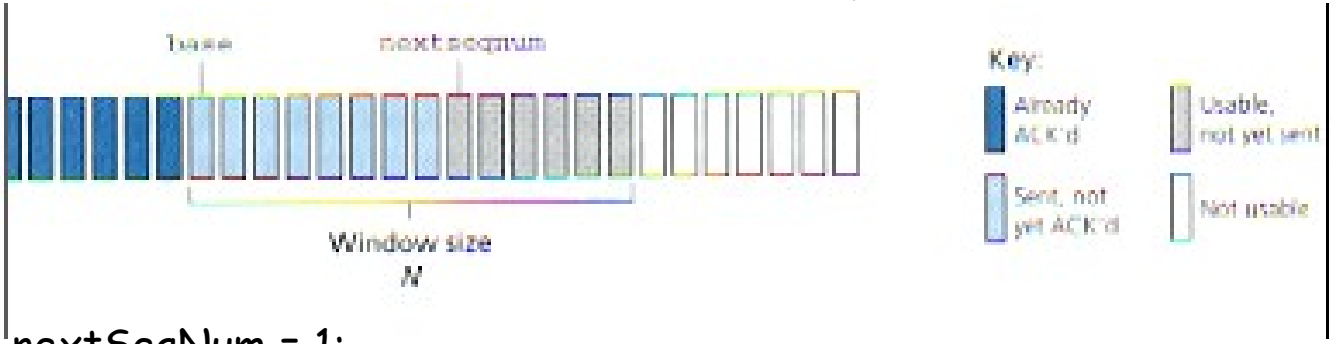
Pipeline

- On augmente l'intervalle de numérotation des messages
- Plusieurs messages peuvent être en transit et non encore accusés
- L'émetteur doit mémoriser des paquets (éventuellement le récepteur)
- Deux méthodes : "Go-Back-N" et "selective repeat"



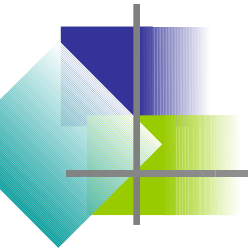


2.3 Go-Back-N ("Sliding window protocol")




```
• int base = 1; int nextSeqNum = 1;
•:: US?data,d ->
    if :: (nextSeqNum<base+N) -> Sbuf[nextSeqNum] = (nextSeqNum,d);
                                 SM!packet,Sbuf[nextSeqNum];
                    if :: (base == nextSeqNum) -> startTimer;
                       :: else -> skip;
                    fi; nextSeqNum++;
       :: else US!refuse;
   fi;
•:: timeout -> startTimer; SM!packet,Sbuf[base];
                  SM!packetSbuf[base+1]; … SM!packetSbuf[nextSeqNum-1];
•:: RS?ack,m -> base = m+1;  /* Cumulative acknowledgment */
                if :: (base == nextSeqNum) -> stopTimer;
                   :: else -> startTimer;
                fi;
```

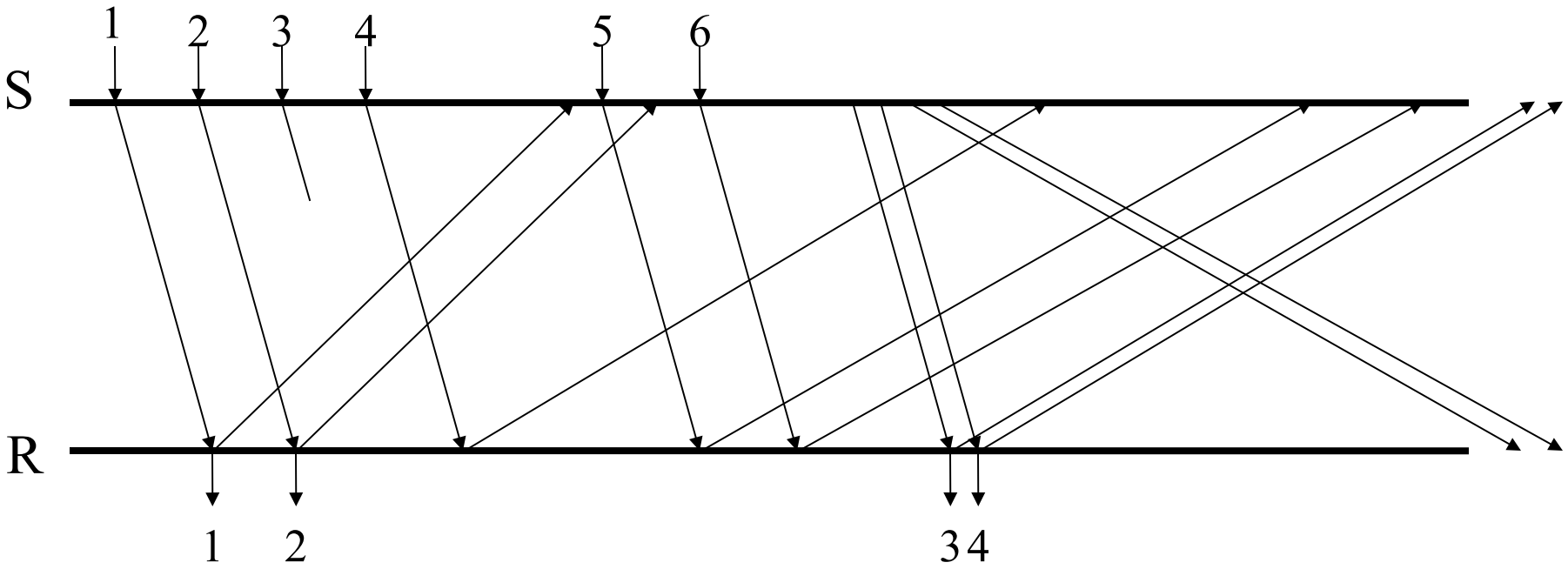
Go-Back-N : le récepteur



```
• int expectedSeqNum = 1;
• :: SR?packet,d,n ->
    if :: (n == expectedSeqNum) -> RU!data,d; RM!ack,expectedSeqNum;
                                   expectedSeqNum++;
       :: else -> RM!ack,expectedSeqNum-1;
    fi;
```

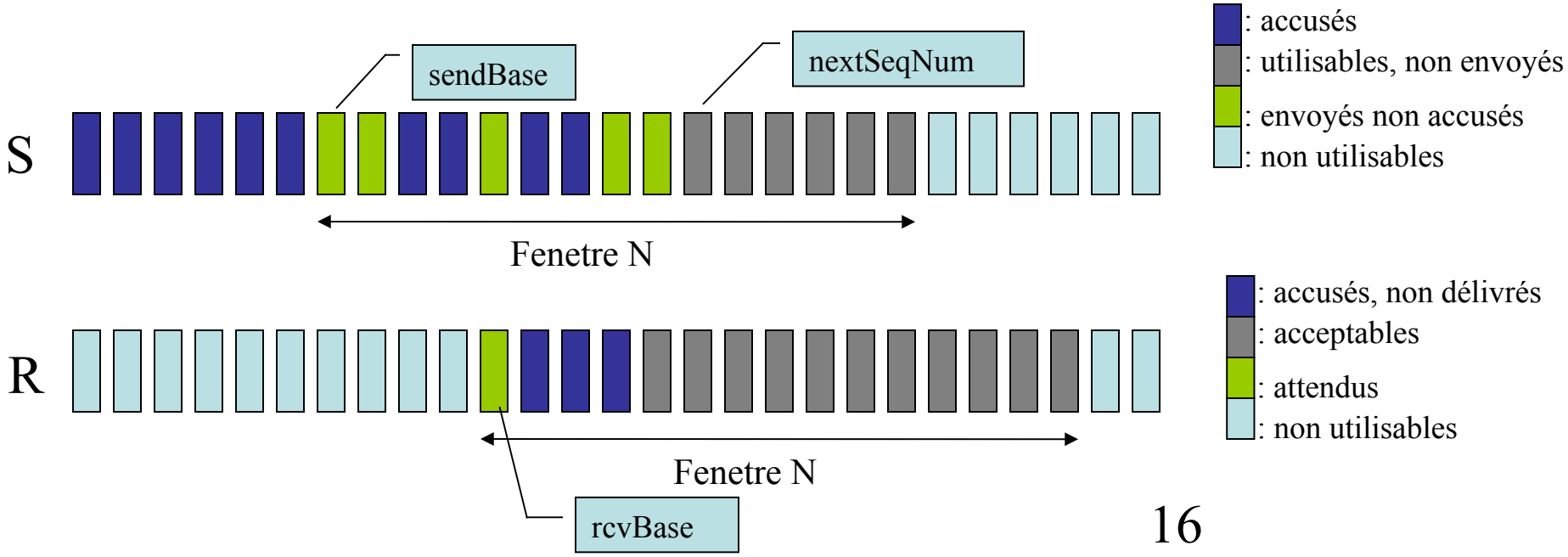
- Cette méthode simple jette les paquets hors-séquence, meme non dupliqués. Eviter ce gaspillage demande un mécanisme de tamponnage plus complexe à la réception pour maintenir le séquencement des données

Go Back N en action (fenêtre de 4)



2.4 "Selective Repeat"

- Le GBN résiste mal à l'augmentation du taux d'erreur car le pipeline se remplit avec des paquets inutiles
- La solution est d'accuser les paquets individuellement. Un ack peut donc concerner un paquet encore à l'intérieur de la fenêtre d'émission



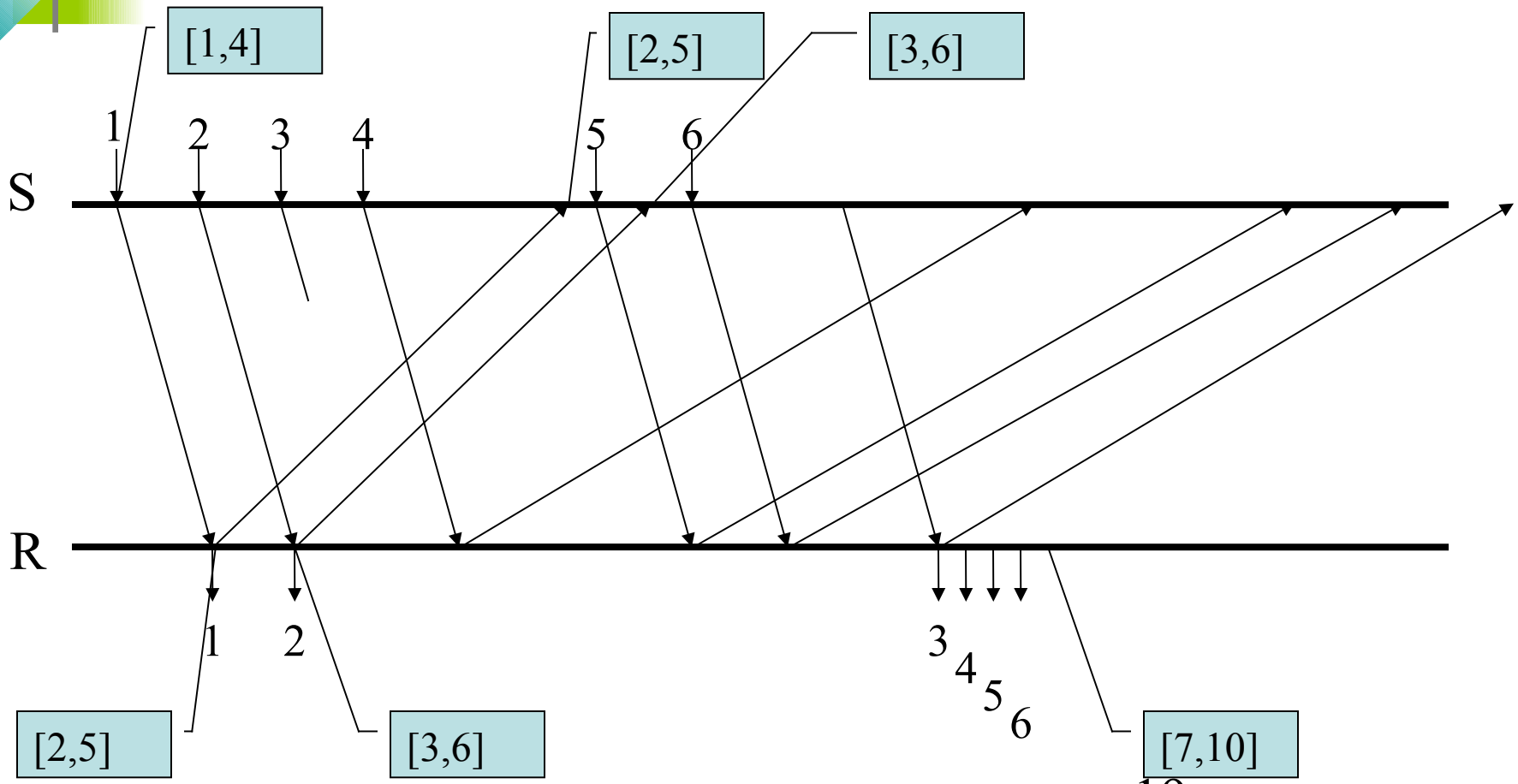
Emetteur SR

- Sur réception utilisateur ->
  si nextSeqNum est dans la fenêtre -> envoi du paquet
  sinon refus (ou tampon utilisateur)
- Retransmission sur timeout (un timer logique par paquet envoyé)
- Sur réception d'un ack,n -> marquage du paquet correspondant dans la fenêtre comme accusé;
  si (n == sendBase) -> sendBase = plus petit numéro de paquet non encore accusé

Récepteur SR

- Sur réception d'un paquet,n ->
  si (rcvBase < n < rcvBase+N) -> envoi ack,n
  si (rcvBase == n) -> envoi ack,n;
  délivrance à l'utilisateur de tous les paquets mémorisés de numéros consécutifs à partir de rcvBase;
  rcvBase est augmenté du nombre de paquets délivrés
  si (rcvBase-N <= n < rcvBase) -> ack,n
  /* meme si déjà accusé */
  sinon ignore le paquet

SR en action (fenêtre de 4)



Et en cas de réordonnancement ?

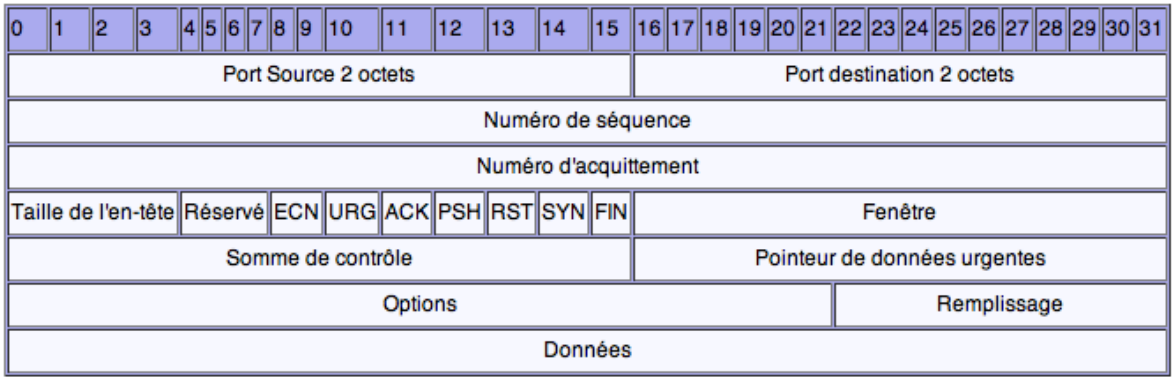
- Trouver un exemple de duplication non détectée, meme pour une fenêtre respectant les conditions…

-> hypothèse de durée de vie maximum pour un paquet.

2.5 TCP : Transmission Control Protocol

- "connection-oriented"
- "full-duplex"
- "piggybacked acknowledgments"
- "point-to-point"
- "Maximum Segment Size" : fragmentation et réassemblage dans les tampons d'émission et de réception : http://www.awl.com/kurose-ross. MSS dépend de l'implémentation
- "cumulative acknowledgments" (GBN)

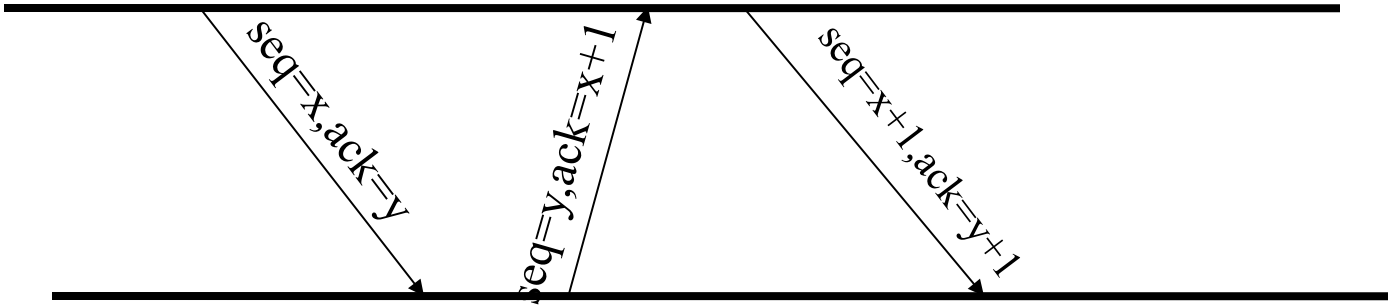
En-tête TCP
(typiquement 20 octets)



- no séquence, no ACK : 32 bits,
- Flags : Syn (ouverture de connexion), Fin (fermeture de connexion),
- Rst (reset), Ack (no. ack actif),
- Urg (pointeur urgent actif), Psh (push, délivrance immédiate) : inutilisés
- Fenêtre annoncée (8ko par défaut, max 64ko), contrôle de flux
- Options : ex. MSS (536o par défaut, max 1460o), timestamp…

La numérotation

- Ce ne sont pas les segments (paquets) qui sont numérotés, mais les octets de données dans le flux de données émetteur-récepteur
- Le numéro d'accusé est le numéro de séquence du prochain octet attendu en provenance du partenaire
- Curieusement, la norme ne dit pas ce que le récepteur doit faire d'un segment hors-séquence



2.6 Réglage des timers : estimation du RTT

- TCP utilise un RTT estimé à partir des RTT mesurés qui fluctuent de façon importante

  RTTestimé =

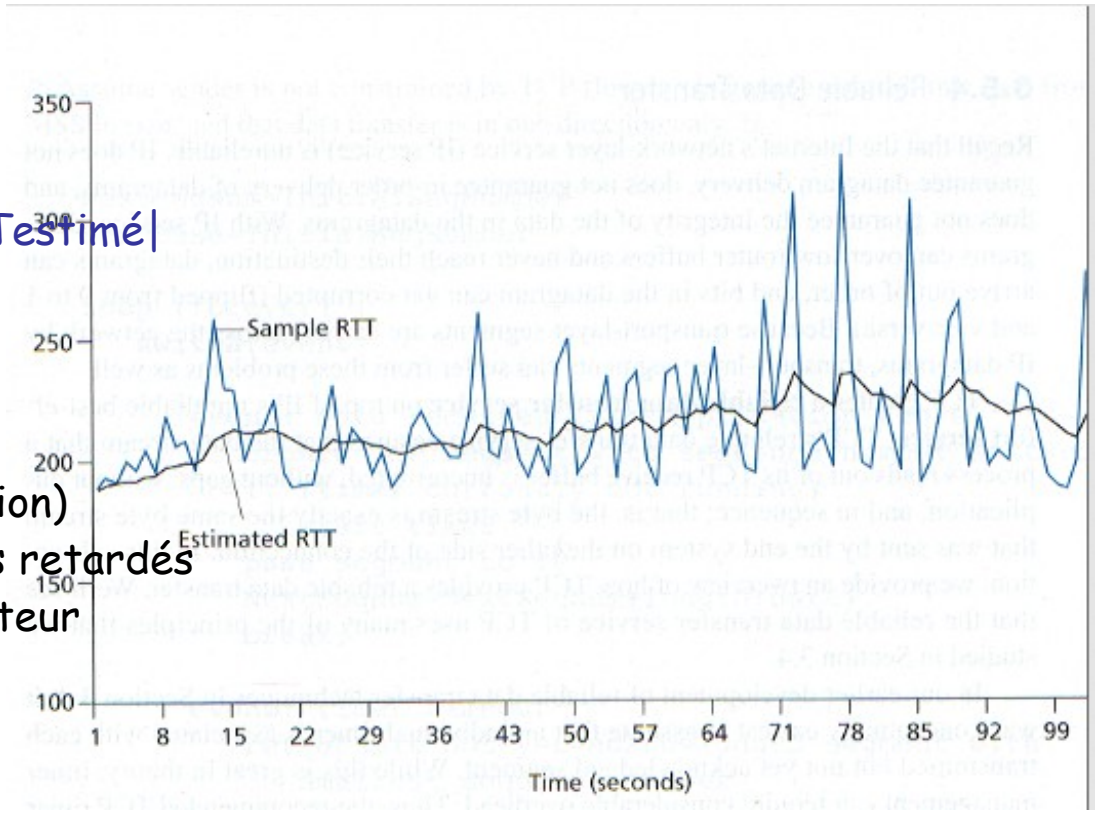
  (1-A)*RTTestimé + A*RTTmesuré

  (1/8 recommandé)
- Variabilité du RTT :

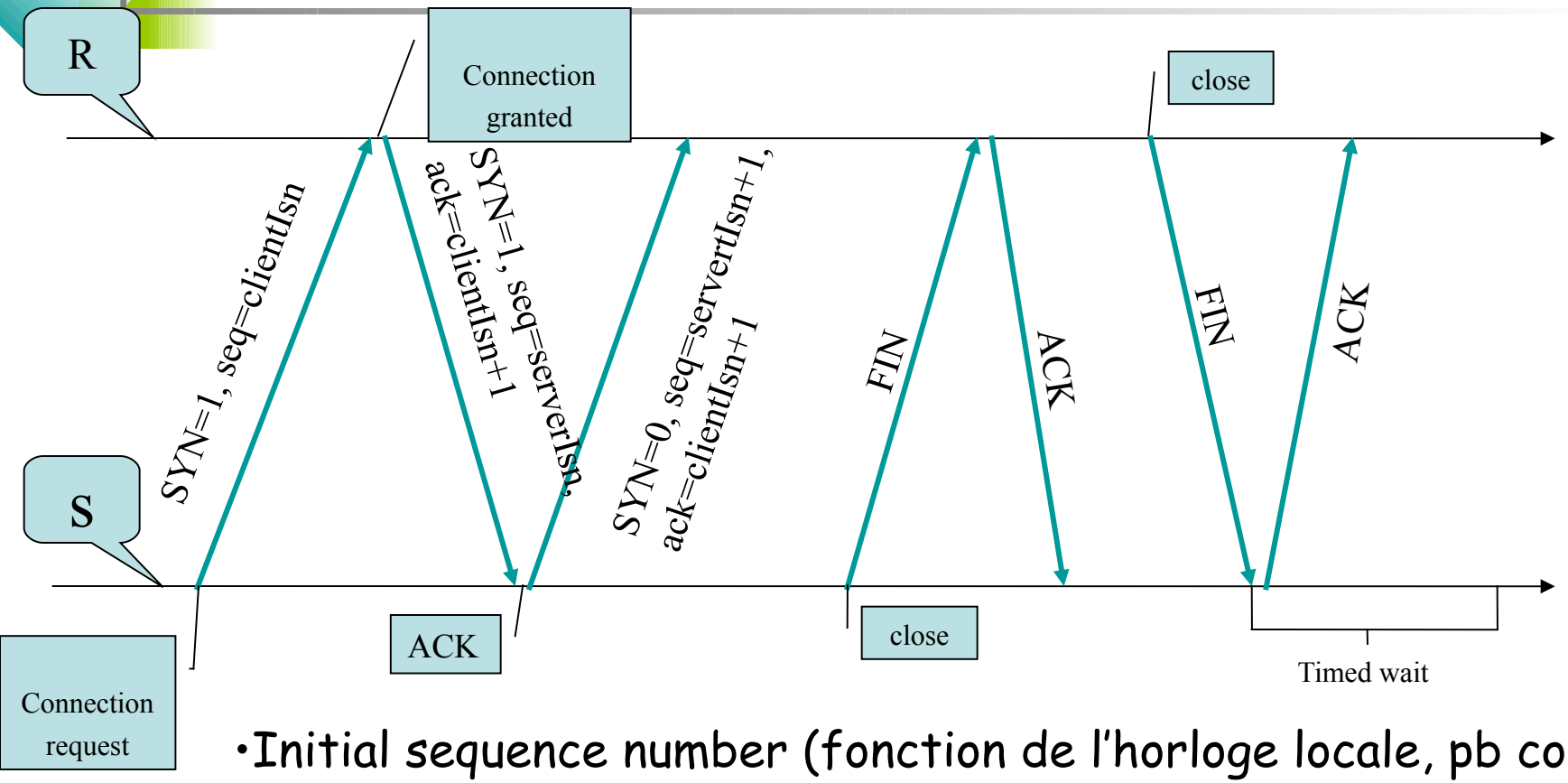
  RTTvar =

  (1-B)*RTTvar + B*|RTTmesuré-RTTestimé|

  (1/4 recommandé)
- Timeout = RTTestimé + 4*RTTvar
- Doublé à chaque retransmission (une forme de maîtrise de la congestion)
- Optimisation en utilisant des accusés retardés
- Accusés tripliqués indiquent à l'émetteur la perte du dernier paquet

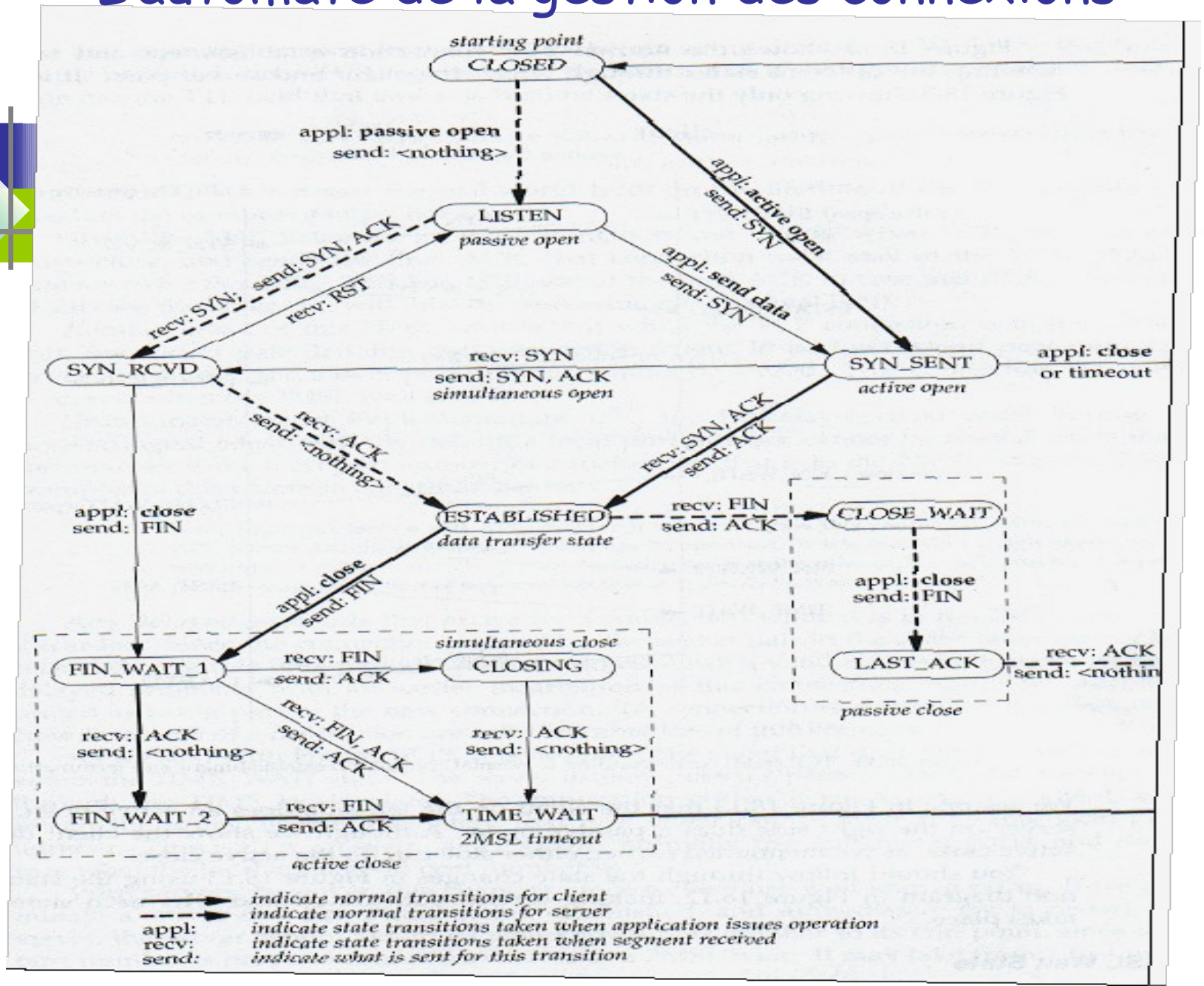


2.7 La phase de connexion-déconnexion:
"TCP three-way-handshake"



- Initial sequence number (fonction de l'horloge locale, pb collisions)
- Allocation des tampons

L'automate de la gestion des connexions



2.8 Le contrôle de flux

- Le récepteur maintient une variable, appelée "receive window" (`RcvWindow`) qui représente la place libre dans le tampon de réception. Cette valeur est passée à l'émetteur dans le champ "fenêtre" des segments.
- `RcvBuffer` est la taille du tampon en réception
- `LastByteRead` est le numéro du dernier octet délivré à l'utilisateur du récepteur
- `LastByteRcvd` est le numéro du dernier octet reçu chez le récepteur
- `LastByteRcvd - LastByteRead <= RcvBuffer`
- `RcvWindow = RcvBuffer - (LastByteRcvd - LastByteRead)` est une variable dynamique
- Initialement `RcvWindow = RcvBuffer`
- L'émetteur maintient 2 autres variables : `LastByteSent` et `LastByteAcked`
- Le contrôle est assuré en maintenant chez l'émetteur : `LastByteSent - LastByteAcked <= RcvWindow`
- Problème du blocage lorsque le récepteur a vidé son tampon ! -> dans ce cas, un octet supplémentaire est envoyé par l'émetteur.

2.8 Le contrôle de flux

- Le récepteur possède un tampon de réception. L'application vient chercher ses données dans ce tampon de façon asynchrone.

- Sans service de contrôle de flux un émetteur pourrait rapidement saturer le tampon de réception dont la taille est fixe.

- Le récepteur informe l'expéditeur de l'espace disponible en insérant la variable *RcvWindow* dans la fenêtre de réception de tous les segments qu'il lui envoie.

- Lorsque *RcvWindow*=0 et que le récepteur n'a rien à envoyer, l'émetteur continue à envoyer des segments de 1 octets qui généreront des acquittements porteur des informations d'actualisation sur le tampon de réception

2.9 Le contrôle de congestion

- Contrôle de flux : contrôle du débordement de la mémoire du récepteur

- Contrôle de congestion : contrôle du débordement du réseau (les routeurs)

Congestion : cas idéal

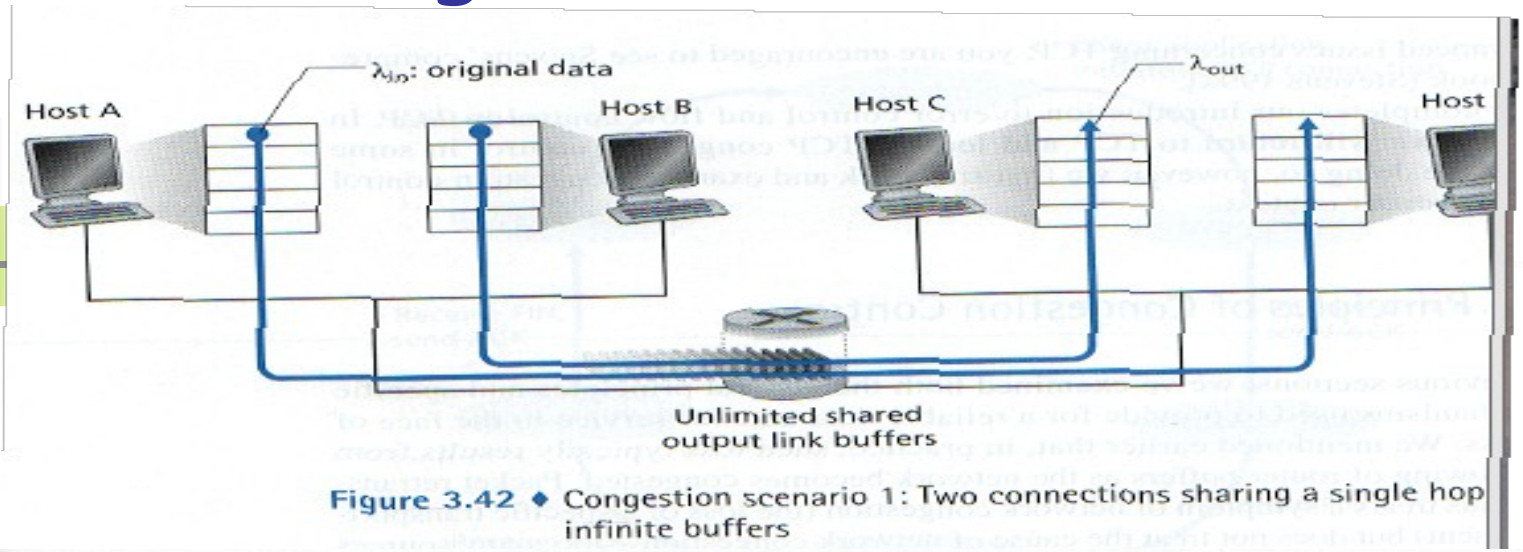
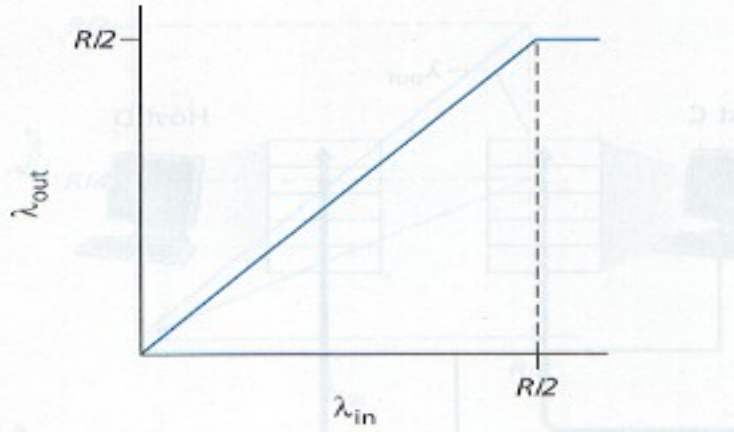


Figure 3.42 ♦ Congestion scenario 1: Two connections sharing a single hop infinite buffers

La durée devient très grande lorsque l'on approche la capacité du canal



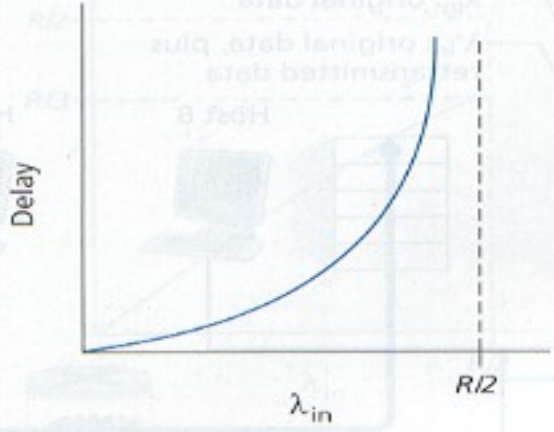


Figure 3.43 ♦ Congestion scenario 1: Throughput and delay as a function of host sending rate

Congestion : cas finitaire

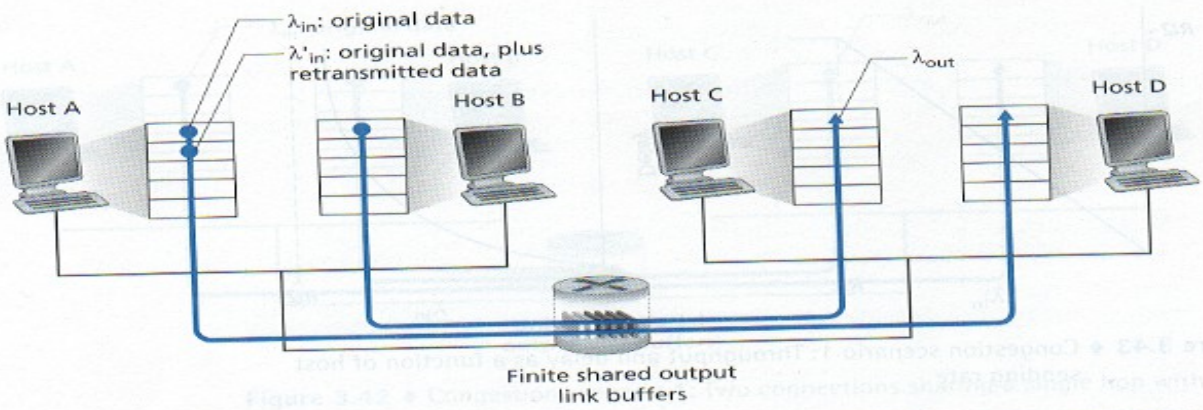


Figure 3.44 ♦ Scenario 2: Two hosts (with retransmissions) and a router with finite buffers.

Le débit utile chute à cause des retransmissions corrigeant les pertes liées au dépassement du canal. L'augmentation des délais cause aussi des retransmissions inutiles sur timeout.

Congestion : cas usuel

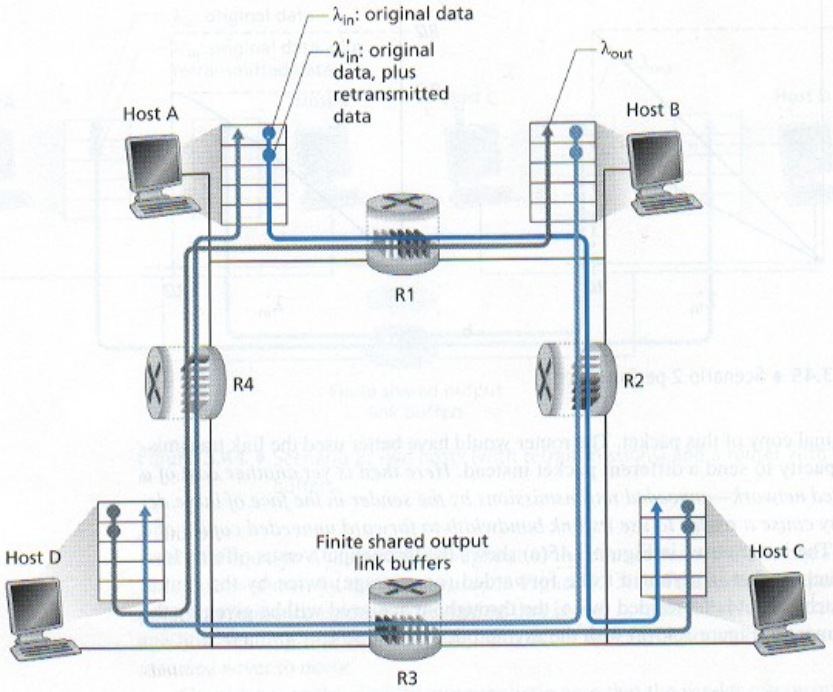


Figure 3.46 ♦ Four senders, routers with finite buffers, and multihop paths

Contention sur le routeur R2 entre le trafic A-C et B-D. Effet "rond-point" : le trafic B-D écroule le trafic A-C en amont. Suggère une priorité souhaitable pour les paquets ayant déjà franchi plusieurs routeurs.

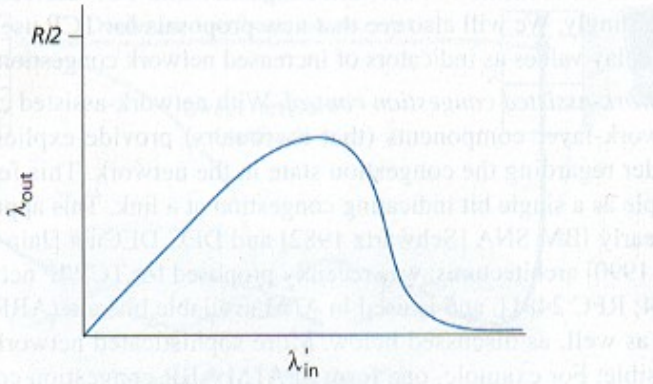


Figure 3.47 ♦ Scenario 3 performance with finite buffers and multihop paths

Le contrôle de congestion

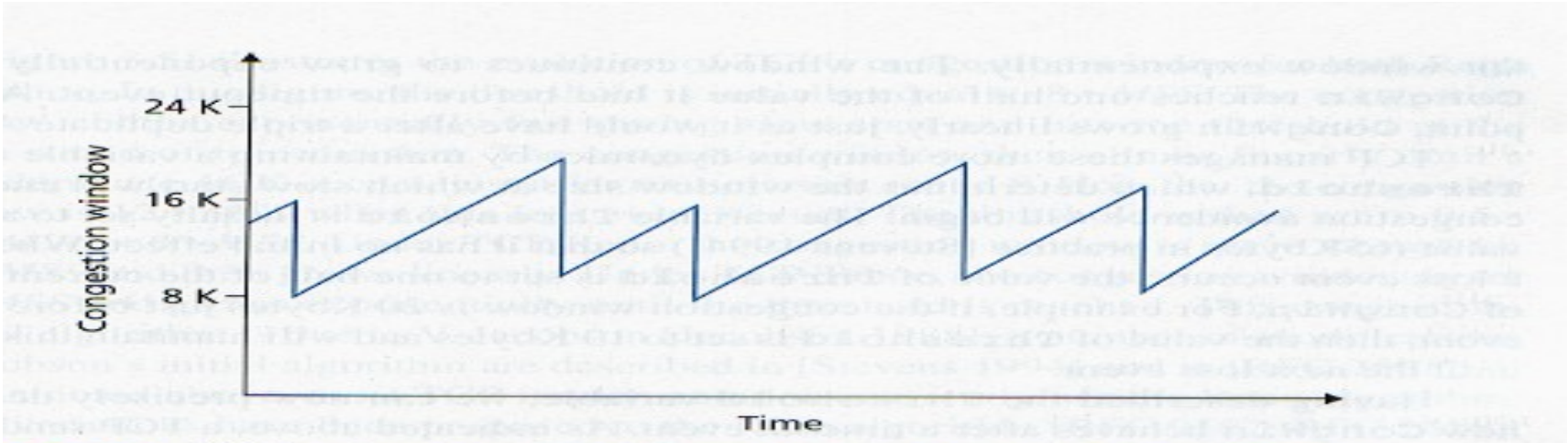
- Contrôle de congestion de bout-en-bout : les couches sous-jacentes n'offrent aucun support, ni mesures : c'est le cas pour TCP qui doit observer la congestion à travers les timeout et les accusés tripliqués -> diminution de la fenêtre
- Contrôle de congestion assisté par le réseau : cas de ATM/ABR, les routeurs informent le niveau supérieur d'une congestion (cellules RM) et ajustement des tampons par un calcul statistique temps-réel complexe

Le contrôle de congestion dans TCP

- L'émetteur maintient une variable supplémentaire, appelée "fenêtre de congestion" (`CongWin`) et assure `LastByteSent - LastByteAcked <= min(CongWin, RcvWindow)`
- En supposant le tampon de réception grand et les retransmissions négligeables, le débit de l'émetteur est approximativement `CongWin/RTT` octet/s
- Le problème de l'algorithme de contrôle est de faire évoluer `CongWin` sur détection de pertes :
  - Incrémentation additive et décrémentation multiplicative,
  - Démarrage lent,
  - Réaction sur timeout

Incrémentation/Décrémentation (algorithme AIMD)

- Les pertes sont "détectées" par timeout ou réception d'accusés tripliqués
- Sur perte, `CongWin` est divisée par 2 sans toutefois descendre en dessous de MSS
- `CongWin` est incrémenté de 1 MSS à chaque RTT (on essaye graduellement…)



3.50 ◆ Additive-increase, multiplicative-decrease congestion control

Démarrage lent (algorithme Slow-Start)

- A l'ouverture de la connexion `CongWin` est initialisé à 1 MSS. Pour éviter une montée trop lente en charge, l'incrémentation est effectuée en doublant en chaque RTT jusqu'à la détection d'une perte. A partir de ce point, c'est l'algorithme AIMD qui est choisi
- La réalité dans TCP est un peu plus complexe : AIMD est la réponse à une réception d'accusés tripliqués, tandis que Slow-Start est utilisé sur timeout, jusqu'à ce que la fenêtre de congestion ait doublé. AIMD prend ensuite le relai

Algorithme de TCP

- L'émetteur gère une variable `Threshold`, initialisée à une grande valeur (65Ko).
- Quand `CongWin <= Threshold`, l'émetteur est dans la phase de démarrage lent et `CongWin` croît exponentiellement
- Quand `ConfWin > Threshold`, l'émetteur est dans la phase d'évitement de congestion et `CongWin` croît linéairement
- Quand un accusé tripliqué est reçu, `Threshold := CongWin/2` et `CongWin := Threshold`
- Quand un timeout arrive, `Threshold := CongWin/2` et `CongWin := MSS` (TCP Reno # TCP Tahoe qui ne possède pas la règle spécifique pour les acks tripliqués)

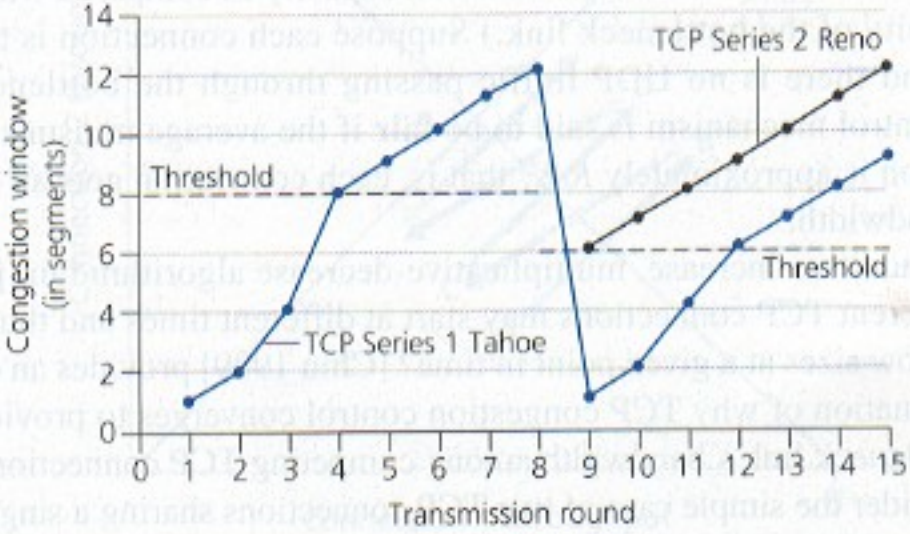


Figure 3.51 ♦ Evolution of TCP's congestion window (Tahoe and Reno)

Conclusion

- TCP est une superposition d'un ensemble de mécanismes simples mais qui donne une algorithmique complexe, sensible à de nombreux paramètres et pour laquelle l'estimation des temps de réponse est difficile
- Celle-ci n'est pas formellement prouvée, mais repose sur de très nombreuses expériences
- Elle continue à évoluer en fonction des nouvelles technologies réseaux et des nouvelles applications

Cours 3 : la couche du niveau réseau

- 3.1 Service de réseau
- 3.2 Routage
- 3.3 Adressage IP
- 3.4 Routage IP
- 3.5 Architecture d'un routeur
- 3.6 IPv6
- 3.7 Diffusion
- 3.8 Mobilité

3.1 Service de réseau : son rôle

- Assure l'acheminement des messages de bout-en-bout -> se doit donc d'être présent dans tous les équipements…
- Le service est simple mais met en œuvre des calculs complexes :
  - La détermination des chemins (routage global)
  - Le relai (routage local : "forwarding")
  - La configuration ("call setup", ATM # IP)

Service avec ou sans connexion ? (datagramme ou circuit virtuel)

- Un circuit virtuel (VC) réserve les ressources et définit un chemin fixe entre extrémités -> l'ouverture et la fermeture d'une connexion encadrent la phase de transfert des données
  - Contrairement à TCP, elle implique la participation de nombreux équipements -> protocoles de signalisation
  - Les routeurs se voient traversés par de nombreux VCs
- ATM, FrameRelay, X25 offrent un service de VC ; Internet n'offre que le "best effort" (datagrammes) -> réordonnancement possible dus à la multiplicité des chemins (cf. Paxson 97 : http://www.acm.org/sigcomm/sigcomm97/papers/p086.html)

Service ATM

- CBR ("Constant Bit Rate") : limite les variations de délais entre émetteur et récepteur à une valeur garantie ("jitter") -> contrôle les pertes et les réordonnancements
- ABR ("Available Bit Rate") : les paquets (cellules de taille fixe) peuvent être perdus mais non réordonnés. Garantit un débit minimum (MCR : "Minimum Cell transmission Rate") et sait indiquer les congestions
- "Best effort" (IP) : aucun contrôle

3.2 Routage

- Calcule un chemin formé de routeurs depuis le routeur source (associé à l'émetteur) jusqu'au routeur destination (associé au récepteur)
- Utilise une abstraction du réseau sous forme de graphe valué
- Le problème est de trouver un chemin de coût minimum -> cf. Cours algo de graphes :
  - Algorithme de routage global ("link state algorithm") : calcul peut être centralisé ou non, mais utilise l'information globale de topologie
  - Algorithme de routage réparti ("decentralized routing algorithm") : le calcul est itératif et n'utilise qu'une connaissance locale accumulée sur les routeurs voisins

Statique/Dynamique

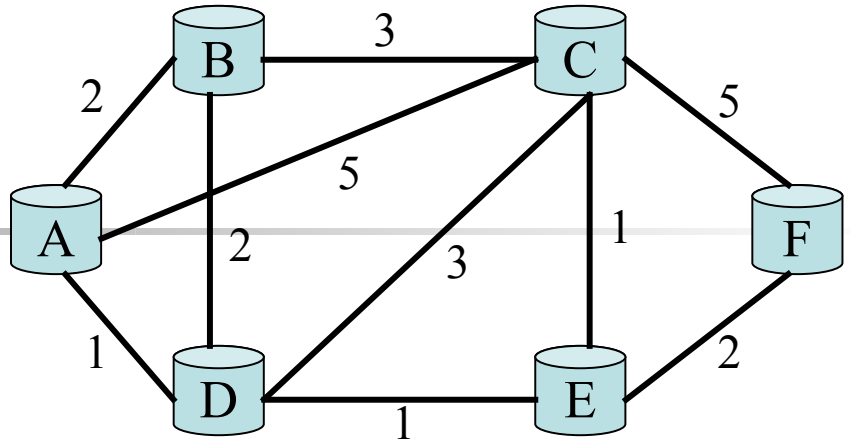
- Statique : les routes sont établies et ne sont recalculées que très peu souvent, en général sur une intervention de l'opérateur
- Dynamique : le routage s'adapte aux changements de topologie et de couts -> attention aux oscillations et boucles
- "Load-sensitive" : les couts varient dynamiquement pour refléter les congestions -> le détournement par routage des zones congestionnées est une solution au contrôle de congestion -> très fragile en pratique
- "Load-insensitive" : IP/RIP, IP/OSPF, IP/BGP

Link state routing (algorithme de Dijkstra)

/* Cas du cout symétrique */

- Init :
  - N = { A } /* A est le nœud source */
  - Pour tout nœud v, si v adjacent à A : D(v) = cost(A,v); p(v) = A
    sinon D(v) = infini
- Répéter jusqu'à tous les nœuds dans N :
  - Sélectionner un w en dehors de N tel que D(w) est minimum
  - Ajouter w à N
  - Pour tout v adjacent à w et en dehors de N :
    si D(v) > D(w) + c(w,v) : D(v) = D(w) + c(w,v); p(v) = w

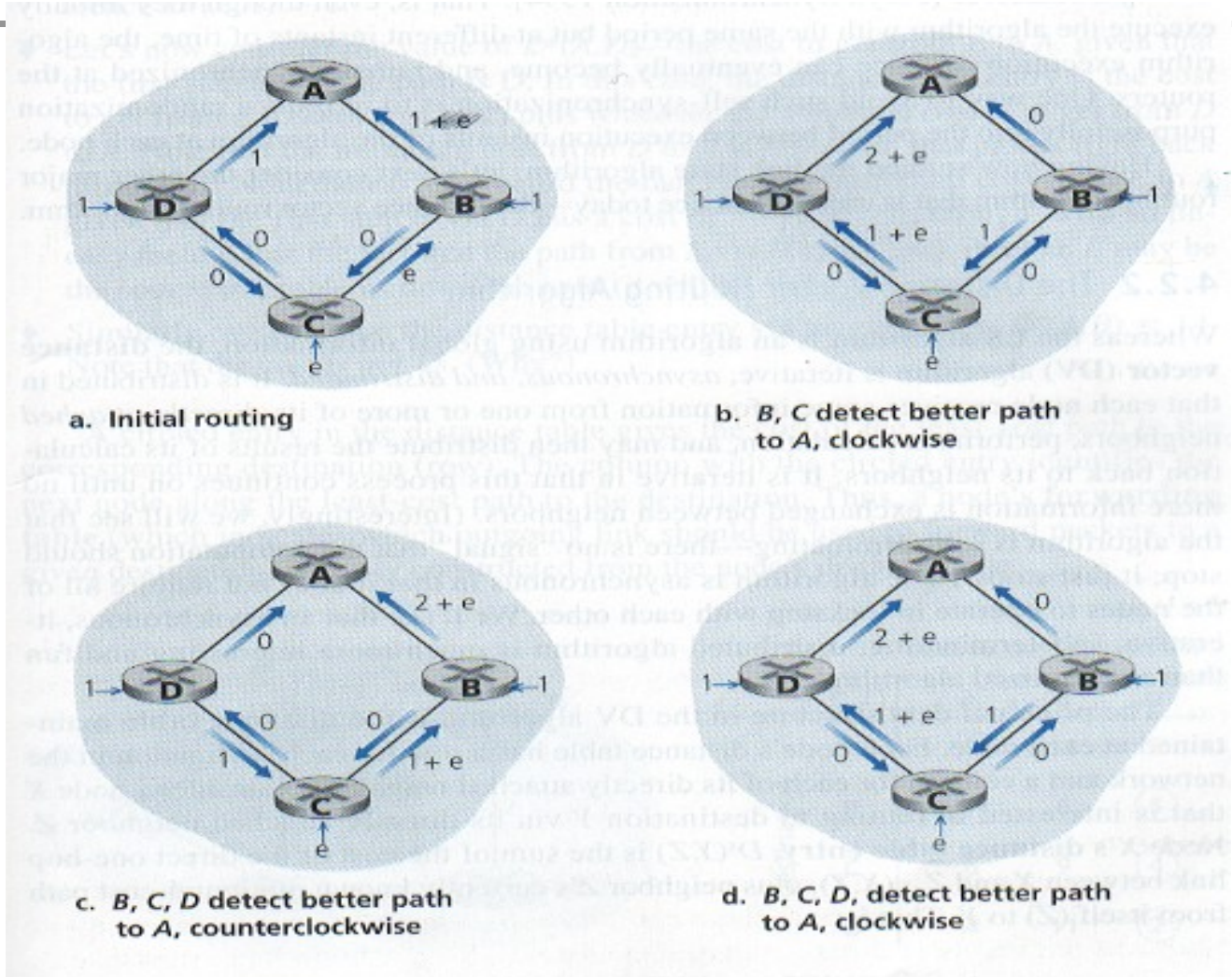
Exemple



Quadratique

pas	N	D(B),p(B)	D(C),p(C)	D(D),p(C)	D(E),p(E)	D(F),p(F)
0	A	2,A	5,A	1,A	oo	oo
1	AD	2,A	4,D		2,D	oo
2	ADE	2,A	3,E			4,E
3	ADEB		3,E			4,E
4	ADEBC					4,E
5	ADEBCF					

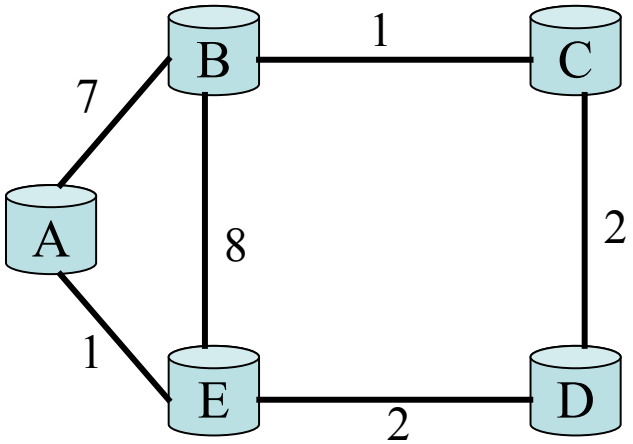
Oscillation possible dans le cas dynamique



Les vecteurs de distance

- Algorithme réparti, auto-stabilisant
- Chaque nœud X maintient une table des distances donnant le voisin Z à utiliser pour joindre la destination Y : $D^x(Y,Z)$
- Calcul réparti de l'invariant suivant :

  $D^x(Y,Z) = c(X,Z) + \min_{w \text{ voisins de } Z}(D^z(Y,w))$
- Lorsqu'un nœud calcule un nouveau cout minimum pour une destination, il informe ses voisins de cette nouvelle valeur



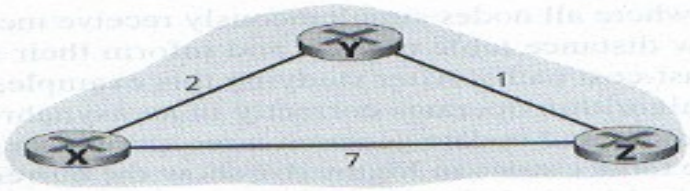
$D^E()$	A	B	D
A	1	14	5
B	7	8	5
C	6	9	4
D	4	11	2

Algorithme de Bellman-Ford

- Init :
  - Pour tout nœud adjacent v: $D^x(.,v)$ = infini; $D^x(v,v) = c(X,v)$
  - Pour tout nœud Y : envoi aux voisins $\min_{w \text{ voisins de } X} D^x(Y,w)$
- Répéter infiniment :
  - Attendre un changement de cout sur un lien d'un voisin v, ou une mise à jour en provenance d'un voisin v
  - Si c(X,v) a été changé par la valeur d :

    pour tout nœud Y : $D^x(Y,v) = D^x(Y,v) + d$

    sinon si réception d'une mise à jour de v pour la destination Y avec la valeur newval : $D^x(Y,v) = c(X,v) + newval$
  - Si $\min_{w \text{ voisins de } X} D^x(Y,w)$ a changé pour une destination quelconque Y : envoi aux voisins la nouvelle valeur de $\min_{w \text{ voisins de } X} D^x(Y,w)$



Node X's table

D^X	cost via Y	cost via Z
dest Y	2	∞
dest Z	∞	7

D^X	cost via Y	cost via Z
dest Y	2	8
dest Z	3	7

D^X	cost via Y	cost via Z
dest Y		
dest Z		

Node Y's table

D^Y	cost via X	cost via Z
dest X	2	∞
dest Z	∞	1

D^Y	cost via X	cost via Z
dest X	2	8
dest Z	9	1

D^Y	cost via X	cost via Z
dest X		
dest Z		

Node Z's table

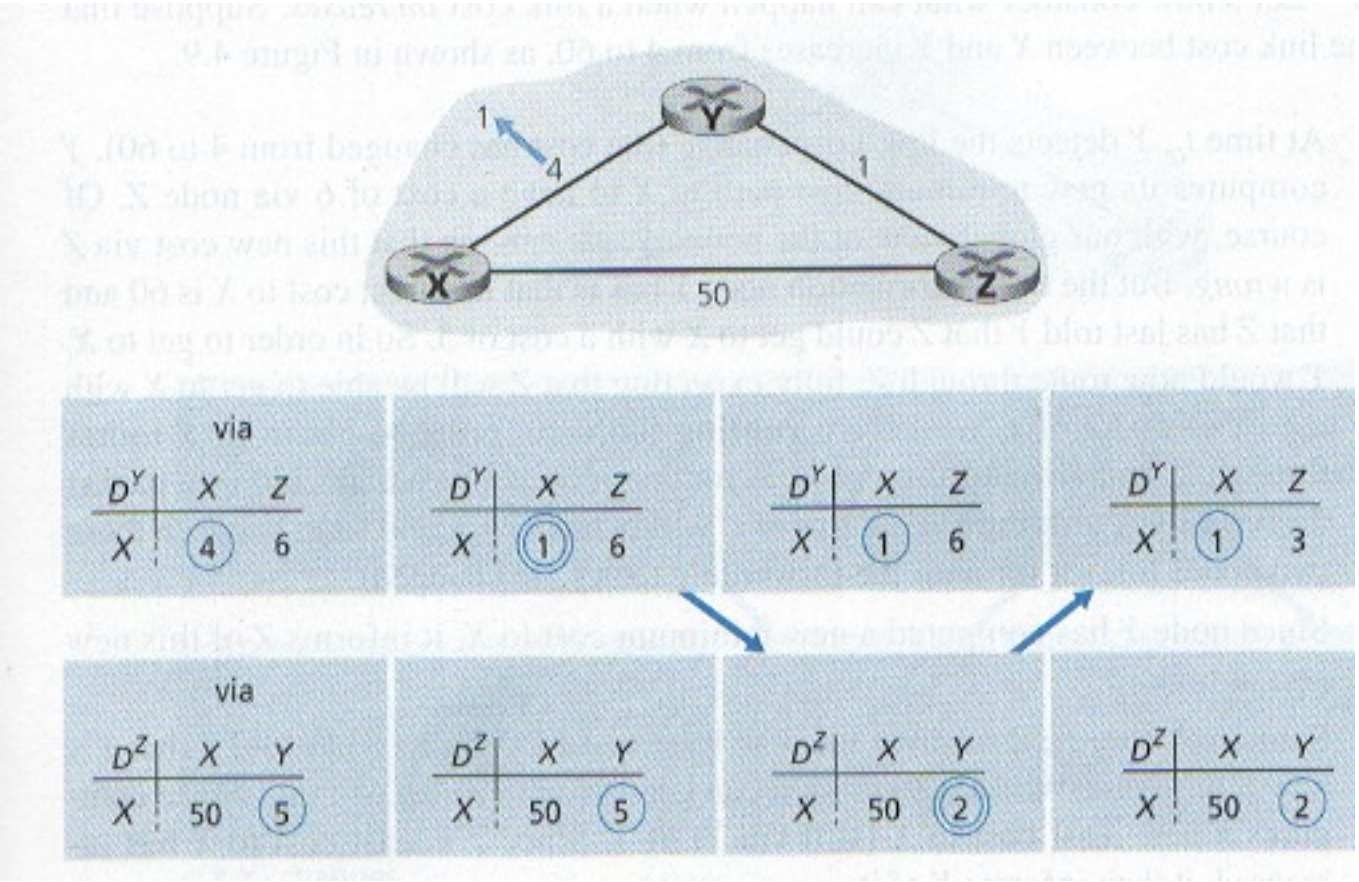
D^Z	cost via X	cost via Y
dest X	7	∞
dest Y	∞	1

D^Z	cost via X	cost via Y
dest X	7	3
dest Y	9	1

D^Z	cost via X	cost via Y
dest X		
dest Y		

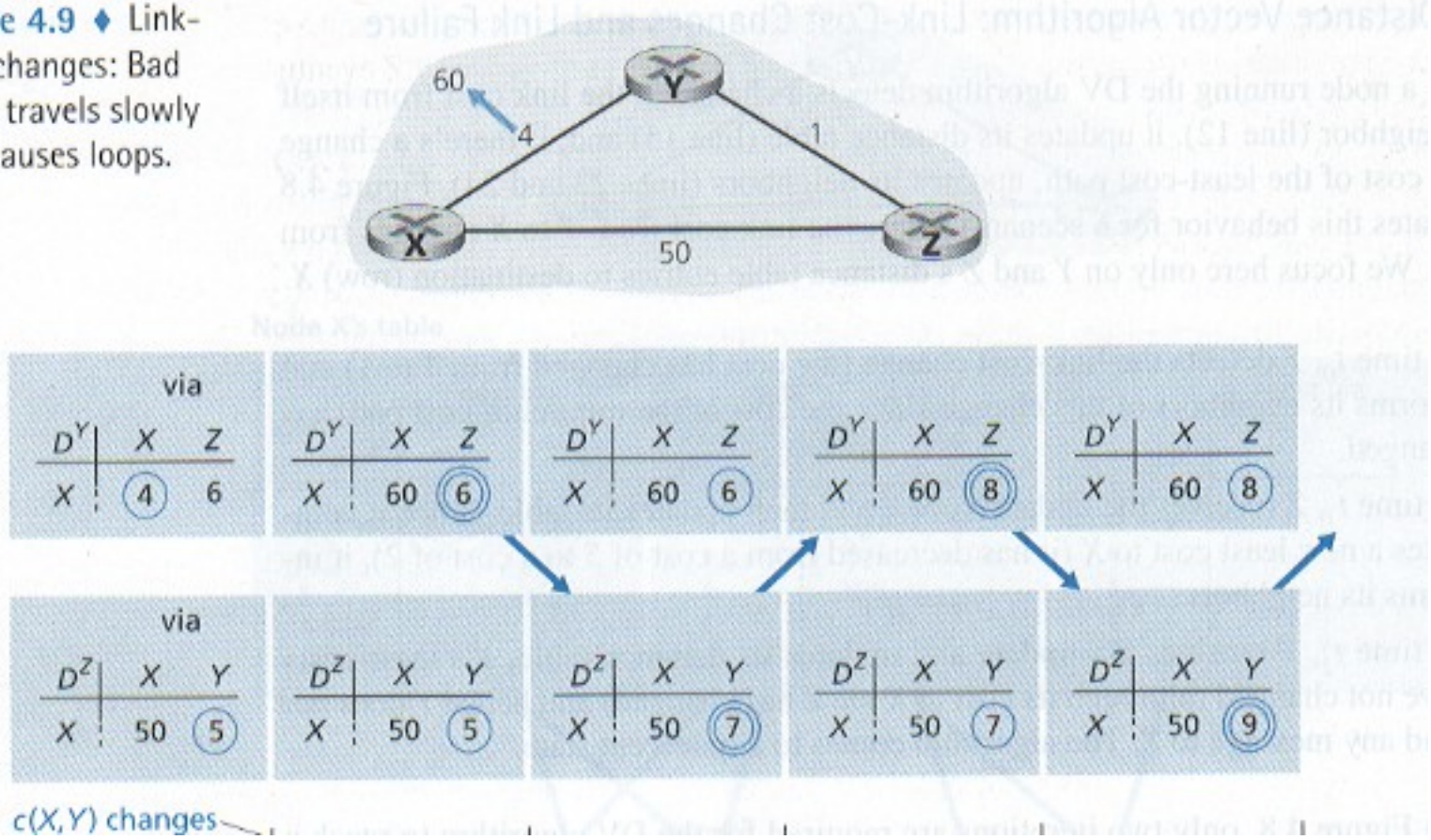
Time

Les bonnes nouvelles vont vite…



Les mauvaises vont lentement…

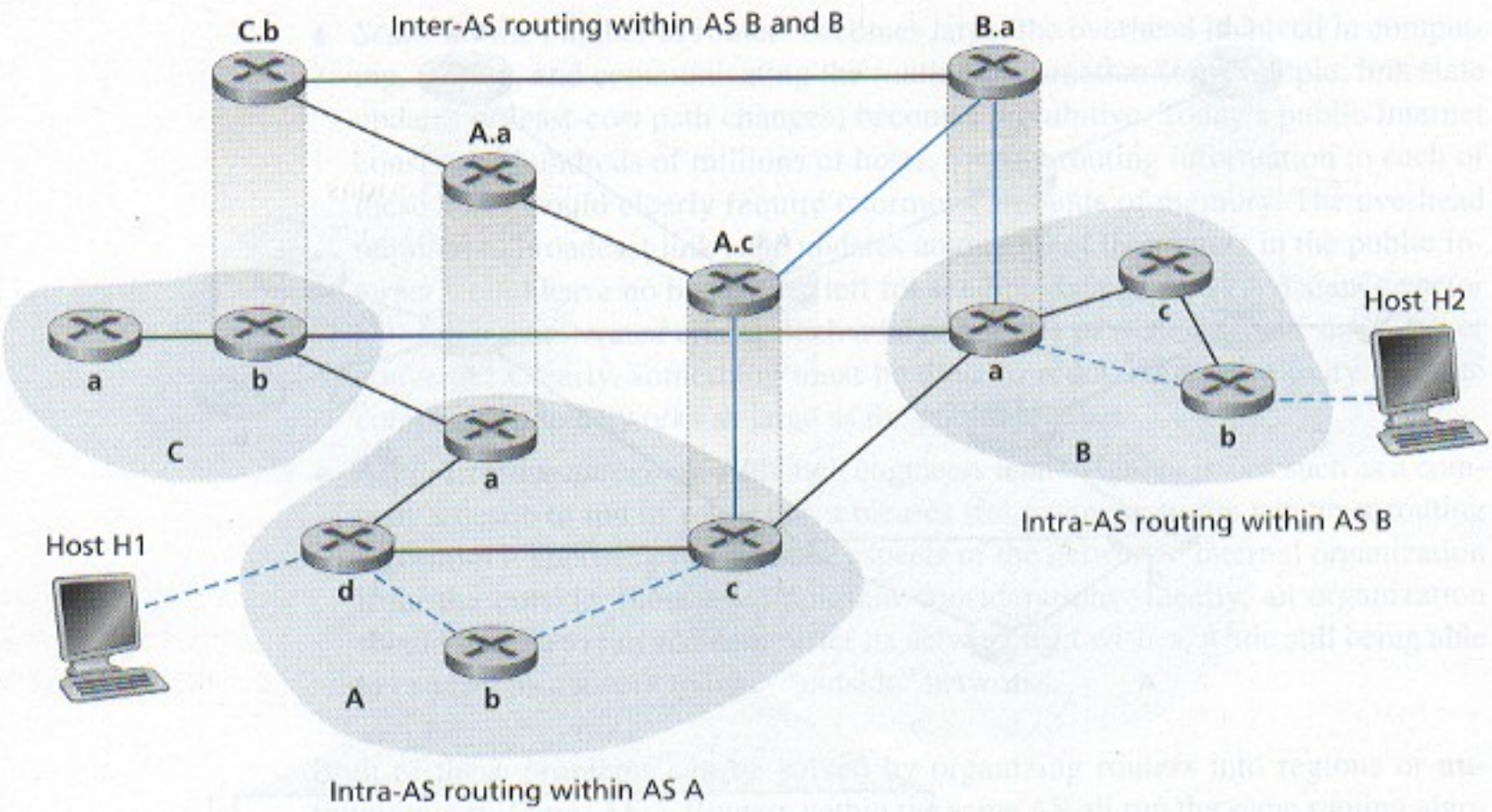
Figure 4.9 ♦ Link-cost changes: Bad news travels slowly and causes loops.



L'avenir ?

- Les deux algorithmes précédents sont ceux utilisés dans l'Internet
- La question générale est un grand problème d'optimisation : le calcul doit-il être réparti sur une connaissance locale ou centralisé sur des nœuds performants spécialisés ?
- En environnement changeant, gagne-t-on vraiment par rapport à l'algorithme trivial de la "patate chaude" qui observe localement la charge des liens sortants
- Mixer un contrôle local approché avec un contrôle global ?
- Doit aussi prendre en compte l'aspect nécessairement hiérarchique du routage (taille et confinement)

Le routage hiérarchique



Le protocole IP (Internet Protocol)

- Adressage : format des paquets et leur traitement dans les routeurs. IPv4 et IPv6
- Détermination des chemins. Calcul des tables de routage : RIP, OSPF, BGP
- Rapport d'erreurs et signalisation : ICMP "Internet Control Message Protocol"

3.3 Adressage IP

- Adresse IP = 32 bits unique dans tout le réseau mondial. Identifie les interfaces des équipements
- Notation pointée décimale : chaque entier décimal est codé par un octet (4 niveaux hiérarchiques donc)
- Ex: 193.32.216.9 ->

  11000001 00100000 11011000 00001001

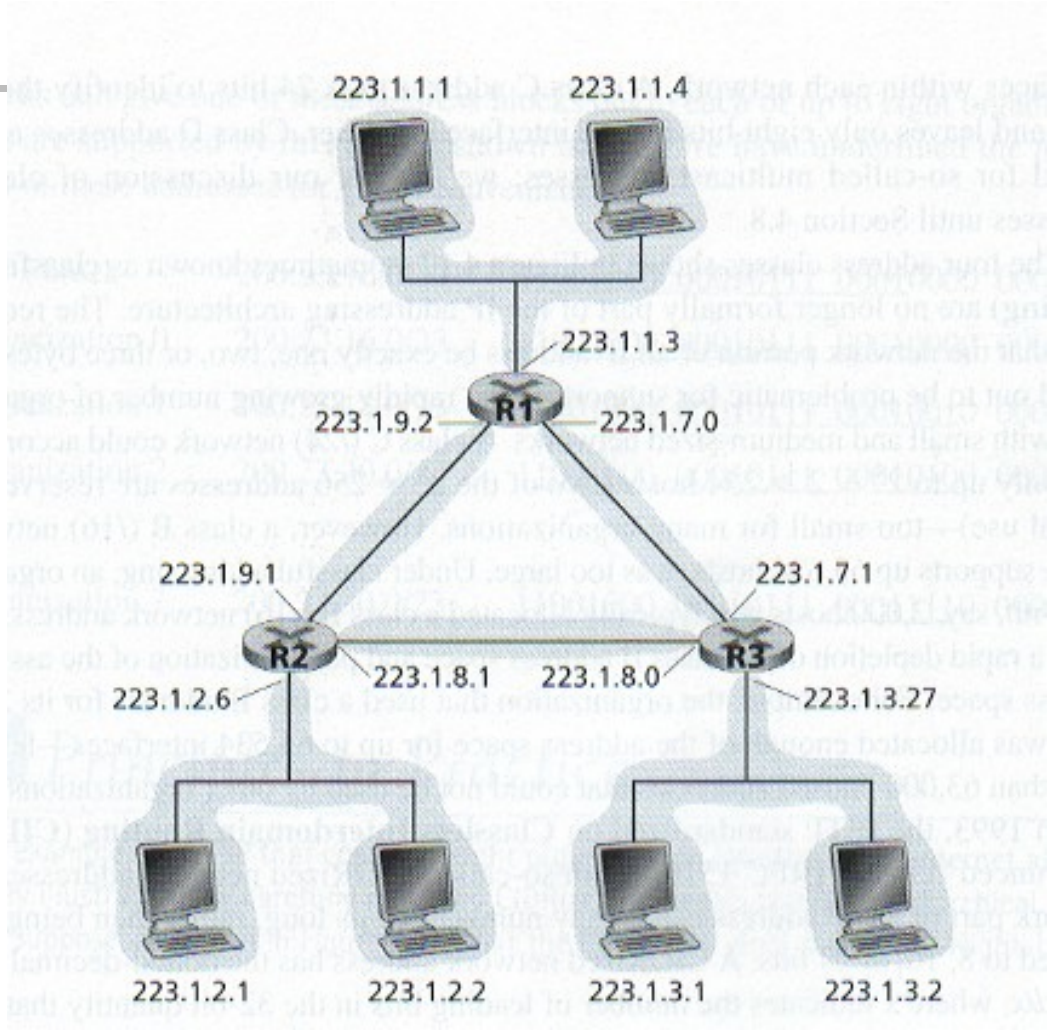
La hiérarchie des ISPs "Internet Service Providers"

- Niveau 1 (cœur de réseau) forme le "backbone" Internet : Uunet (Worldcom), Sprint, AT&T, … Couverture internationale et gros débits
- Niveau 2 : couverture nationale ou régionale. Connectés à un petit nombre d'ISPs cœur. Peuvent être directement connectés à l'intérieur du niveau aussi ("peers")
- Niveau 3 : celui des réseaux locaux des organisations
- Le dernier niveau est celui des machines utilisateurs

-> Codage des adresses sur 4 octets

Adressage vs architecture

- Adresse = réseau/interface
- 6 "réseaux" dans cette architecture possédant entre 2 et 3 interfaces chacun
- Un réseau est défini par un ilôt d'interfaces
- Il y a des millions d'ilôts sur la toile



Les formats d'adresses IPv4

- 4 classes d'adressage :
  - A : 0 net / host / host / host : 1.0.0.0 à 127.255.255.255
  - B : 10 net / réseau / host / host : 128.0.0.0 à 191.255.255.255
  - C : 110 net / net / net / host : 192.0.0.0 à 223.255.255.255
  - D : 1110 adresse de diffusion : 224.0.0.0 à 239.255.255.255
- La classe C ne peut être utilisée que pour 254 hôtes : devenu trop petit pour les organisations, mais la classe B est trop vaste (65534 hôtes) et gaspille des adresses
- En 1997, un nouveau format CIDR ("Classless Interdomain Routing") a été défini pour désigner les adresses des réseaux : a.b.c.d/x, x étant le nombre de bits codant l'adresse réseau dans l'adresse IP

L'obtention des adresses réseaux

- L'administrateur du réseau contacte son ISP qui retourne un bloc d'adresses :
- ISP : 200.23.16.0/20 ->

  200.23.16.0/23 : 11001000 00010111 00010000 00000000

  200.23.18.0/23 : 11001000 00010111 00010010 00000000

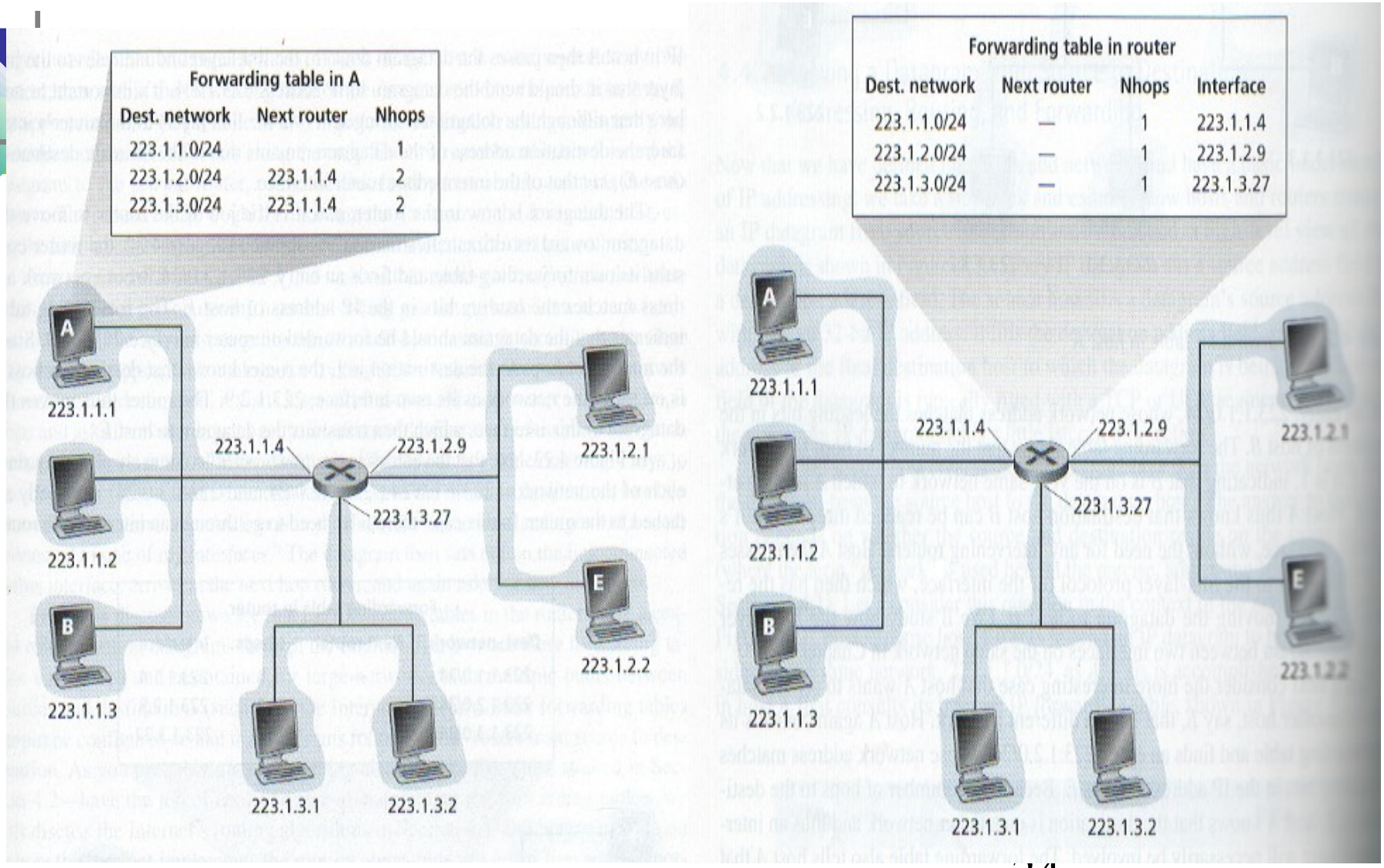
  200.23.20.0/23 : 11001000 00010111 00010100 00000000

  200.23.30.0/23 : 11001000 00010111 00011110 00000000
- Comment l'ISP obtient-il son adresse ?
- -> ICANN ("Internet Corporation for Assigned Names and Numbers"), gère aussi les serveurs DNS racines

L'obtention des adresses des machines

- Configuration manuelle : changement d'un fichier de config
- Configuration dynamique : DHCP ("Dynamic Host Configuration Protocol"). Permet d'acquérir automatiquement une adresse IP, mais aussi les adresses du premier routeur et du serveur DNS. DHCP peut être configuré pour allouer toujours la même adresse IP à une machine donnée. Mais le plus souvent cette adresse est temporaire car il y a plus de machines que d'adresses disponibles dans l'organisation

Les tables de routage

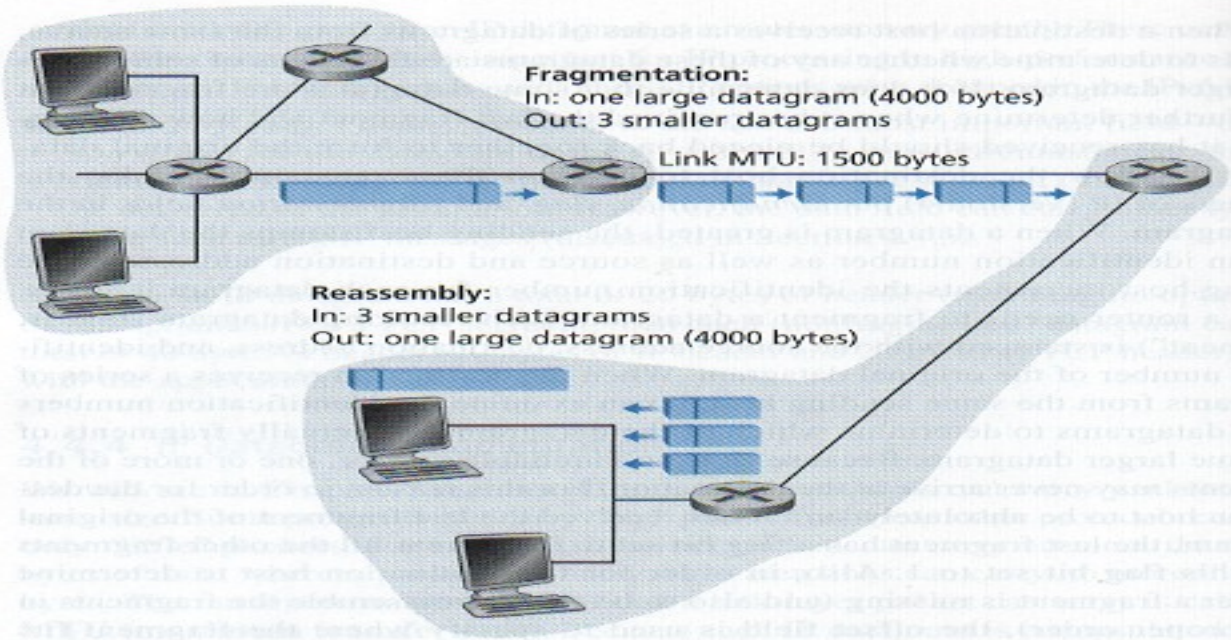


Format des datagrammes IPv4

- Version number : 4 bits
- Header length : 4 bits (taille variable liée aux options), 20 octets min
- Type of service : 8 bits (pour un service différencié)
- Datagram length : 16 bits (65535 octets max, 576 en pratique)
- Identifiers, flags, fragmentation offset : 32 bits (fragmentation -> voir plus loin)
- Time-to-live : 8 bits (décrémenté à chaque passage de routeur. Le datagramme est détruit lors du passage à 0)
- Protocol : 8 bits (à destination, indique le protocole du dessus : TCP/UDP)
- Header checksum : 8 bits (pour protection entre routeurs)
- Source and destination IP addresses : 32 bits chacune
- Options : taille variable -> enlevé dans IPv6
- Data

Fragmentation

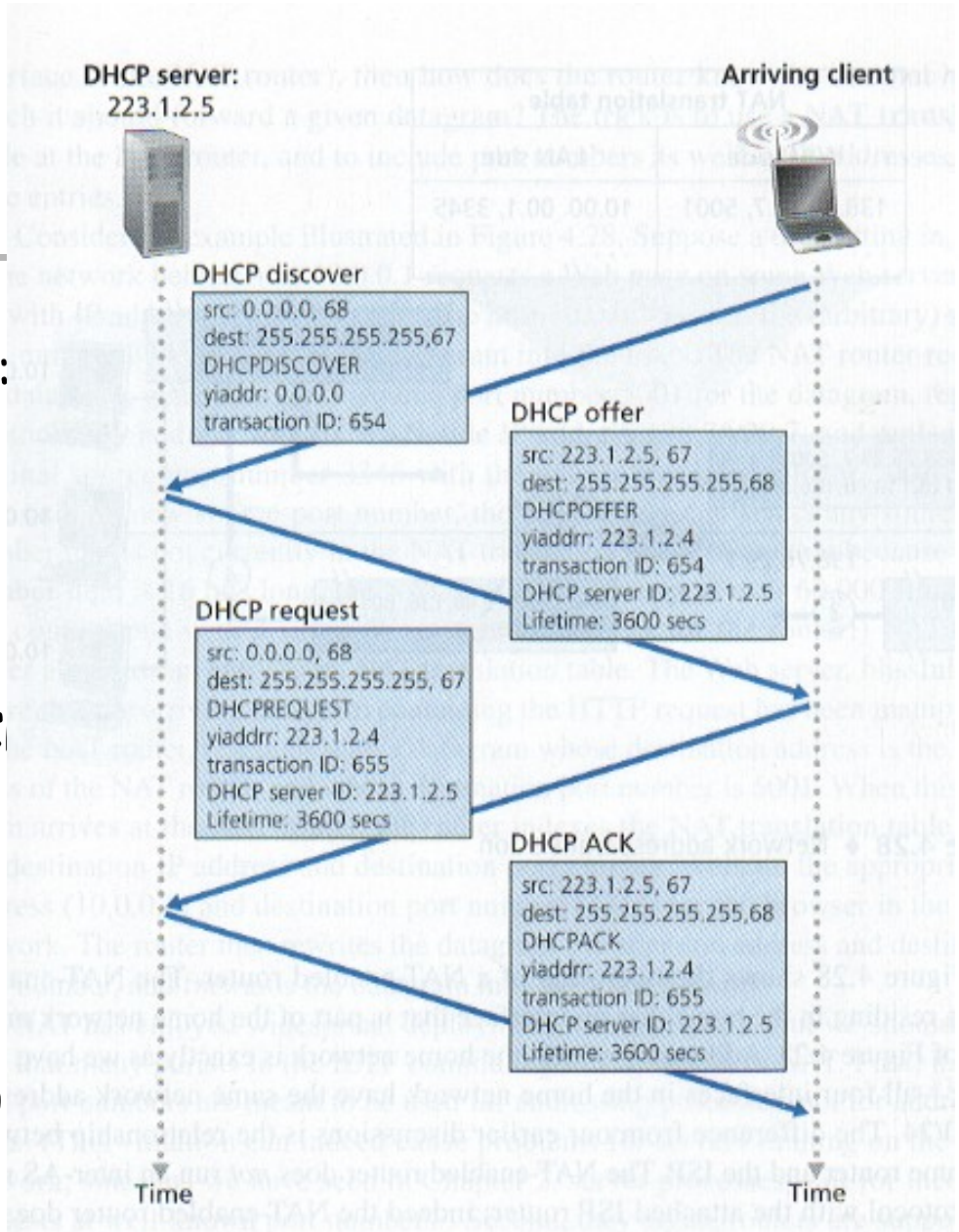
- Pb : tous les protocoles liens du réseau n'implantent pas forcément des datagrammes de même taille (MTU "Maximum Transfer Unit"). Lorsqu'un routeur reçoit un gros paquet qu'il doit retransmettre dans des petits, on doit fragmenter
- Le travail de réassemblage n'est confié qu'à l'extrémité pour ne pas charger les routeurs
- A la création d'un datagramme, l'émetteur estampille le message avec un numéro d'identification incrémenté à chaque envoi
- Lors de la fragmentation, l'estampille est conservée pour permettre le réassemblage de datagrammes de même id
- Pour se prémunir des pertes, un flag sert à distinguer le dernier fragment
- Pour traiter les réordonnancements, le champ offset sert à spécifier l'emplacement des fragments dans l'original



Fragment	Bytes	ID	Offset	Flag
1st fragment	1,480 bytes in the data field of the IP datagram	identification = 777	offset = 0 (meaning the data should be inserted beginning at byte 0)	flag = 1 (meaning there is more)
2nd fragment	1,480 byte information field	identification = 777	offset = 1,480 (meaning the data should be inserted beginning at byte 1,480)	flag = 1 (meaning there is more)
3rd fragment	1,020 byte (= 3,980–1,480–1,480) information field	identification = 777	offset = 2,960 (meaning the data should be inserted beginning at byte 2,960)	flag = 0 (meaning this is the last fragment)

DHCP

- Protocole d'application : DHCP/UDP/IP
- Utilisation de l'adresse IP de diffusion dans le réseau 255.255.255.255
- Plusieurs serveurs DHC peuvent répondre
- Notion de bail DHCP
- L'adresse IP obtenue contient l'adresse réseau (pb pour la mobilité inter-réseaux)



3.4 Routage IP

- L'existence de plusieurs protocoles de routage possibles fait que l'on travaille par domaine
- On distingue donc le routage :
  - Intra-domaine : RIP "Routing Information Protocol" et OSPF "Open Shortest Path First"
  - Inter-domaine : BGP "Border Gateway Protocol" entre passerelles

RIP

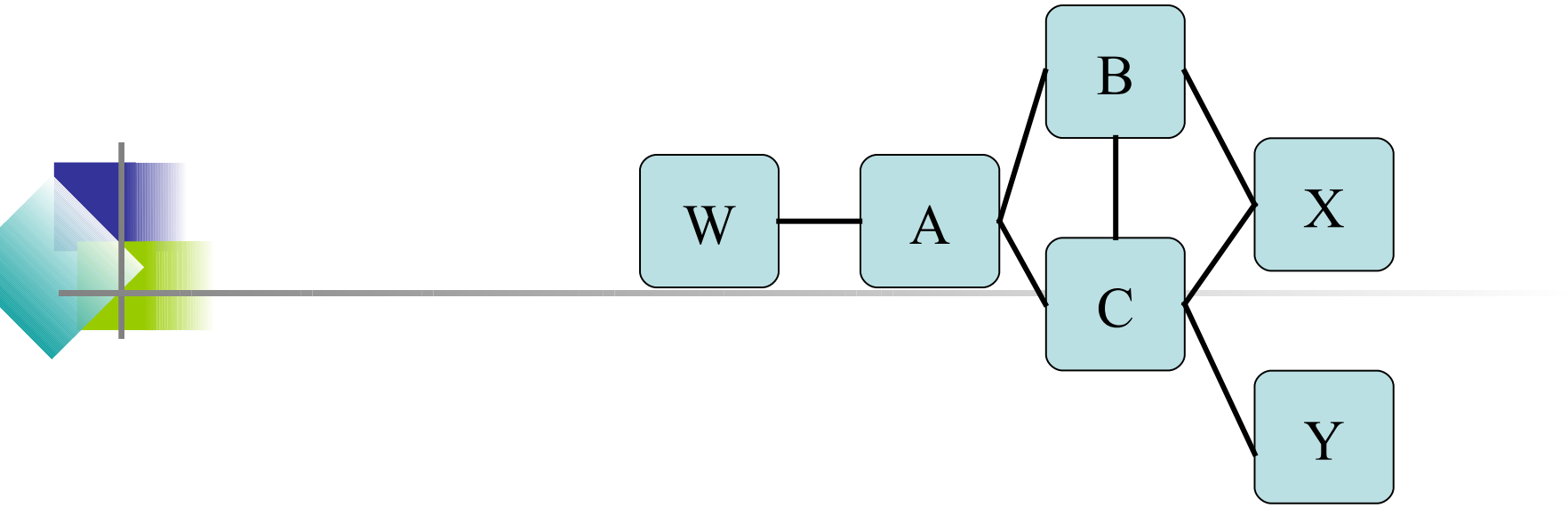
- Principe des vecteurs de distance
- Modèle de cout ultra-simple : on compte de nombre de liens (un lien coûte 1)
- Les tables de routage sont mises à jour entre voisins toutes les 30" environ (pour prendre en compte les changements de topologie éventuels)
- Si un routeur ne reçoit rien d'un voisin en 180", ce voisin est déclaré en panne et l'information de changement de topologie est transmise
- Curieusement, RIP est mis en œuvre au dessus de UDP (lui-même au dessus de IP !)

OSPF

- Link-state protocol (Dijkstra)
- Construit le graphe dirigé complet du domaine
- Le coût des liens est positionné par l'administrateur du domaine
- L'information de routage est diffusée à l'ensemble des routeurs
- L'algorithme est relancé dès qu'il y a un changement détecté par un routeur. Il est aussi relancé systématiquement toutes les 30'
- OSPF utilise IP, donc doit se préoccuper du transfert fiable et de la diffusion (protocole complexe de fait)
- D'autres facilités :
  - Authentification des routeurs
  - Autorise l'utilisation de liens multiples de même coût
  - Permet de hiérarchiser le domaine (routeurs d'interface)

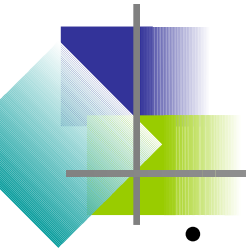
BGP

- Connecte les routeurs d'interface (passerelles)
- Utilise le principe des vecteurs de distance
- Les messages inter-domaine sont routés par BGP. Une fois arrivés sur un domaine, ils sont pris en charge par le routage intra-domaine
  - Un routeur BGP reçoit des messages de routes venant d'un routeur BGP voisin. Il peut aussi faire du filtrage (pour éviter les boucles par exemple)
  - Il sélectionne la meilleure route si il connaît plusieurs possibilités
  - Il propage les informations de route



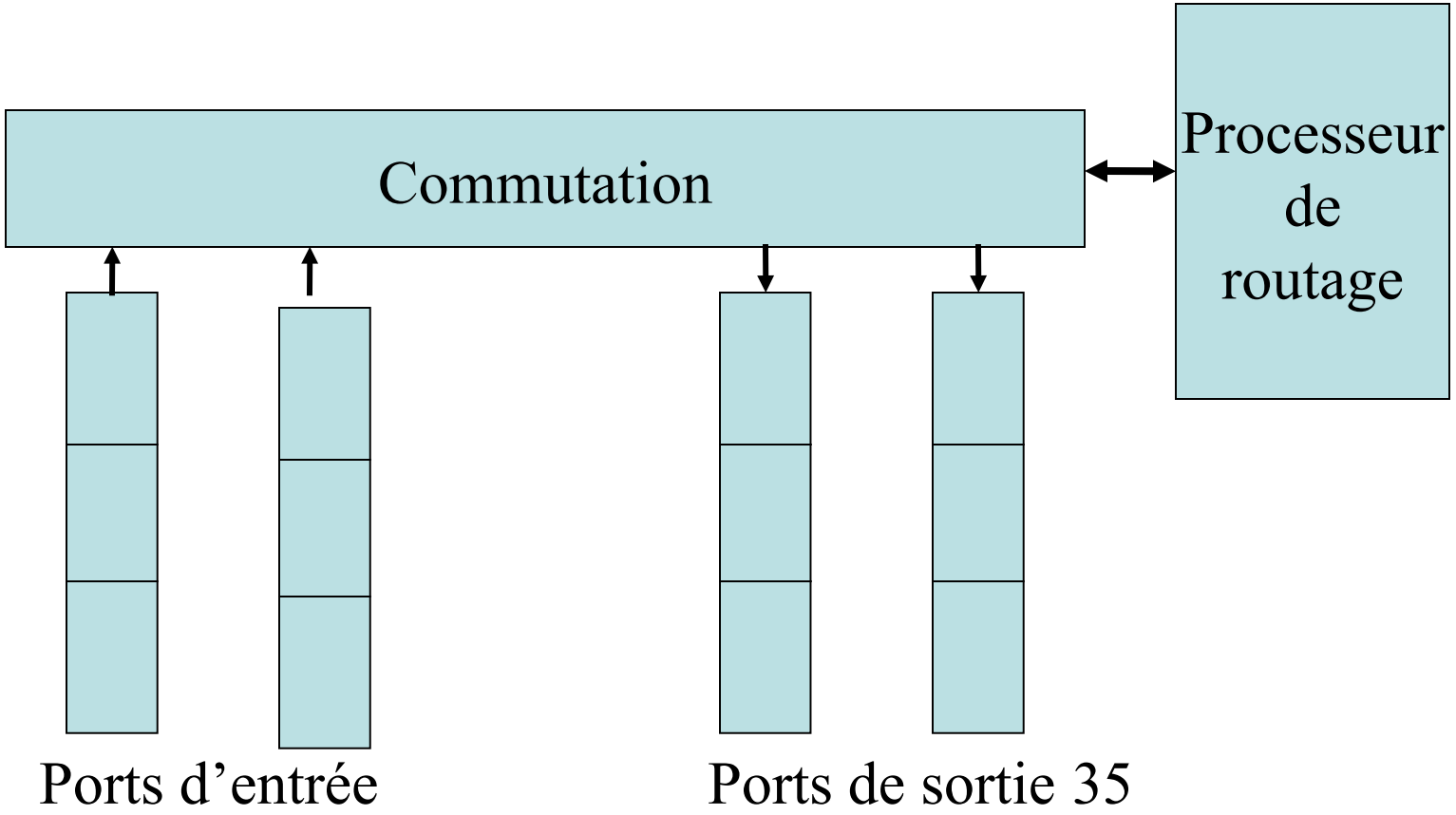
- Exemple de réseau de réseaux
- Supposons que B a appris de A que A a un chemin AW pour aller à W
- B a donc installé dans sa table le chemin BAW et prévient X qu'il peut aller à W via B
- Doit-il prévenir C ? -> accords secrets entre ISP

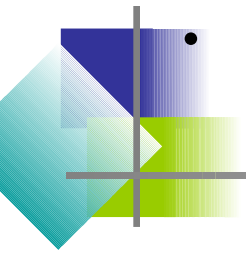
BGP utilise TCP (port 179)



- Les messages BGP :
  - Open : premier contact avec un routeur BGP voisin (authentification + date)
  - Update : envoi d'un nouveau chemin. Permet aussi de supprimer un chemin
  - Keepalive : sert à avertir ses voisins de sa bonne santé, sans envoyer de nouvelle information de chemin. Sert d'accusé pour le message Open
  - Notification d'erreur

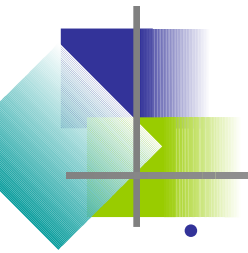
3.5 Architecture d'un routeur





- Le processeur de routage maintient les tables, mais pour des raisons de performances, une copie peut être utilisée sur le processus associé à un port d'entrée (le décapsulage et l'examen du message d'entrée doit se faire à haut débit)
- Utilisation d'arbres de recherche pour augmenter la performance de la recherche dans la table : exemple de la recherche binaire guidée par la suite de bits de l'adresse de destination
- Pas toujours suffisant -> utilisation de mémoires adressables par le contenu
- Les messages non encore routés doivent attendre

Commutation

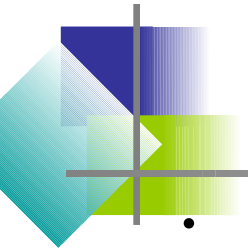


- Par mémoire : l'arrivée d'un message déclenche une interruption ; le processeur copie alors le message en mémoire ; il extrait l'adresse de destination ; regarde dans la table de routage ; recopie le message dans le tampon de sortie correspondant
- Par bus : les ports de sortie attrapent les messages qui leur sont destinés. Le bus est le goulet d'étranglement
- Par réseau d'interconnexion (crossbar, delta, omega)

Politiques de service

- Les messages en attente d'envoi dans les ports de sortie peuvent être traités en fonction de politiques différentes -> FCFS (problèmes de blocage), WFQ ("weighted fair queuing"), QoS
- Les messages en entrée peuvent être :
  - Jetés en cas de tampon plein,
  - Prendre la place d'un message déjà en attente (QoS)
  - Jetés lorsqu'on approche la fin du tampon pour prévenir plus rapidement de la congestion
- -> "Active Queue Management" : exemple de l'algorithme RED ("Random Early Detection") : le message est admis dans le tampon de sortie si la taille moyenne du tampon est en dessous d'un seuil minimum. Si la taille moyenne est supérieure au seuil maximum (ou si le tampon est plein) le message est perdu. Sinon le message est perdu avec une probabilité dépendant des seuils.

3.6 IPv6



- Les principaux changements :
  - Des adresses IP de 128 bits
  - Adresse "anycast" (1 parmi un groupe : gestion de serveurs miroirs par exemple)
  - Entête de longueur fixe (40 octets)
  - Possibilité de marquer un flot (priorité, service différencié)
- Les conséquences :
  - Pas de fragmentation/réassemblage : l'émetteur doit se débrouiller. Un paquet trop long pour un routeur sera jeté avec avertissement "Packet too big" via ICMP
  - Pas de checksum sur l'entête (on fait confiance au niveau du dessous, améliore les performances)

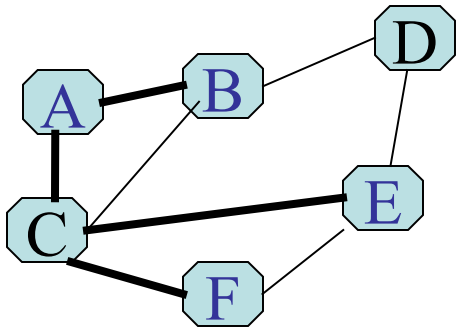
Transitoire IPv4-IPv6

- Le basculement global et simultané n'est plus possible (NCP->TCP il y a 20 ans a déjà été difficile)
- Approche "Dual-stack" : les nœuds implantent à la fois IPv4 et IPv6. L'adresse du destinataire retournée par DNS indiquera si le destinataire accepte IPv6. On échangera en IPv6 ssi l'émetteur et le récepteur sont en IPv6
- Approche "Tunneling" : le paquet IPv6 est enfoui dans un paquet IPv4 pour passer de façon transparente via des routeurs IPv4
- La transition est annoncée durer 15 ans ! -> réelle difficulté d'introduire des changements dans le niveau lien qui touche des milliards de machines -> apparition de très nombreux protocoles applicatifs en réaction

3.7 Le routage pour la diffusion

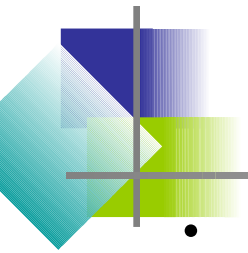
- Plusieurs solutions possibles :
  - Utilisation de plusieurs connexions par l'émetteur
  - Le récepteur rediffuse le message à d'autres récepteurs du groupe et ainsi de suite…
  - Le réseau de routeurs prend en charge la diffusion en essayant d'utiliser au mieux les ressources liens
    - Le groupe est identifié par une adresse IP multicast unique
    - Les récepteurs finaux seront accédés par une indirection (les routeurs intermédiaires connectés aux destinataires reconnaîtront que ceux-ci sont dans le groupe demandé)
    - -> IGMP ("Internet Group Management Protocol")

IGMP



- Intervient entre la machine d'extrémité et le routeur auquel elle est connectée
- La coordination entre routeurs multicast est effectuée par un autre protocole : PIM, DVMRP, MOSPF
- Les machines d'extrémité s'abonnent et se désabonnent en dialoguant via IGMP avec le routeur
- Diffusion d'un membre du groupe à tous les autres : technique de l'arbre couvrant du groupe
- Problème de Steiner : trouver un arbre de coût minimum -> NP complet, mais il existe de bonnes heuristiques
- Autre algorithme : trouver un centre du groupe à partir duquel les messages seront re-acheminés

3.8 Mobilité



- Routage indirect : envoi à l'adresse fixe qui transmet à l'adresse mobile (encapsulation de l'adresse finale pour décapsulation par le routeur auquel s'est connecté le mobile) -> inefficacité du triangle de routage
- Routage direct : l'émetteur doit avoir un routeur agent qui par dialogue avec l'adresse fixe, va récupérer l'adresse mobile
- Mobile IP : combine ces facilités dans un protocole plutôt complexe

Cours 4 : le réseau local (lien)

- 4.1 Service de lien
- 4.2 Détection et correction d'erreurs
- 4.3 Protocoles à accès multiple
- 4.4 Adressage dans un réseau à diffusion
- 4.5 Ethernet
- 4.6 Sans fil

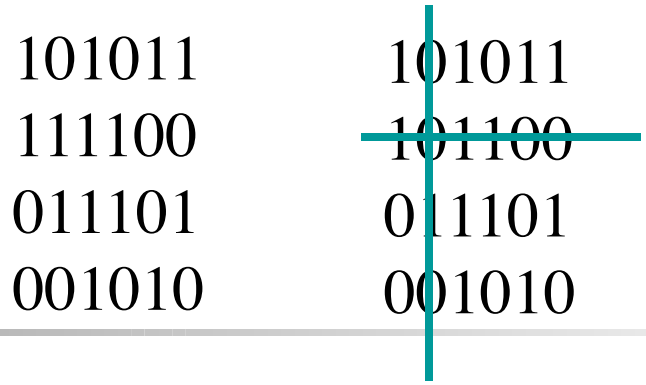
4.1 Service de lien

- Entre deux nœuds du réseau : les trames (paquets/datagrammes/trames)
- Mis en œuvre dans un coupleur, généralement matériel, avec ses propres processeurs (ex. Carte PMCIA)
- Offre des services comparables au transport de bout en bout (contrôle des erreurs, du flux, …)

4.2 Détection et correction d'erreurs

- Au niveau bit : changement de valeur possible sur le canal physique (pas de perte car synchro à ce niveau là)
- EDC ("Error Detection and Correction") bits
- (D + EDC) -> (D' + EDC') : la probabilité d'erreurs non détectées n'est pas nulle : dépend de la technique utilisée et de l'overhead admissible
- Parité / Checksum / Codes cycliques

Parité



- EDC = 1 bit
- Nb 1 dans D+EDC pair
- Si le récepteur en compte un nombre impair, il sait qu'il y a eu au moins 1 bit erroné
- Si le nombre d'erreurs est pair, aucune erreur n'est détectée : la question est la probabilité de plusieurs erreurs dans une même trame -> problème des phénomènes de rafales
- Généralisation à 2 dimensions -> parité ligne et parité colonne -> permet de corriger 1 erreur bit (le bit qui provoque une erreur de parité à la fois sur la ligne et la colonne) et de détecter 2 erreurs bit

Checksum

- Internet (UDP/TCP) : les données sont traitées comme une suite d'entiers sur 16 bits et sont additionnés. Le complément à 1 de la somme forme le checksum transmis dans la trame
- Le récepteur calcule la somme des entiers reçus. Si un 0 apparaît, une erreur est détectée

0110011001100110	0110011001100110
0101010101010101	0101010101010101
1011101110111011	0000111100001111
0000111100001111	0011010100110101
1100101011001010	1111111111111111

"Cyclic Redondancy Check" (CRC)

- Les suites de bits sont vues comme les coefficients d'un polynôme. Les données sont transmises avec un code dépendant d'un générateur G de r+1 bits, R = reste de ($D \times 2^r / G$)
- On transmet ($D \times 2^r$) xor R
- Le récepteur divise la donnée reçue par G. Si le reste est différent de 0, une erreur est détectée
- Détecte des rafales d'erreurs de moins de r+1 bits. Au delà la probabilité de détection est $1-0.5^r$

- IEEE G =

  10000010011000001000111011011011 1

D = 101110 G = 1001

```
101110000 | 1001
  1010    | 101011
    1100
     1010
      011
```

4.3 Protocoles à accès multiple

- Canal de diffusion partagé : cable Ethernet, radio sans-fil (Wifi), satellite, …
- Collisions : les signaux des trames se mélangent, rendant les trames inaudibles pour les différents récepteurs -> toutes les trames en collision sont perdues…
- Coordination nécessaire entre les émetteurs -> rôle du protocole d'accès :
  - Protocoles de partitionnement du canal,
  - Protocoles à accès aléatoire,
  - Protocoles tournants

Protocoles à partitionnement

- Trois méthodes :
  - Division temporelle ("Time Division Multiplexing" TDM) : le train de bits est divisé en parts ("slots") allouées statiquement pour chaque récepteur : 123412341234. L'émetteur attend son slot. Typiquement la taille du slot permet de loger un paquet

    -> sans collision et équitable, mais débit R/N même si un seul émetteur + attente
  - Division fréquentielle ("Frequency Division Multiplexing" FDM) : la bande de fréquence est divisée en N -> fournit N sous-canaux de débit R/N : mêmes avantages et inconvénients que la technique TDM

- Division par code
("Code Division Multiple Acces" CDMA)
  - Chaque bit émis est codé avec un mot c de longueur M
  - En représentant le bit 0 par -1, le bit de donnée d_i est transformé en la suite de bits $Z_{i,m}$ pour m de 1 à M : $Z_{i,m} = d_i \times c_m$
  - Le récepteur retrouve le bit de donnée par l'opération :

    $d_i = 1/M \sum_{m=1,M} Z_{i,m} \times c_m$
  - Exemple : C = 111-11-1-1-1

    -1 1 -> -1-1-11-1111 111-11-1-1-1 -> -1 1

Traitement des interférences : N émetteurs

- Les valeurs de chaque émetteur s'ajoutent sur un mini-slot : $Z_{i,m}^{*} = \sum_{s=1,N} Z_{i,m}^{s}$
- Astucieusement, on peut démêler par la même formule que précédemment les valeurs à la réception si les mots de code de chaque émetteur sont bien choisis :

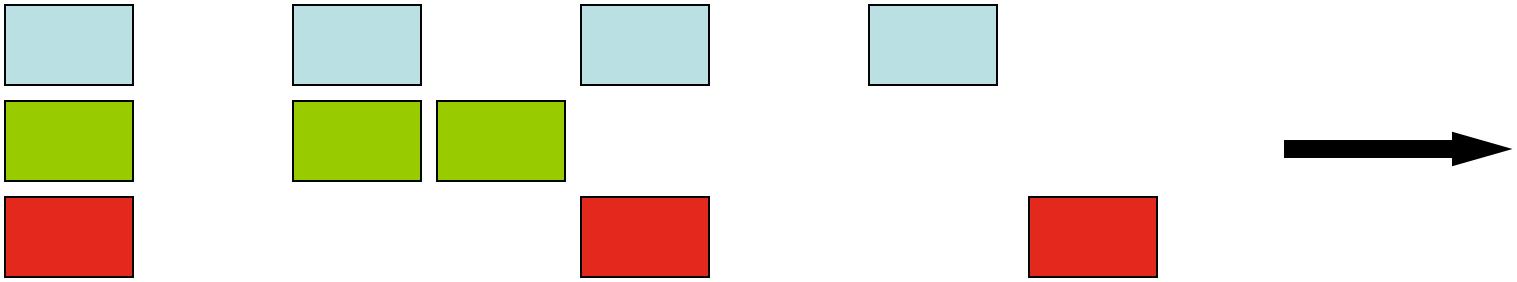
  $d_i = 1/M \sum_{m=1,M} Z_{i,m}^{*} \times c_m$
- Outre le choix des codes, il y a aussi des difficultés pratiques dans la régulation de puissance électrique sur le canal
- Exemple : c1 = 111-11-1-1-1, c2 = 1-1111-111

  -1 1 // 1 1 -> 0-2020022 20202-200 -> -1 1 (récepteur 1)

Accès aléatoire

- Transmission au débit max, mais retransmission en cas de collision avec une attente aléatoire (indépendante pour chaque émetteur) -> des centaines de protocoles (Aloha/CSMA/Ethernet …)
- Exemple le plus simple : "Slotted ALOHA"
  - Trames de L bits, débit du canal R
  - Le temps est divisé en slots de taille L/R secondes
  - Les nœuds ne transmettent qu'en début de trame. Ils sont synchronisés et connaissent donc tous le même début de trame
  - Tous les nœuds détectent une collision avant la fin du slot
  - En cas de collision, la trame est retransmise dans des slots prochains avec une probabilité p

Slotted ALOHA



- Performances : à charge max, N émetteurs
- Probabilité de succès pour un nœud arbitraire : $Np(1-p)^{N-1}$
- Choix du meilleur p pour N arbitrairement grand :
  p = 1/e (0.37)
- 37% seulement du débit canal est utilisé !

Pure ALOHA

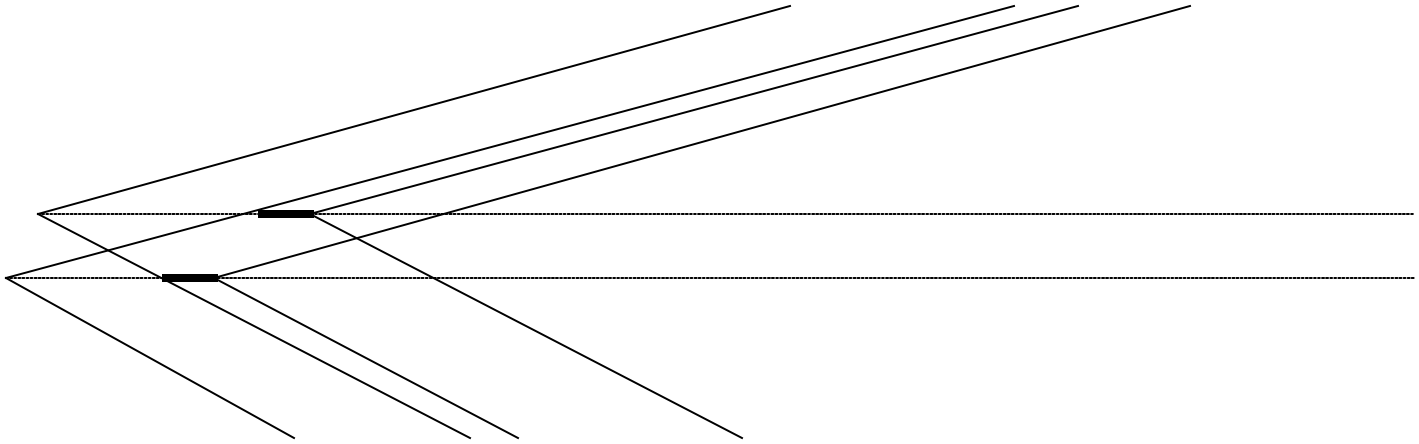
- Pas de synchronisation : l'émission n'attend pas le prochain slot libre (complètement décentralisé)
- Probabilité de succès pour un nœud arbitraire : $Np(1-p)^{2(N-1)}$,

  optimisé pour p = 1/(2e) la moitié du cas "slotted"

CSMA
"Carrier Sense Multiple Access"

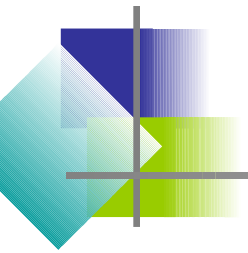
Règles de politesse : "écouter avant de parler, arrêter de parler si quelqu'un d'autre prend la parole"

- Les collisions sont dues au délai de propagation
- CSMA/CD



Temps de détection et d'arrêt

Protocoles tournants



- "polling" : existence d'un nœud maître qui donne la parole à tour de rôle -> délais de notification et vulnérabilité
- Passage d'un jeton de parole : vulnérabilité du jeton

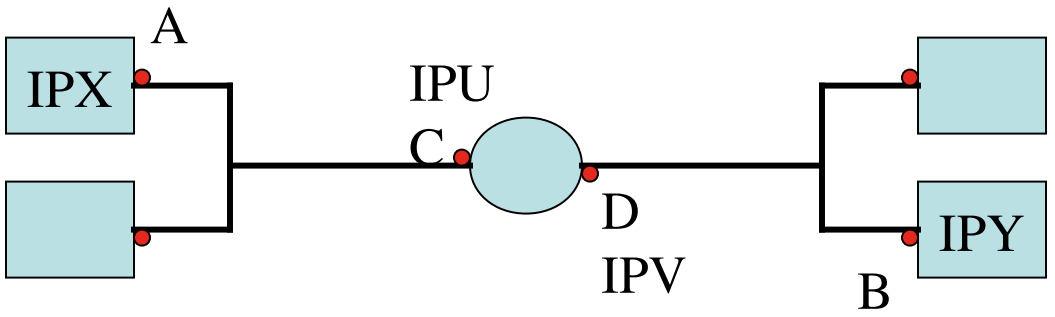
4.4 Adressage dans un réseau à diffusion

- Il s'agit de re-construire une liaison bi-point entre nœuds connectés à un canal en diffusion
- Un coupleur possède une adresse physique (typiquement 6 octets). C'est une adresse "en dur" (ROM) unique (assuré par le constructeur qui achète des bits d'adresse, par exemple 24 fixés par l'IEEE et 24 au gré de la compagnie)
- Par opposition à l'adresse IP qui est hiérarchique et peut changer lorsque la machine se déplace
- L'adresse physique du destinaire sur le même réseau local est passée dans le message et filtrée par le destinataire. A noter aussi l'adresse de diffusion : FF-FF-FF-FF-FF-FF

Protocole de résolution d'adresse (ARP)

- Le module ARP sur chaque nœud donne la correspondance adresse IP / adresse physique du coupleur
- Se trouve l'équivalent d'un DNS limité au réseau local
- Une entrée dans l'annuaire a une durée de vie limitée (typiquement 20 minutes)
- Lorsqu'une entrée n'est pas présente, le protocole de résolution est déclenché :
  - Diffusion d'une "requête ARP" sur le réseau avec les adresses IP et physique de l'émetteur et l'adresse IP du destinataire
  - Les récepteurs comparent leur adresse IP à celle du destinataire. Le nœud sélectionné renvoit à l'émetteur une "réponse ARP" avec son adresse physique. Ce qui permet à l'émetteur de mettre à jour son annuaire
- ARP est "plug-and-play" (s'adapte tout seul à un changement de l'environnement)

La traversée d'un routeur



- Déclenchement de l'ARP à chaque arrivée dans un nouveau réseau local

-> ARP Request (IPX,A,IPU,FF) /* IPU routage pour IPY */
<- ARP Reply (IPU,C,IPX,A)
-> Message IP (IPX,A,IPU,C,IPY)
-> ARP Request (IPV,D,IPY,FF) /* IPV routage pour IPY */
<- ARP Reply (IPY,B,IPV,D)
-> Message IP (IPV,D,IPY,B)

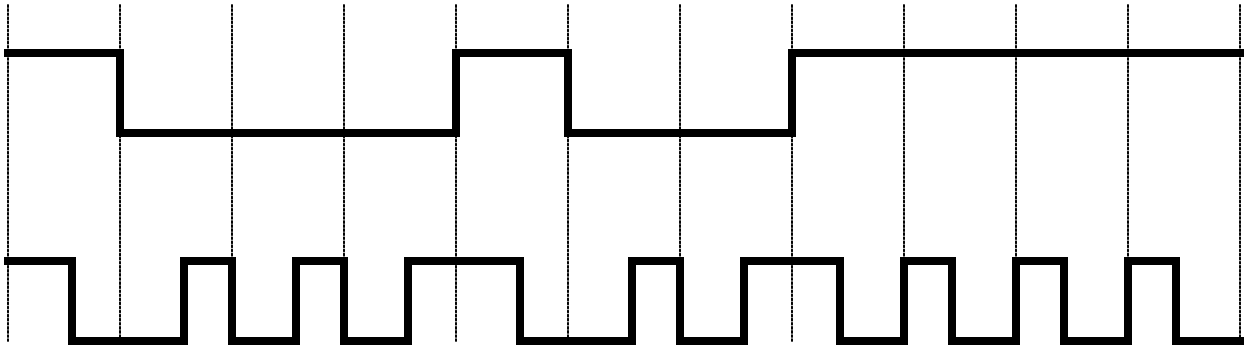
4.5 Ethernet

- Les datagrammes IP sont encapsulés dans les trames Ethernet
- Format de la trame :

  Préambule / adresse dest / adresse source / type / données / CRC
  - Données : de 46 à 1500 octets : au delà, il faut fragmenter; en deçà il faut bourrer
  - Adresses : 6 octets. Le récepteur jette les trames qui ne lui sont pas destinées
  - Type : 2 octets. Utilisé pour multiplexer les protocoles de niveau réseau (IP/NovellIPX/AppleTalk)
  - CRC : 4 octets. Les trames erronées sont simplement jetées
  - Préambule : 8 octets = 10101010101010101 … 10101011 : sert à re-synchroniser les horloges des coupleurs (dérive). Le drapeau 11 final sert à avertir que les bits suivants seront une adresse physique. La fin de la trame est simplement détectée par une absence de courant

Transmission en “bande de base”

- Transmission directe sur le canal sans changement de fréquence (contrairement à l’ADSL)
- Codage Manchester pour garder la synchronisation des horloges : chaque bit contient une transition en son milieu (1 -> 10; 0 -> 01)



Ethernet CSMA/CD

- Attendre l'absence de courant + 96 bits pour transmettre
- Si détecte la présence d'un autre transmetteur, il arrête sa transmission et insère un signal d'encombrement de 48 bits. Puis il attend avant de réessayer une transmission : tire une valeur aléatoire K entre 0 et $2^{min(n,10)}-1$ pour la $n^{ième}$ collision et attend K x 512 bits (attente exponentielle)
- Le signal d'encombrement est pour faire en sorte qu'un émetteur émette suffisamment de bits avant de s'arrêter pour cause de collision : ce qui risque de ne pas montrer suffisamment d'énergie pour la détection de collision chez les autres
- La norme Ethernet donne une limite de distance entre nœuds pour que l'algorithme d'attente exponentielle fonctionne
- Efficacité dans l'utilisation de la bande passante : 1/(1 + 5P/T) avec P : temps de propagation, T temps de transmission

Technologies

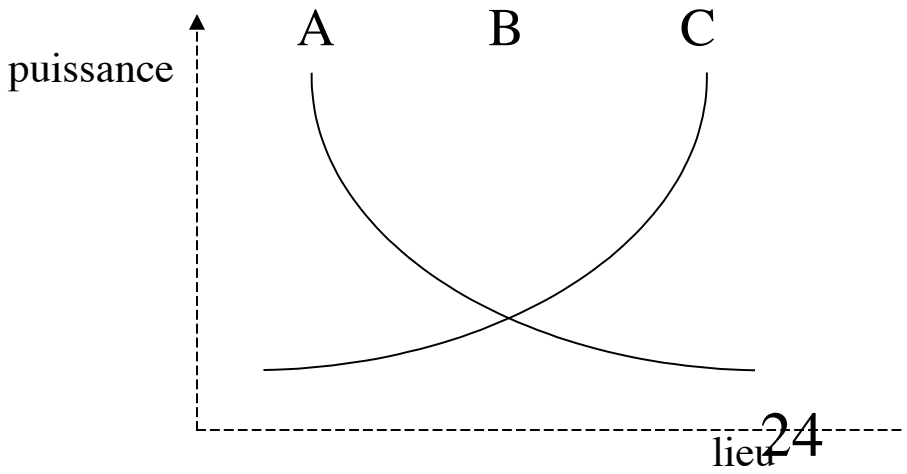
- 10Base2 (10Mbps, 200m). Cable coaxial fin linéaire sur lequel sont fixées les machines (30 max car les connexions provoquent aussi des atténuations du signal). Les cables peuvent être enchaînés à l'aide de "répéteurs" (5 segments max)
- 10BaseT et 100BaseT : "fast Ethernet". Topologie en étoile (concentrateur ou "hub"). Le concentrateur assure la diffusion en recopiant le bit en entrée sur toutes les sorties. 100BaseT utilise des paires torsadées. Le codage Manchester est remplacé par le codage 4B5B (4 bits sur 5 périodes)
- Gigabit Ethernet et 10Gbps Ethernet : même principe que BaseT. Existe aussi la notion de switch pour les communications point-a-point

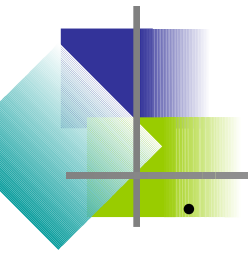
4.6 Sans fil

- IEEE 802.11 : les éléments de base d'une architecture réseau sont les cellules ("Basic Service Set"). Une cellule contient plusieurs stations sans fil et une station de base
- Une cellule peut aussi se former spontanément par proximité d'un ensemble de stations. Il n'y a pas alors de station de base spécifique -> réseaux ad-hoc
- "Media Access Control" : protocole MAC fondé sur la technique CSMA/CA (CA pour "Control Avoidance"). Le canal partagé est un canal radio (fréquence). L'écoute consiste à mesurer l'énergie sur la fréquence choisie. Si le canal est libre pendant un temps DIFS ("Distributed Inter Frame Space"), la station est autorisée à émettre. Après réception complète, le récepteur attend un temps SIFS ("Short Inter Frame Spacing") et renvoit un accusé -> il n'y a pas de détection de collision pendant la transmission comme dans Ethernet

Problème de la station cachée

- Une obstruction physique (montagne) entre les stations A et C peut faire qu'elles soient en interférence pour une communication avec B, mais sans s'en apercevoir : la collision n'est pas détectable
- Une autre raison peut être l'atténuation ("fading"). L'interférence en B n'est pas détectée par A ni C





- Les trames contiennent un champ de durée de transmission pour que les autres récepteurs puissent calculer un temps minimum d'attente
- Il y a aussi des messages courts d'allocation du canal ("Request To Send" RTS et "Clear To Send" CTS) : RTS contient la durée de transmission données+ack+SIFS. CTS est la réponse d'autorisation. Les autres entendent le RTS et CTS et s'abstiennent donc d'émettre
- Autres facilités : synchronisation des horloges, gestion de la puissance, gestion de l'insertion et du retrait des stations…

Bluetooth

- 2.45 Ghz dans la bande radio sans licence
- 721 kbps ou 3 canaux voie à 64 kbps
- 10 à 100m de portée
- Toute une pile de protocoles en fait :
  - Protocole bande de base -> notion de réseau Bluetooth "piconet"
  - Protocole de gestion des liens (orienté connexion)
  - Protocole L2CAP : adaptation pour les niveaux supérieurs

Cours 4 : le réseau local (lien)

- 4.1 Service de lien
- 4.2 Détection et correction d'erreurs
- 4.3 Protocoles à accès multiple
- 4.4 Adressage dans un réseau à diffusion
- 4.5 Ethernet
- 4.6 Sans fil

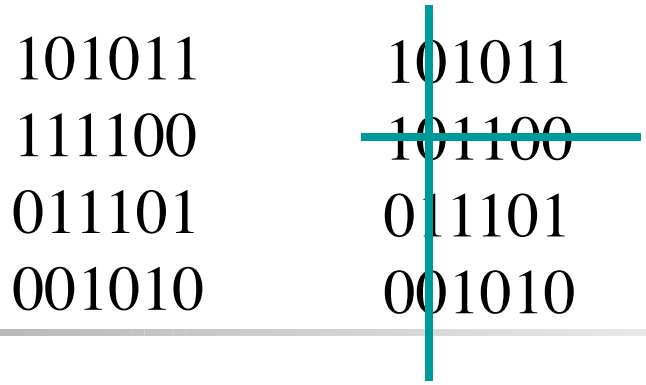
4.1 Service de lien

- Entre deux nœuds du réseau : les trames (paquets/datagrammes/trames)
- Mis en œuvre dans un coupleur, généralement matériel, avec ses propres processeurs (ex. Carte PMCIA)
- Offre des services comparables au transport de bout en bout (contrôle des erreurs, du flux, …)

4.2 Détection et correction d'erreurs

- Au niveau bit : changement de valeur possible sur le canal physique (pas de perte car synchro à ce niveau là)
- EDC ("Error Detection and Correction") bits
- (D + EDC) -> (D' + EDC') : la probabilité d'erreurs non détectées n'est pas nulle : dépend de la technique utilisée et de l'overhead admissible
- Parité / Checksum / Codes cycliques

Parité



- EDC = 1 bit
- Nb 1 dans D+EDC pair
- Si le récepteur en compte un nombre impair, il sait qu'il y a eu au moins 1 bit erroné
- Si le nombre d'erreurs est pair, aucune erreur n'est détectée : la question est la probabilité de plusieurs erreurs dans une même trame -> problème des phénomènes de rafales
- Généralisation à 2 dimensions -> parité ligne et parité colonne -> permet de corriger 1 erreur bit (le bit qui provoque une erreur de parité à la fois sur la ligne et la colonne) et de détecter 2 erreurs bit

Checksum

- Internet (UDP/TCP) : les données sont traitées comme une suite d'entiers sur 16 bits et sont additionnés. Le complément à 1 de la somme forme le checksum transmis dans la trame
- Le récepteur calcule la somme des entiers reçus. Si un 0 apparaît, une erreur est détectée

0110011001100110	0110011001100110
0101010101010101	0101010101010101
1011101110111011	0000111100001111
0000111100001111	0011010100110101
1100101011001010	1111111111111111

"Cyclic Redondancy Check" (CRC)

- Les suites de bits sont vues comme les coefficients d'un polynôme. Les données sont transmises avec un code dépendant d'un générateur G de r+1 bits, R = reste de ($D \times 2^r / G$)
- On transmet ($D \times 2^r$) xor R
- Le récepteur divise la donnée reçue par G. Si le reste est différent de 0, une erreur est détectée
- Détecte des rafales d'erreurs de moins de r+1 bits. Au delà la probabilité de détection est $1-0.5^r$

- IEEE G =

  10000010011000001000111011011011 1

D = 101110 G = 1001

```
101110000 | 1001
  1010    | 101011
    1100
     1010
      011
```

4.3 Protocoles à accès multiple

- Canal de diffusion partagé : cable Ethernet, radio sans-fil (Wifi), satellite, …
- Collisions : les signaux des trames se mélangent, rendant les trames inaudibles pour les différents récepteurs -> toutes les trames en collision sont perdues…
- Coordination nécessaire entre les émetteurs -> rôle du protocole d'accès :
  - Protocoles de partitionnement du canal,
  - Protocoles à accès aléatoire,
  - Protocoles tournants

Protocoles à partitionnement

- Trois méthodes :
  - Division temporelle ("Time Division Multiplexing" TDM) : le train de bits est divisé en parts ("slots") allouées statiquement pour chaque récepteur : 123412341234. L'émetteur attend son slot. Typiquement la taille du slot permet de loger un paquet

    -> sans collision et équitable, mais débit R/N même si un seul émetteur + attente
  - Division fréquentielle ("Frequency Division Multiplexing" FDM) : la bande de fréquence est divisée en N -> fournit N sous-canaux de débit R/N : mêmes avantages et inconvénients que la technique TDM

- Division par code
("Code Division Multiple Acces" CDMA)
  - Chaque bit émis est codé avec un mot c de longueur M
  - En représentant le bit 0 par -1, le bit de donnée d_i est transformé en la suite de bits $Z_{i,m}$ pour m de 1 à M : $Z_{i,m} = d_i \times c_m$
  - Le récepteur retrouve le bit de donnée par l'opération :

    $d_i = 1/M \sum_{m=1,M} Z_{i,m} \times c_m$
  - Exemple : C = 111-11-1-1-1

    -1 1 -> -1-1-11-1111 111-11-1-1-1 -> -1 1

Traitement des interférences : N émetteurs

- Les valeurs de chaque émetteur s'ajoutent sur un mini-slot : $Z_{i,m}^{*} = \sum_{s=1,N} Z_{i,m}^{s}$
- Astucieusement, on peut démêler par la même formule que précédemment les valeurs à la réception si les mots de code de chaque émetteur sont bien choisis :

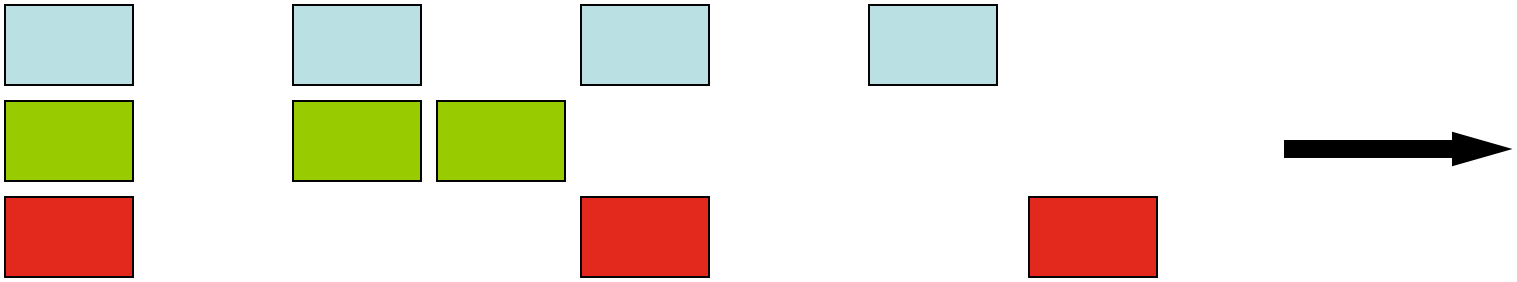
  $d_i = 1/M \sum_{m=1,M} Z_{i,m}^{*} \times c_m$
- Outre le choix des codes, il y a aussi des difficultés pratiques dans la régulation de puissance électrique sur le canal
- Exemple : c1 = 111-11-1-1-1, c2 = 1-1111-111

  -1 1 // 1 1 -> 0-2020022 20202-200 -> -1 1 (récepteur 1)

Accès aléatoire

- Transmission au débit max, mais retransmission en cas de collision avec une attente aléatoire (indépendante pour chaque émetteur) -> des centaines de protocoles (Aloha/CSMA/Ethernet …)
- Exemple le plus simple : "Slotted ALOHA"
  - Trames de L bits, débit du canal R
  - Le temps est divisé en slots de taille L/R secondes
  - Les nœuds ne transmettent qu'en début de trame. Ils sont synchronisés et connaissent donc tous le même début de trame
  - Tous les nœuds détectent une collision avant la fin du slot
  - En cas de collision, la trame est retransmise dans des slots prochains avec une probabilité p

Slotted ALOHA



- Performances : à charge max, N émetteurs
- Probabilité de succès pour un nœud arbitraire : $Np(1-p)^{N-1}$
- Choix du meilleur p pour N arbitrairement grand :
  p = 1/e (0.37)
- 37% seulement du débit canal est utilisé !

Pure ALOHA

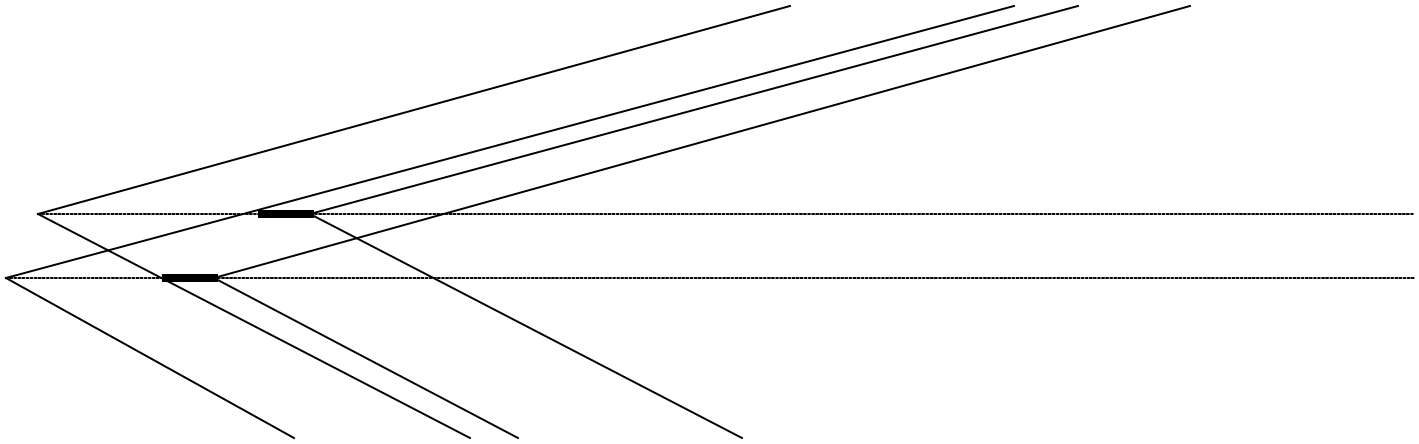
- Pas de synchronisation : l'émission n'attend pas le prochain slot libre (complètement décentralisé)
- Probabilité de succès pour un nœud arbitraire : $Np(1-p)^{2(N-1)}$,

  optimisé pour p = 1/(2e) la moitié du cas "slotted"

CSMA
"Carrier Sense Multiple Access"

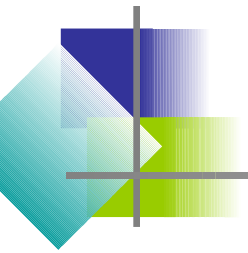
Règles de politesse : "écouter avant de parler, arrêter de parler si quelqu'un d'autre prend la parole"

- Les collisions sont dues au délai de propagation
- CSMA/CD



Temps de détection et d'arrêt

Protocoles tournants



- "polling" : existence d'un nœud maître qui donne la parole à tour de rôle -> délais de notification et vulnérabilité
- Passage d'un jeton de parole : vulnérabilité du jeton

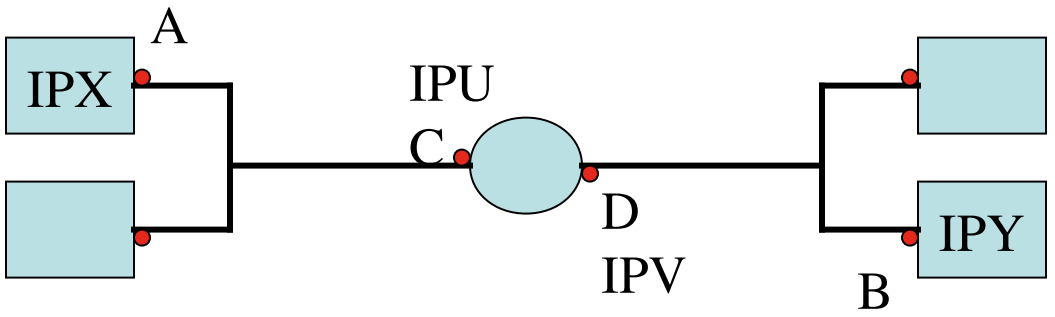
4.4 Adressage dans un réseau à diffusion

- Il s'agit de re-construire une liaison bi-point entre nœuds connectés à un canal en diffusion
- Un coupleur possède une adresse physique (typiquement 6 octets). C'est une adresse "en dur" (ROM) unique (assuré par le constructeur qui achète des bits d'adresse, par exemple 24 fixés par l'IEEE et 24 au gré de la compagnie)
- Par opposition à l'adresse IP qui est hiérarchique et peut changer lorsque la machine se déplace
- L'adresse physique du destinaire sur le même réseau local est passée dans le message et filtrée par le destinataire. A noter aussi l'adresse de diffusion : FF-FF-FF-FF-FF-FF

Protocole de résolution d'adresse (ARP)

- Le module ARP sur chaque nœud donne la correspondance adresse IP / adresse physique du coupleur
- Se trouve l'équivalent d'un DNS limité au réseau local
- Une entrée dans l'annuaire a une durée de vie limitée (typiquement 20 minutes)
- Lorsqu'une entrée n'est pas présente, le protocole de résolution est déclenché :
  - Diffusion d'une "requête ARP" sur le réseau avec les adresses IP et physique de l'émetteur et l'adresse IP du destinataire
  - Les récepteurs comparent leur adresse IP à celle du destinataire. Le nœud sélectionné renvoit à l'émetteur une "réponse ARP" avec son adresse physique. Ce qui permet à l'émetteur de mettre à jour son annuaire
- ARP est "plug-and-play" (s'adapte tout seul à un changement de l'environnement)

La traversée d'un routeur



- Déclenchement de l'ARP à chaque arrivée dans un nouveau réseau local

-> ARP Request (IPX,A,IPU,FF) /* IPU routage pour IPY */
<- ARP Reply (IPU,C,IPX,A)
-> Message IP (IPX,A,IPU,C,IPY)
-> ARP Request (IPV,D,IPY,FF) /* IPV routage pour IPY */
<- ARP Reply (IPY,B,IPV,D)
-> Message IP (IPV,D,IPY,B)

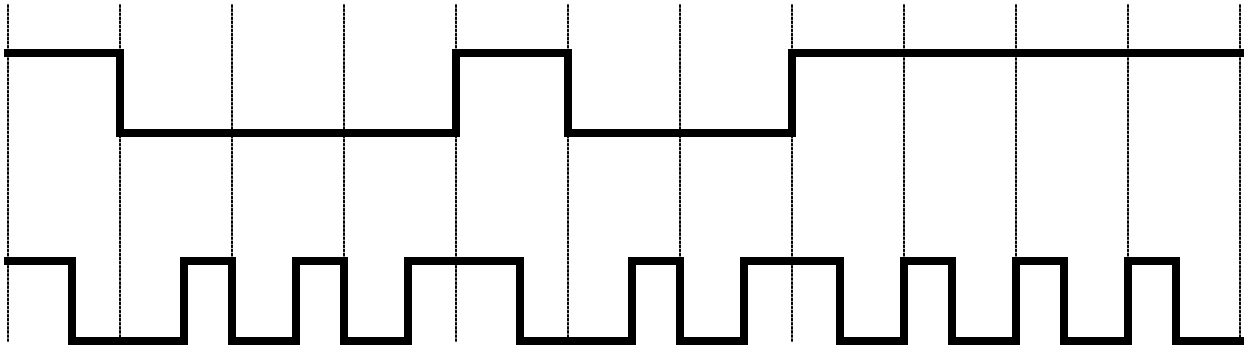
4.5 Ethernet

- Les datagrammes IP sont encapsulés dans les trames Ethernet
- Format de la trame :

  Préambule / adresse dest / adresse source / type / données / CRC
  - Données : de 46 à 1500 octets : au delà, il faut fragmenter; en deçà il faut bourrer
  - Adresses : 6 octets. Le récepteur jette les trames qui ne lui sont pas destinées
  - Type : 2 octets. Utilisé pour multiplexer les protocoles de niveau réseau (IP/NovellIPX/AppleTalk)
  - CRC : 4 octets. Les trames erronées sont simplement jetées
  - Préambule : 8 octets = 10101010101010101 … 10101011 : sert à re-synchroniser les horloges des coupleurs (dérive). Le drapeau 11 final sert à avertir que les bits suivants seront une adresse physique. La fin de la trame est simplement détectée par une absence de courant

Transmission en “bande de base”

- Transmission directe sur le canal sans changement de fréquence (contrairement à l’ADSL)
- Codage Manchester pour garder la synchronisation des horloges : chaque bit contient une transition en son milieu (1 -> 10; 0 -> 01)



Ethernet CSMA/CD

- Attendre l'absence de courant + 96 bits pour transmettre
- Si détecte la présence d'un autre transmetteur, il arrête sa transmission et insère un signal d'encombrement de 48 bits. Puis il attend avant de réessayer une transmission : tire une valeur aléatoire K entre 0 et $2^{min(n,10)}-1$ pour la $n^{ième}$ collision et attend K x 512 bits (attente exponentielle)
- Le signal d'encombrement est pour faire en sorte qu'un émetteur émette suffisamment de bits avant de s'arrêter pour cause de collision : ce qui risque de ne pas montrer suffisamment d'énergie pour la détection de collision chez les autres
- La norme Ethernet donne une limite de distance entre nœuds pour que l'algorithme d'attente exponentielle fonctionne
- Efficacité dans l'utilisation de la bande passante : 1/(1 + 5P/T) avec P : temps de propagation, T temps de transmission

Technologies

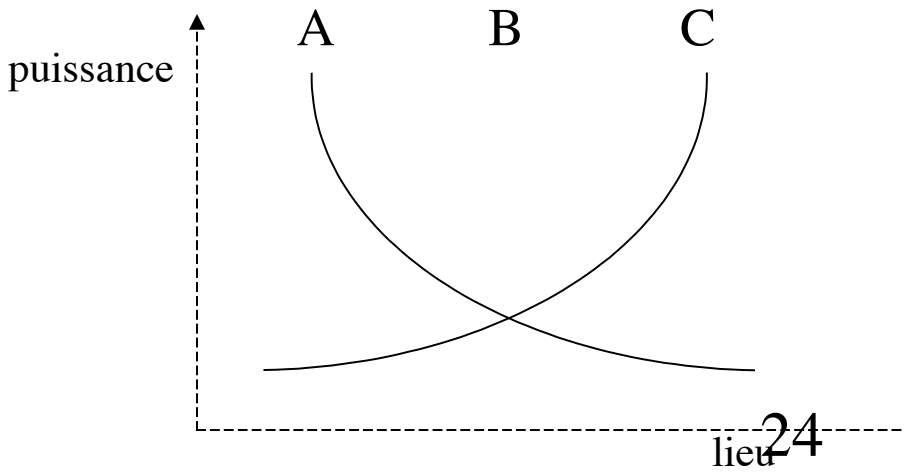
- 10Base2 (10Mbps, 200m). Cable coaxial fin linéaire sur lequel sont fixées les machines (30 max car les connexions provoquent aussi des atténuations du signal). Les cables peuvent être enchaînés à l'aide de "répéteurs" (5 segments max)
- 10BaseT et 100BaseT : "fast Ethernet". Topologie en étoile (concentrateur ou "hub"). Le concentrateur assure la diffusion en recopiant le bit en entrée sur toutes les sorties. 100BaseT utilise des paires torsadées. Le codage Manchester est remplacé par le codage 4B5B (4 bits sur 5 périodes)
- Gigabit Ethernet et 10Gbps Ethernet : même principe que BaseT. Existe aussi la notion de switch pour les communications point-a-point

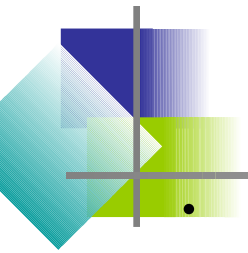
4.6 Sans fil

- IEEE 802.11 : les éléments de base d'une architecture réseau sont les cellules ("Basic Service Set"). Une cellule contient plusieurs stations sans fil et une station de base
- Une cellule peut aussi se former spontanément par proximité d'un ensemble de stations. Il n'y a pas alors de station de base spécifique -> réseaux ad-hoc
- "Media Access Control" : protocole MAC fondé sur la technique CSMA/CA (CA pour "Control Avoidance"). Le canal partagé est un canal radio (fréquence). L'écoute consiste à mesurer l'énergie sur la fréquence choisie. Si le canal est libre pendant un temps DIFS ("Distributed Inter Frame Space"), la station est autorisée à émettre. Après réception complète, le récepteur attend un temps SIFS ("Short Inter Frame Spacing") et renvoit un accusé -> il n'y a pas de détection de collision pendant la transmission comme dans Ethernet

Problème de la station cachée

- Une obstruction physique (montagne) entre les stations A et C peut faire qu'elles soient en interférence pour une communication avec B, mais sans s'en apercevoir : la collision n'est pas détectable
- Une autre raison peut être l'atténuation ("fading"). L'interférence en B n'est pas détectée par A ni C





- Les trames contiennent un champ de durée de transmission pour que les autres récepteurs puissent calculer un temps minimum d'attente
- Il y a aussi des messages courts d'allocation du canal ("Request To Send" RTS et "Clear To Send" CTS) : RTS contient la durée de transmission données+ack+SIFS. CTS est la réponse d'autorisation. Les autres entendent le RTS et CTS et s'abstiennent donc d'émettre
- Autres facilités : synchronisation des horloges, gestion de la puissance, gestion de l'insertion et du retrait des stations…

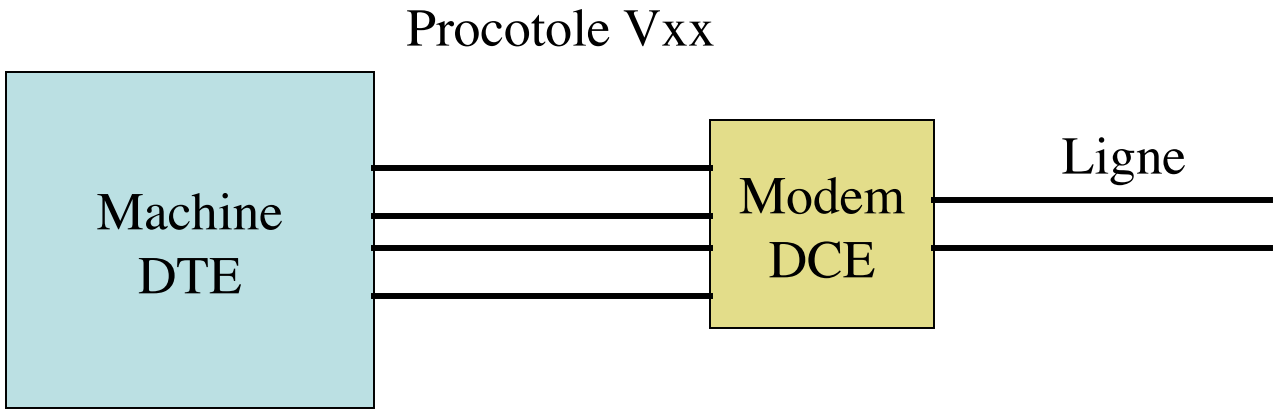
Bluetooth

- 2.45 Ghz dans la bande radio sans licence
- 721 kbps ou 3 canaux voie à 64 kbps
- 10 à 100m de portée
- Toute une pile de protocoles en fait :
  - Protocole bande de base -> notion de réseau Bluetooth "piconet"
  - Protocole de gestion des liens (orienté connexion)
  - Protocole L2CAP : adaptation pour les niveaux supérieurs

Cours 5 : quelques éléments de transmission numérique

- 5.1 Modulation analogique
- 5.2 Compression
- 5.3 Modulation par implusions codées (PCM)
- 5.4 Le cas de l'ADSL

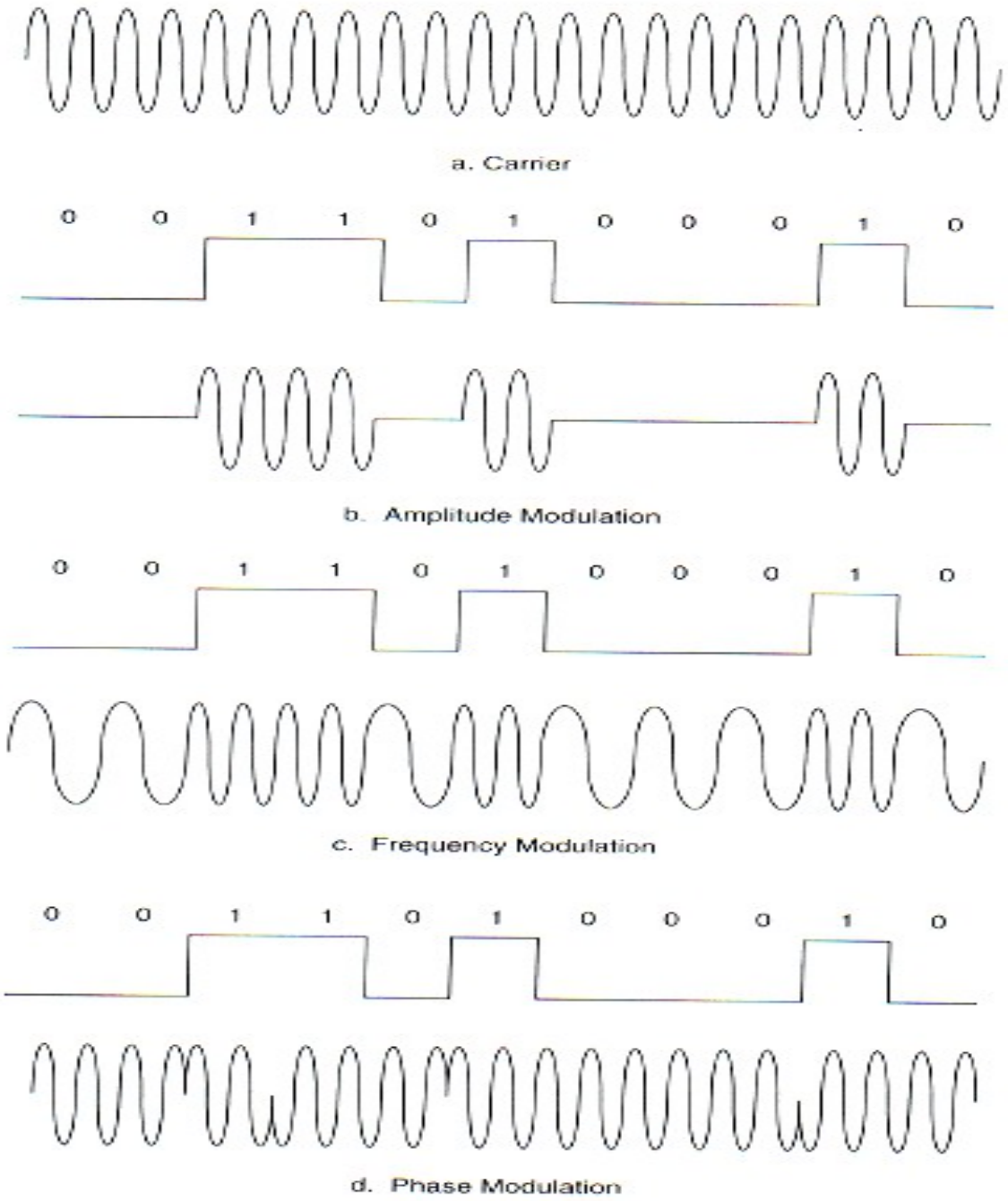
Modem : modulateur-démodulateur



- Modem 300 bauds (ref E. Baudot) 1980 -> ADSL 20Mbps/8Mbps/1Mbps

5.1 Modulation analogique

- Difficulté de transmettre du courant continu sur de longues distances à travers des supports différents
- Puissance demandée importante pour transmettre en basse fréquence
- -> modulation d'une porteuse (courant oscillant) s(t) = A cos (2 ft +) par une information portée sur l'amplitude (A), la fréquence (f) ou la phase ( )


a. Carrier
0 0 1 1 0 1 0 0 0 1 0
b. Amplitude Modulation
0 0 1 1 0 1 0 0 0 1 0
c. Frequency Modulation
0 0 1 1 0 1 0 0 0 1 0
d. Phase Modulation

BPSK
(Binary Phase Shift Keying)

- Train binaire m(t) = 1 bit toutes les B périodes
- Modulation :

  BPSK(t)=Asin(2 ft+{O si m(t)=1, sinon})

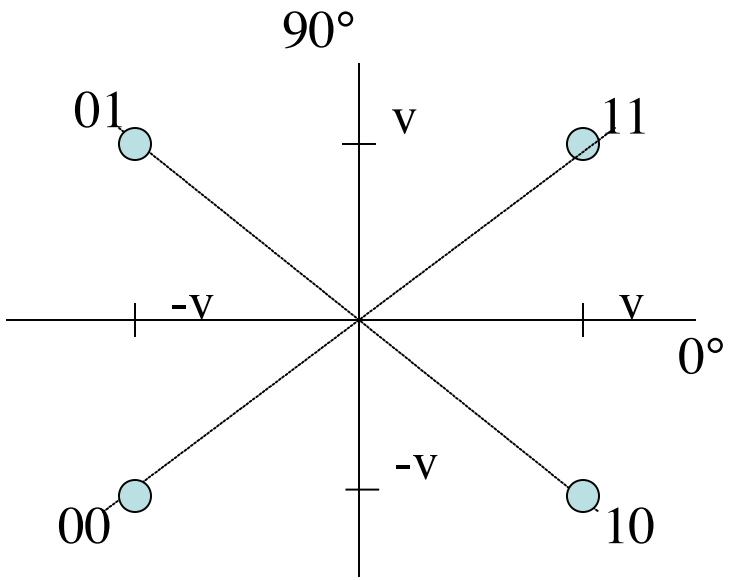
  = Asin(2 ft+ (1-m(t)))

  = (2m(t)-1)Asin(2 ft)
- Démodulation :

  m(t)=1/2(1+signe(BPSK(t)sin(2 ft)))

Codage symbole

- On groupe les bits pour former un nouveau vocabulaire : ex. Dibits 10 11 00 10
- Utilisation conjointe amplitude-phase : ex QAM ("Quadrature Amplitude Modulation")



QPSK (Quadratic Phase Shift Keying)

- Train binaire m(t) = 1 dibit toutes les B périodes (on double la fréquence)
- n(t) = 0 si 11, 1 si 01, 2 si 00, 3 si 10
- Modulation :

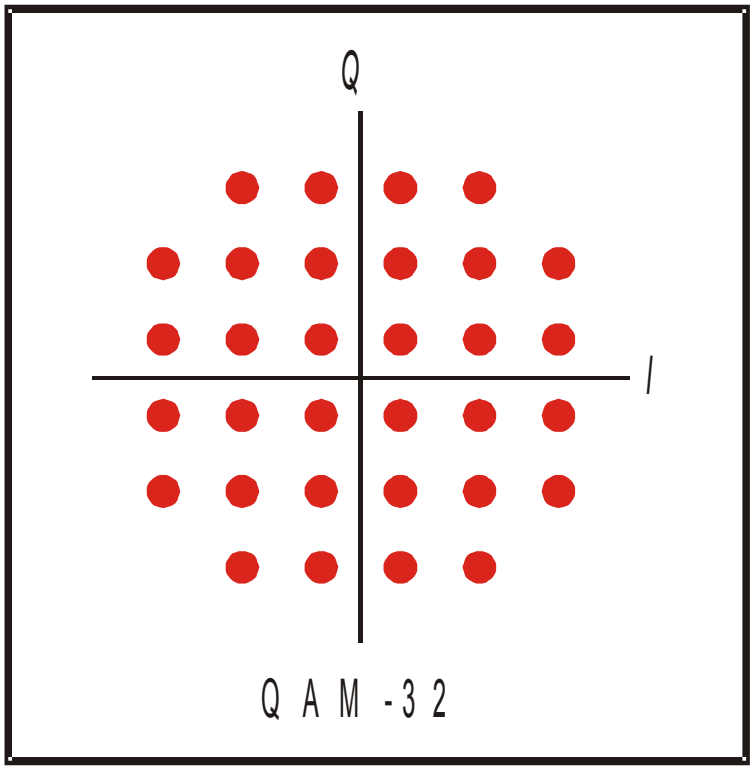
  QPSK(t)=A√2/2[sin(2 ft+ /2(1/2+n(t)) +
  cos(2 ft+ /2(1/2+n(t))]

- Démodulation :

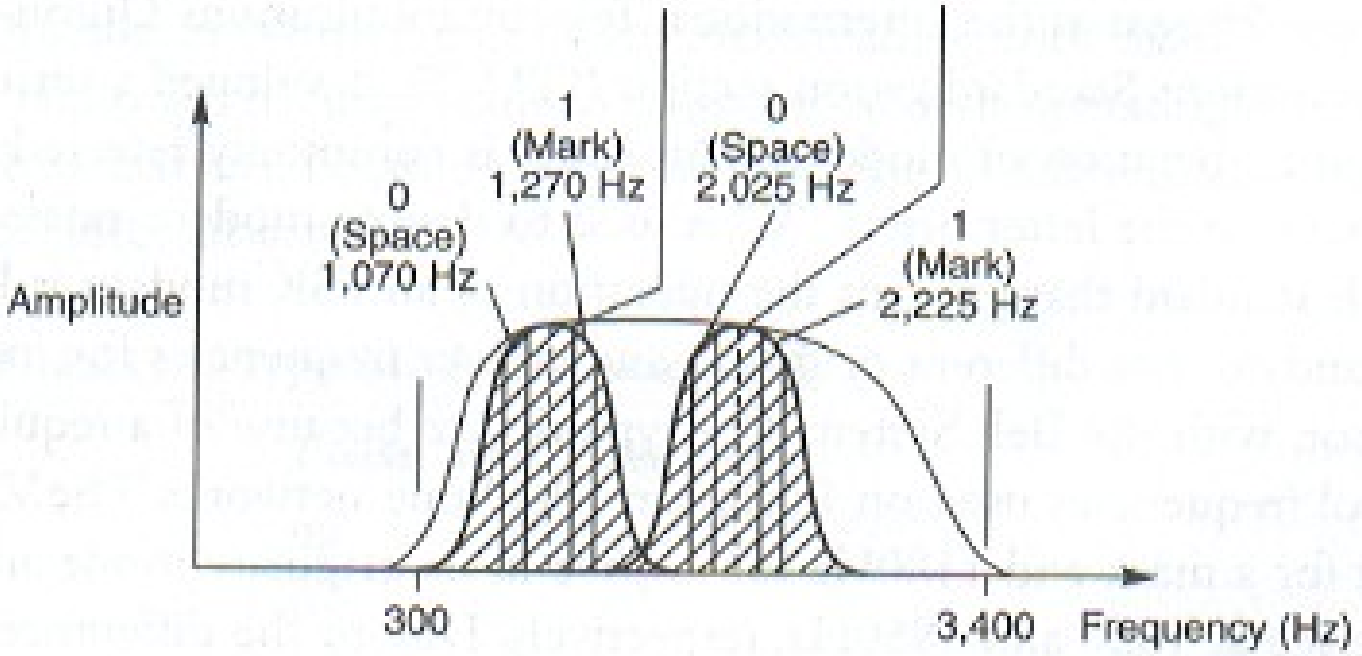
  [signe(BPSK(t)sin(2 ft))), signe(BPSK(t)sin(2 ft)))]
  permet de trouver le bon quadrant

Combinaison de modulations

- Combiner
  - Modulation de phase
  - Modulation d'amplitude
- Exemple: QAM-32
  - Symboles: 5 bits
  - 32 points dans la constellation
- Effet de flou de la transmission bruitée



Full-duplex : utilisation de plusieurs porteuses



300 - 3400 Hz : la bande vocale du téléphone…

5.2 Compression

- Classique : "Run Length Encoding" RLE
  - Si un élément (16 bits ici) est répété 3 fois ou plus, on écrit le nombre d'itérations suivi de sa valeur
  - Sinon la suite n'est pas codée. On écrit 00, puis le nombre de valeurs, puis ces valeurs
  - Exemple : 07 07 07 07 07 FB FB 89 23 ->
    05 07 00 04 FB FB 89 23
- A dictionnaire : A. Lempel, J. Ziv & T. Welch : algorithme LZW
- Couplage compression/détection-correction erreurs -> les modems haut-débit intègrent une détection-correction

Compression LZW

- Un dictionnaire des suites de lettres est construit dynamiquement pendant l'émission. Il est reconstruit de la même façon à la réception (il n'y a donc pas transmission du dictionnaire)
- Lorsqu'une chaîne est présente dans le dictionnaire, seul sa position est transmise. A la réception, la connaissance des positions précédentes et de la méthode de construction du dictionnaire permet de reconstruire celui-ci et donc de décoder

Code du compresseur

```
STRING = get input character
WHILE there are still input characters DO
    CHARACTER = get input character
    IF STRING+CHARACTER is in the string table then
            STRING = STRING+character
    ELSE
            output the code for STRING
            add STRING+CHARACTER to the string table
            STRING = CHARACTER
    END of IF
END of WHILE
output the code for STRING
```

string	char/dico	sortie
A	B / AB	1
B	A / BA	2
A	B	
AB	A / ABA	3
A	B	
AB	A	
ABA	-	5

Dico

1	2	3	4	5
A	B	AB	BA	ABA

Code du décompresseur

```
Read OLD_CODE
output translation of OLD_CODE
CHARACTER = OLD_CODE
WHILE there are still input characters DO
    Read NEW_CODE
    IF NEW_CODE is not in the translation table THEN
            STRING = get translation of OLD_CODE
            STRING = STRING+CHARACTER
    ELSE
            STRING = get translation of NEW_CODE
    END of IF
    output STRING
    CHARACTER = first character in STRING
    add OLD_CODE + CHARACTER to the translation table
    OLD_CODE = NEW_CODE
END of WHILE
```

Transmission synchrone/asynchrone

- Asynchrone : insertion de bits "spéciaux" en début et fin de caractère
- Synchrone :
  - Ligne supplémentaire pour transmettre l'horloge
  - Utilisation de la suite de bits elle-même comme horloge -> codage Manchester par exemple

Modulation par impulsion codée (PCM, MIC)

- Son numérique -> convertisseurs analogique/numérique
- Echantillonnage
- Quantification
- Codage
- Exemples :
  - Téléphone : bande = 4kHz, fe = 8kHz, code 8 bits -> débit 64kbs
  - Hifi : bande = 20kHz, fe = 44.1 kHz, code 16 bits -> débit 176kbs

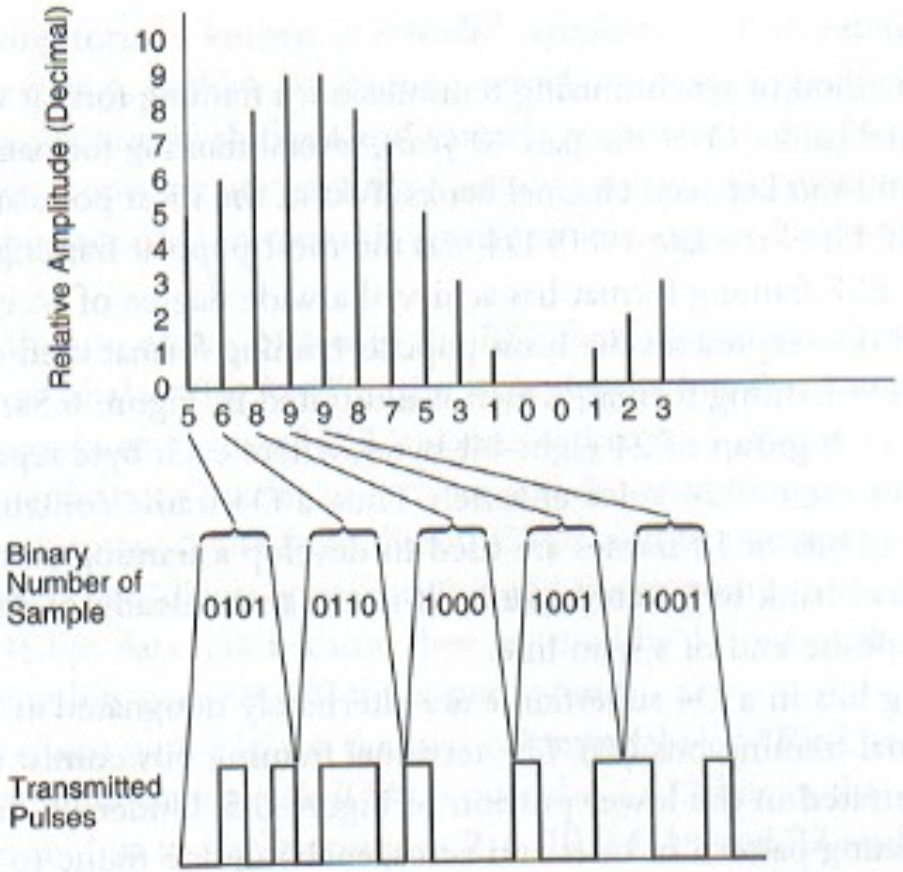
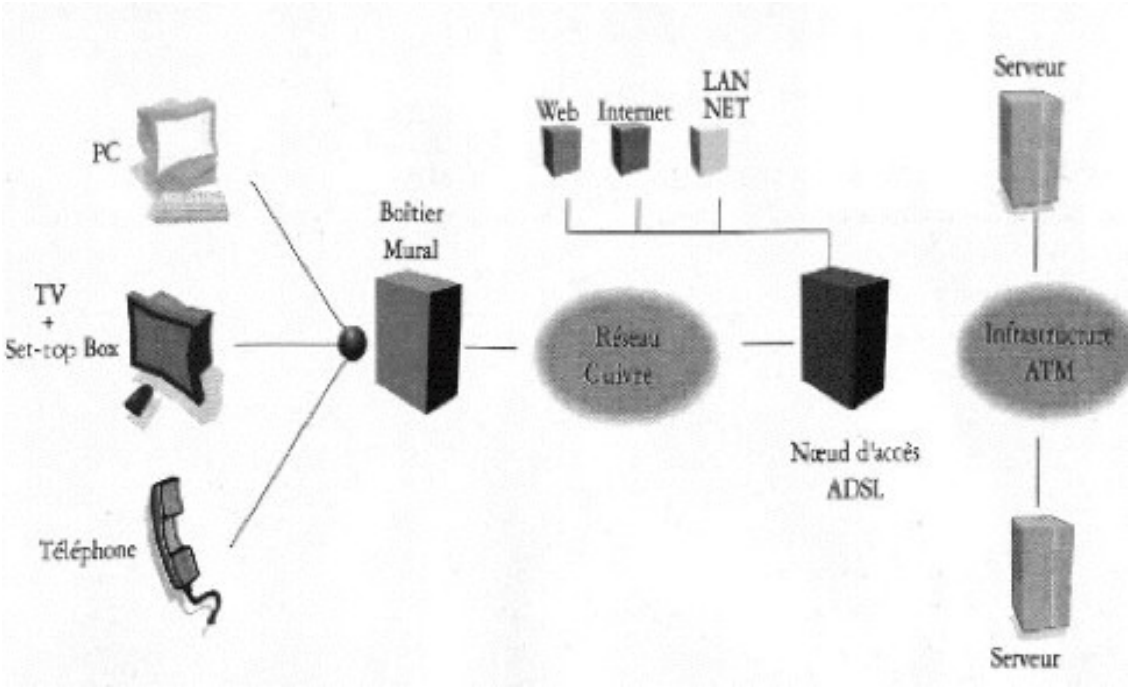
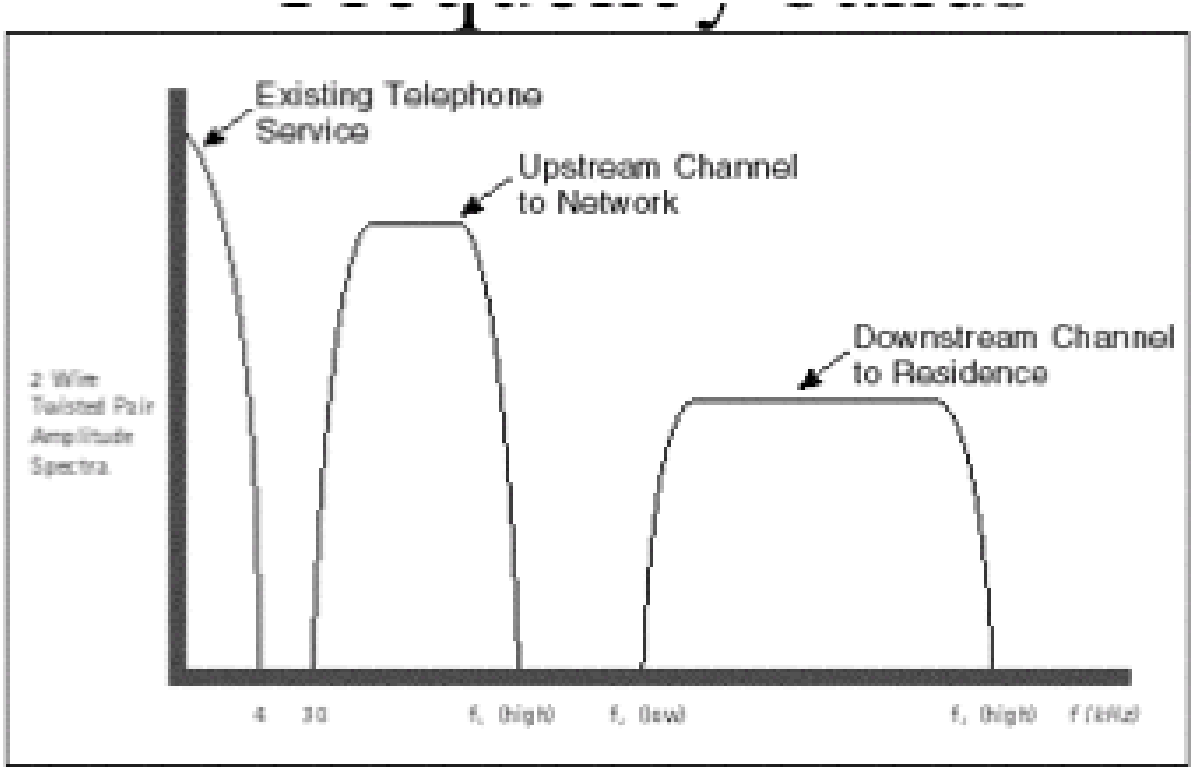


Figure 6.3 Pulse code modulation.

5.3 ADSL

- "Asymmetric Digital Subscriber Link"
- Paire torsadée
- Bande asymétrique :
  8Mbs entrant/
  1Mbps sortant
- Distance courte au commutateur





- FDM
- 0 - 4kHz : voix
- 94 - 106kHz : voie sortante
- 120 - 420kHz : voie entrante

- Modulation QAM
- Utilisation du DMT "Discrete Multi-Tone" : séparation en plusieurs sous-canaux pour partitionner les symboles

Cours 6 : la sécurité dans les réseaux

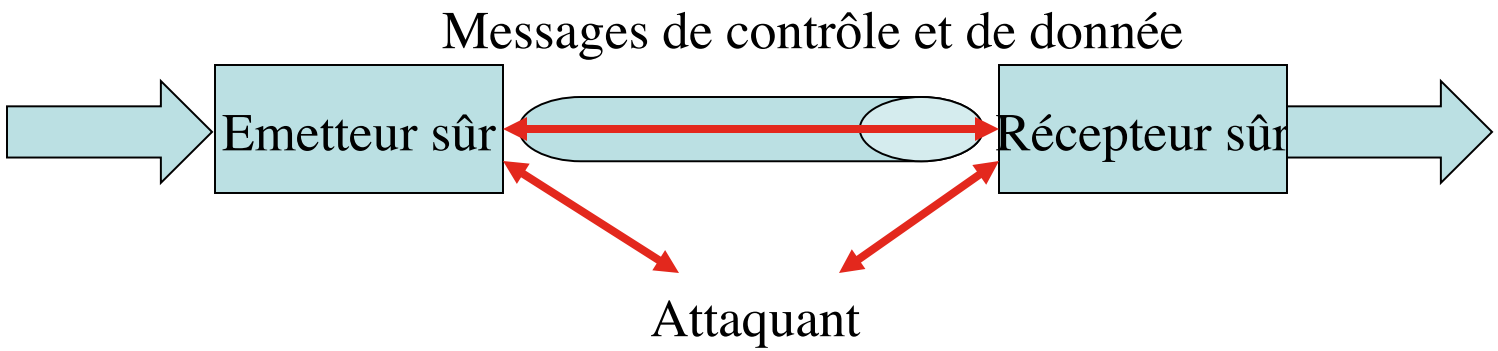
- 6.1 Propriétés de sécurité
- 6.2 Cryptage des données (DES)
- 6.3 Cryptage à clé publique
- 6.4 Authentification
- 6.5 Intégrité
- 6.6 Controle d'accès (pare-feux)
- 6.7 Attaques et contre-mesures
- 6.8 Applications sécurisées
- 6.9 Un peu de virologie

Une urbanisation numérique fragile (Internet/sans fil, "ubiquité")

- Les enjeux :
  - La maîtrise du cycle de vie des patrimoines numériques (transport, traitement, stockage) : question de souveraineté -> technique maîtresse du cryptage
  - La valorisation des contenus : enjeu économique -> tatouage, …
  - La confiance dans l'univers (république) numérique : enjeu social -> authentification, certificats, tiers de confiance, …
  - La sécurisation des infosphères :
    - Niveau individuel : préservation de l'intimité (filature électronique)
    - Niveau organisation : prévention des attaques, architectures de sécurité
    - Niveau état : invulnérabilité des infrastructures critiques, prévention des catastrophes

6.1 Propriétés de sécurité (les objectifs)

- Confidentialité : entre un émetteur et un récepteur. Nécessite l'encryptage de la donnée par l'émetteur et son décryptage par le récepteur
- Authentification : l'émetteur et le récepteur doivent s'assurer de l'identité du partenaire
- Intégrité et non répudiation : le message transmis n'est pas altéré
- Disponibilité et contrôle d'accès : l'attaquant ne peut empêcher l'accès aux ressources



Statistiques sur la sécurité

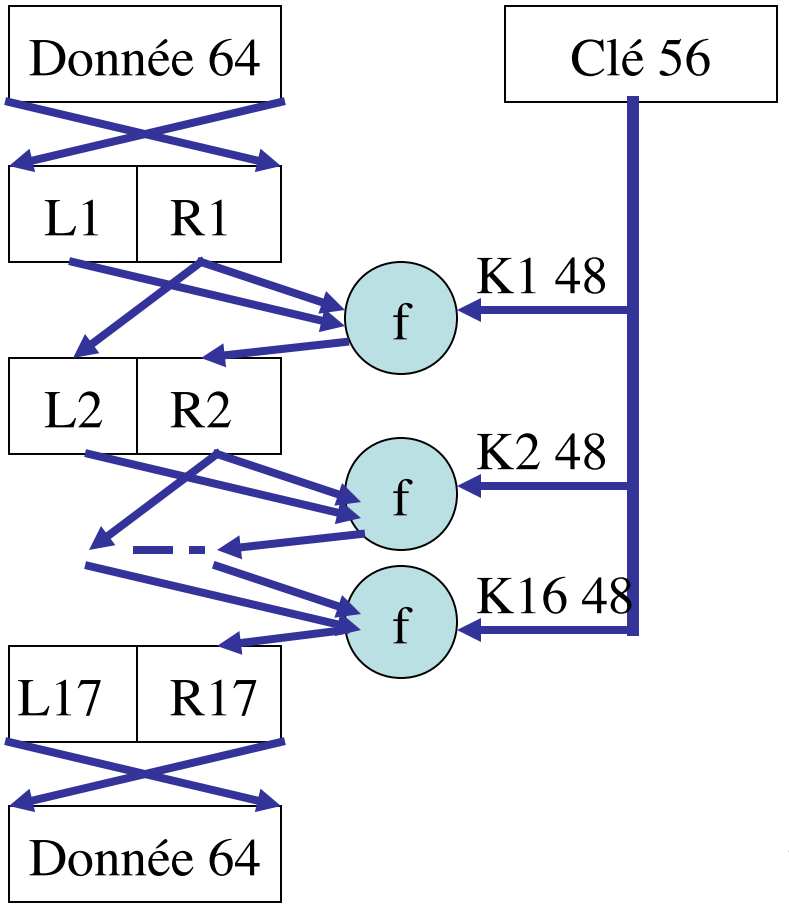
- 20000 attaques réussies par mois sur un ou plusieurs sites dans le monde
- 70000 virus en 2004 (augmentation d'environ 1000 par mois, 10000 actifs, exemple d'un pic infection : 10% du trafic mail)
- 20 milliards de messages SPAM par jour
- Cybercriminalité évaluée à 50 milliards d'euros par an

- Tolérance étonnante des utilisateurs

6.2 Cryptage des données

("Data Encryption Standard" DES 1993)

- Encode des textes de 8 octets (contenant chacun un bit de parité) en utilisant une clé de 56 bits
- Fait en sorte que chaque bit de donnée dépendent de tous les autres et de la clé



Le "défi" DES

- Décryptage de "*Strong cryptography makes the world a safer place*" en 4 mois par des volontaires recrutés sur Internet qui ont testé 18×10^{15} clés
- En 1999 : 22 heures avec un réseau de 100000 machines
- Multiple DES, AES (jusqu'à des clés de 256 bits, chaque tour est constitué d'une substitution non linéaire, d'une permutation linéaire et de l'addition de la clé) : chiffrement en 2Gbit/s sur ASIC

6.3 Cryptage à clé publique

- Non seulement pour le cryptage, mais aussi pour l'authentification et la signature électronique
- Fonctions à sens unique
- Chaque utilisateur A possède 2 clés : une publique (K_A^+) et une privée (K_A^-)
- On peut trouver des clés sûres K_A^+ et K_A^- telles que : $K_A^-(K_A^+(m)) = K_A^+(K_A^-(m)) = m$
  - A!B : $K_B^+(m)$
  - B?A : $m=K_B^-(K_B^+(m))$
- Algorithme RSA (Rivest, Shamir, Adelman)

Algorithme RSA

- Pour recevoir un message crypté, B doit :
  - Choisir 2 entiers premiers p et q
  - Calculer n = pq
  - Choisir e < n premier avec (p-1) et (q-1)
  - Soit d tel que ed=1 mod (p-1)(q-1) (théorème de Bachet de Méziriac)
  - $K_B^+ = (n,e)$, $K_B^- = (n,d)$
- Pour envoyer une suite de bits représentée par un nombre m < n, A envoie $c = m^e \bmod n$
- Pour décrypter, B calcule $m = c^d \bmod n$
- En pratique n est représenté sur 1024 bits

Petit exemple

- p = 5, q = 7
- n = 35, (p-1)(q-1) = 24
- e = 5, d = 29

Texte	Rep. Num	m^e	Code $c = m^e \bmod n$	c^d	$m = c^d \bmod n$	Texte
E	5	3125	10	10000000000000000000000000000000	5	E
N	14	537824	14	17286737396774711015672 16945987584	14	N
S	19	2476099	24	106200365064067167761572429136211990 28224	19	S

L'exponentiation est coûteuse. RSA est plus sûr mais beaucoup moins rapide que DES

La théorie

- Cryptage/décryptage calcule :
  $(m^e \bmod n)^d = (m^e)^d \bmod n = m^{ed} \bmod n$
- Théorème (Euler) : $x^{(p-1)(q-1)} = 1 \bmod n$
- $m^{ed} \bmod n = m^{(1+k(p-1)(q-1))} \bmod n$
  $= m.(m^{(p-1)(q-1)})^k \bmod n = m$
- La sécurité est fondée sur la non connaissance d'algorithme rapide de factorisation (non garantie)

6.4 Authentification

- Preuve en ligne de l'identité
- Il ne suffit pas de donner son identité en clair, car un intrus peut se faire passer pour un autre, y compris en cas d'envoi de mot de passe (qui peut être observé)
- Le cryptage du mot de passe n'est pas une solution (l'intrus peut rejouer le scénario) -> mots de passe à usage unique : notion de "nonce"

Exemple de protocole cryptographique

- A!B : je suis A
- B!A : nonce R
- A!B : $K_A^-(R)$
- B!A : demande K_A^+
- A!B : K_A^+
- B calcule $K_A^+(K_A^-(R)) = R$, authentifiant A

Attaque

- Intrus!B : je suis *A*
- B!Intrus : nonce R
- Intrus!B : $K_{Intrus}^{-}(R)$
- B!Intrus : demande K_{A}^{+}
- Intrus!B : K_{Intrus}^{+}
- B calcule $K_{Intrus}^{+}(K_{Intrus}^{-}(R)) = R$, authentifiant Intrus comme *A* !

Mais A pourrait se rendre compte de cette attaque

Attaque par interception

- A!B : je suis A
- Intrus!B : je suis A
- B!Intrus : nonce R
- Intrus!B : $K_{Intrus}^{-}(R)$
- B!Intrus : demande K_A^{+}
- Intrus!B : K_{Intrus}^{+}
- Intrus!A : R
- A!Intrus : $K_A^{-}(R)$
- Intrus!A : demande K_A^{+}
- A!Intrus : K_A^{+}
- B!Intrus : $K_{Intrus}^{+}(X)$ /* donnée cryptée X */
- Intrus décrypte X = $K_{Intrus}^{-}(K_{Intrus}^{+}(X))$
- Intrus!A : $K_A^{+}(X)$
- A décrypte X = $K_A^{-}(K_A^{+}(X))$

A et B ne se rendent compte de rien…

Protocole Needham-Schroeder, 1978

S est un serveur de clés publiques

- A!S : B
- S!A : $K_S^-(K_B^+,B)$
- A!B : $K_B^+(R_A,A)$
- B décrypte $(R_A,A) = K_B^-(K_B^+(R_A,A))$
- B!S : A
- S!B : $K_S^-(K_A^+,A)$
- B!A : $K_A^+(R_A,R_B)$
- A décrypte $(R_A,R_B) = K_A^-(K_A^+(R_A,R_B))$ /* Retrouver R_A authentifie B */
- A!B : $K_B^+(R_B)$
- B décrypte $R_B = K_B^-(K_B^+(R_B))$ /* Retrouver R_B authentifie A */

Attaque de Lowe, 1995

1. $A!S : I$ /* Le dialogue vers I sera utilisé pour usurper l'id A */
2. $S!A : K_S^-(K_I^+, I)$
3. $A!I : K_I^+(R_A, A)$
4. $I!S : B$
5. $S!I : K_S^-(K_B^+, B)$
6. $I_A!B : K_B^+(R_A, A)$ /* I se fait passer pour A */
7. B décrypte $(R_A, A) = K_B^-(K_B^+(R_A, A))$
8. $B!S : A$
9. $S!B : K_S^-(K_A^+, A)$
10. $B!A : K_A^+(R_A, R_B)$ /* I intercepte : Correction : $K_A^+(R_A, R_B, B)$ */
11. $I!A : K_A^+(R_A, R_B)$
12. A décrypte $(R_A, R_B) = K_A^-(K_A^+(R_A, R_B))$
13. $A!I : K_I^+(R_B)$
14. $I_A!B : K_B^+(R_B)$ /* I se fait passer pour A */

Attaque avec confusion nom et nonce, 1996

6.5 Intégrité

- Signature électronique ("off line") :
  1. A!B : $K_A^-(m)$
  2. B décrypte $K_A^+(K_A^-(m)) = m$ pour vérifier que le document m a bien été signé par A
- Empreintes : utiliser un résumé (analogue à un checksum) H(m)
  1. La signature est $K_A^-(H(m))$
  2. H est une fonction de hachage telle qu'il est très difficile de trouver deux messages x et y tels que H(x) = H(y)

1. Algorithme MD5 : résumé de 128 bits (conjecture : trouver un message d'un résumé donné demanderait 2^{128} opérations)
2. SHA-1 (Secure Hash Algorithm) : 160 bits (US federal standard)

6.6 Controle d'accès

- Un pare-feu est un serveur qui s'intercale entre le réseau administré et le réseau extérieur (il peut être au niveau d'une organisation ou même installé sur votre machine)
  - Filtrage des paquets (sur les adresses IP ou les ports TCP, sur le type des messages : TCP Syn, TCP ack, messages ICMP). Par exemple, filtrage de Telnet pour interdire les connexions distantes, ou filtrage d'UDP pour interdire les applications des vendeurs audio-vidéo offrant un mode UDP par défaut. Utilisation de listes noires. Utilisation du bit ack pour rendre dissymétrique la connexion (utilisation de serveurs externes depuis l'intérieur, mais interdiction d'utiliser des serveurs internes depuis l'extérieur) -> besoin de formaliser une "politique de sécurité" (actuellement mis en œuvre par un ensemble de règles)
  - Passerelle d'application : exemple de la machine transit de l'Irisa (seules les connexions telnet de cette machine ne sont pas filtrées). Cette application peut dialoguer avec l'utilisateur pour l'authentifier. L'inconvénient est qu'elle dépend de l'application considérée. Preuves de sécurité des serveurs ?

6.7 Attaques et contre-mesures

- Scrutation ("mapping"): "ping" peut être utilisé pour trouver des adresses IP (celles qui répondent). Idem au niveau TCP sur les ports. -> les pare-feux peuvent repérer les comportements de scrutation.
- Espionnage des paquets ("packet sniffing") : facile à l'intérieur d'un réseau local Ethernet. Les trames espionnées sont passées à un programme décodeur pour récupérer l'information applicative pertinente (mots de passe par exemple) : exemple du FBI Carnivore à travers l'ensemble du réseau. -> détection des machines utilisant le mode d'observation (adresse IP de diffusion). -> cryptage systématique.
- Usurpation d'identité ("spoofing") : changer l'adresse source IP -> demander aux routeurs de vérifier que les adresses entrantes sont correctes.

- Dénis de service (“Denial of service attacks”) : rendre le réseau inutilisable. Exemples : 1/ inondation de demandes de connexions TCP avec des adresses usurpées. 2/ envoi de fragments sans jamais compléter le paquet. 3/ envoi d’enquêtes ICMP pour un destinataire usurpé.
- Dénis par attaques réparties : installation clandestine d’attaquants et déclenchement synchronisé. -> difficile à contourner. -> vers une plus grande traçabilité des échanges.
- Prise d’otage (“Hijacking”) : un intrus s’interpose entre les partenaires

Exemples d'attaques pas dénis de service

- 8 décembre 2010 : les sites de PayPal, MasterCard et PostFinance (banque suisse) sont attaqués. Les deux derniers restent inaccessibles durant plusieurs heures.
- 6 février 2007 : attaque sur les serveurs racine du DNS. 6 des 13 serveurs sont affectés, 2 le sont sévèrement.
- 25 juin 2009 : le nombre de recherches concernant la mort de Michael Jackson est tellement important que Google News croit à une attaque automatique.

6.8 Applications sécurisées

- Messagerie cryptée : PGP (“Pretty Good Privacy”)
- SSL (“Secure Sockets Layer”) et TLS (“Transport Layer Security”)
- Ipsec : une pile IP pour la sécurité
  - Authentification Header Protocol (AH)
  - Encapsulation Security Payload Protocol (ESP)
- Sécurité radio : espionnage autour des batiments -> protocole WEP (“Wired Equivalent Privacy”).

6.9 Un peu de virologie
[Eric Filiol, Springer/IRIS]

- Auto-reproduction :
  - Autoprint :

```
p="p=%c%s%c;main(){printf(p,34,p,34);}";main(){printf(p,34,p,34);}
```

    voir aussi : http://www.nyx.net/~gthompso/quine.htm
  - Automates cellulaires (constructeurs universels) :

    Von Neumann (1948), Codd (1968), Langton (1984), Byl (1989), Ludwig (1993)
- Notion de virus (fin 1970 (officiellement))

  Thèse Fred Cohen (1986), directeur Léonard Adleman (US)

Table de transitions de Byl2 (CHDBG-N)

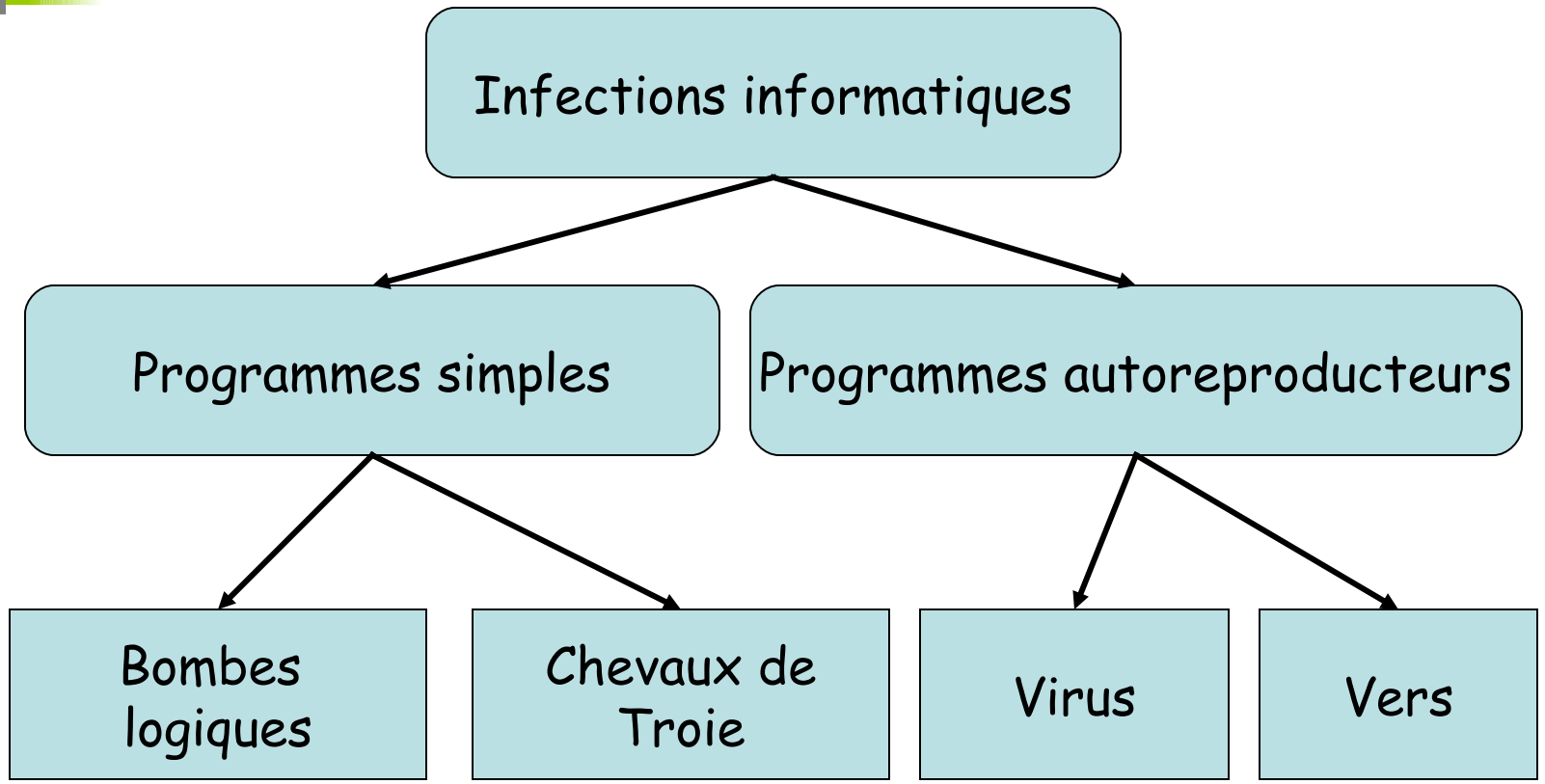
- 00003-1/00012-2/00013-1/00015-2/00025-5/00031-5/00032-3/00042-2/0****-0
- 10000-0/10001-0/10003-3/10004-0/10033-0/10043-1/10321-3/11253-1/12453-3/1****-4
- 20000-0/20015-5/20022-0/20202-0/20215-5/20235-3/20252-5/2****-2
- 30001-0/30003-0/30011-0/30012-1/30121-1/30123-1/31122-1/31123-1/31215-1/31223-1/31233-1/31235-5/31432-1/31452-5/3****-3
- 40003-5/40043-4/40212-4/40232-4/40242-4/40252-0/40325-5/4****-3
- 50022-5/50032-5/50212-4/50222-0/50322-0/5****-2

Une séquence reproductrice

22	22	
2312	2342	...
2342	2332	
25	22	

Duplication en 25 pas

Taxonomie


Infections informatiques
Programmes simples
Programmes autoreproducteurs
Bombes logiques
Chevaux de Troie
Virus
Vers

Principes de l'infection

- Le programme infectant est porté par un programme hôte ("dropper")
- Lorsque le dropper est exécuté :
  - Le programme infectant prend la main
  - Puis il la rend au programme hôte sans trahir sa présence
- Scénario d'attaque :
  - Recherche des programmes cibles (fichiers exécutables, évitement de la sur-infection (signature exploitable))
  - Copie du code dans la cible
  - Programme d'anti-détection
  - Charge finale (différée ou non)

Infections simples

- Bombes logiques (souvent les charges finales). Souvent différées (exemple du programmeur système)
- Chevaux de Troie (“trojans”) :
  - Sa partie serveur est installée à l’insu de la victime (téléchargement à partie de sites leurres par exemple)
  - Celle-ci donne discrètement à l’attaquant l’accès à des ressources
  - Le client (attaquant) recherche les machines infectées sur le réseau et en prend le contrôle
- Leurres (fausses bannières par exemple)

Modes d'action des virus

- Par écrasement de code :
  - Entête : le code infecté devient non exécutable
  - Ailleurs : il faut insérer une instruction de saut vers le début du virus (sinon, permet un semblant de furtivité, l'erreur se déclenchant après un début d'exécution)
  - Remplacement (détectable car les fichiers infectés ont tous la même taille)
- Par recouvrement de code :
  - Prepend : difficile à cause des adresses à recalculer
  - Append : insertion d'un saut initial, puis restauration

- Par entrelacement de code :
  - Utilise le format PE des exécutables Windows dans lequel le code est fragmenté par plages fixes non nécessairement remplies. Le virus peut se glisser dans les espaces libres (la taille de l'exécutable infecté reste inchangée)
- Par accompagnement de code :
  - Le code viral identifie une cible et duplique son code en créant un fichier supplémentaire
  - Lorsque l'utilisateur exécute le programme cible, la copie virale est exécutée en premier (propagation) :
    - Exécution préemptive (sous DOS, f.com -> f.exe -> f.bat)
    - Utilisation du PATH
    - Renommage et sauvegarde du fichier cible
- Virus de code source (stdio.h par exemple)

Quelques virus

- Elk Cloner (1981) : premier virus « dans la nature ». Transmis par les disquettes, infecte Apple II. Affiche un court texte tous les 50 allumages de l'ordinateur.
- Storm (2007) : cheval de Troie transmis par mail. Botnet pair-à-pair. Taille du réseau : entre 1 et 10 millions de machines.

Techniques anti-antivirales

- Furtivité :
  - On cache le virus dans des zones déclarées faussement défectueuses par exemple
  - Désinfection après activation
- Polymorphisme :
  - Le virus se réécrit sous une forme différente (changement de signature)
  - Chiffrement : le début du code en clair est de déchiffrer la suite (la procédure de chiffrement peut aussi être changée à chaque infection)
- Attaque des anti-viraux

Techniques antivirales

- Recherche de signatures (base de signatures à mettre à jour)
- Analyse spectrale : indice de programmation "déviante" par rapport à ce que produit un compilateur
- Règles heuristiques
- Contrôle d'intégrité des fichiers
- Surveillance comportementale
- Emulation du code dans une zone confinée