FORMATION ISN 2011

RESEAUX

Jean-Yves Ramel
Pascal Bourquin

Plan

- Introduction (JYR - 1h)
  - « Explication de texte » du programme réseaux en ISN
  - Notions fondamentales des réseaux
  - Modèle OSI
  - Transmission de l'information (OSI 1)
- OSI n°2 (PB - 30')
  - Accès au support, trames
  - Etude d'un protocole : HDLC
- OSI n°3 (JYR & PB - 1h30)
  - Interconnexion de réseaux locaux
    - Commutation
  - IP
    - Adressage, routage
    - Protocoles inter-routeurs (RIP, OSPF)

Réseaux

- Réseaux = 10% science + 90% « bidouille »
- Les 10% de science
  - Modèle théorique de classification des fonctions réseaux et de leurs interactions
  - Spécification et preuves de protocole
    - Automates d'états finis
  - Transmission de l'information
    - Théorie de l'information, codage, modulation de porteuse, bande passante, etc.
    - Technologies : filaire, optique, hertzien, etc.
  - Algorithmes décentralisés et leurs protocoles
    - Algorithmes de partage équitable : CSMA/CD, CSMA/CA, jeton, etc.
    - Algorithmes basés sur les graphes : routage IP, RIP/OSPF, spanning tree Ethernet
  - Sécurité des échanges réseau
    - Cryptographie et protocoles : intégrité, confidentialité, authentification
  - Ingéniérie du trafic réseau
    - Probabilités : congestion, lissage du trafic, gestion de files d'attente, etc.
- Les « bidouilles »
  - Assurer une compatibilité ascendante
  - Etre fin politiquement (normalisations, compatibilité avec le concurrent, législation, etc.)
  - Travailler chez CISCO est un avantage pour pouvoir imposer ses idées ☺

Extraits du programme → plan de la ½ journée

Notions fondamentales

Modèle OSI

OSI n°1

OSI n°2

HDLC

- **Réseaux**

Savoirs	Capacités	Observations
Transmission point à point Principes de base d'une transmission d'informations numériques entre un émetteur et un récepteur.	♦ **Établir** une communication sérielle entre deux machines.	On s'interroge sur la qualité d'une liaison série point à point. On se limite à l'analyse d'un trafic de type « chat » (échange de caractères codés). On introduit la notion de protocole (règles, formats et conventions, sur lesquels il est nécessaire de s'accorder pour communiquer). Au-delà de deux machines, le modèle de la liaison point à point ne convient plus.
Adressage sur un réseau Mécanismes d'adressage pour identifier des machines distantes.	**Décrire** une situation d'adressage sur un type de réseau particulier. ♦ **Analyser** le trafic (trames) sur un réseau et mettre ainsi en évidence la notion de protocole.	On introduit ces notions en comparant différents types d'adressages existants (téléphone, courrier postal). On fait appel à un outil d'analyse pour visualiser la transmission des trames nécessaires au dialogue entre machines numériques.

Extraits du programme → plan de la ½ journée

Commutation

OSI n°3

- IP
- Routage IP
- RIP/OSPF

Routage Mécanismes induits par la communication sur un réseau dont la structure est de type graphe. Notions de paquets, de chemins, de routage.	♦ **Analyser** les entêtes de messages électroniques, pour décrire le chemin suivi par l'information.	On se limite à la mise en œuvre d'une séance de travaux pratiques, avec analyse d'entêtes de courriels prédéfinis reçus (aspect distribué et non fiable des réseaux de grande taille, difficulté du passage à l'échelle). On explique la différence entre les réseaux de type arborescent et de type graphe.
Supranationalité des réseaux	**Prendre conscience** du caractère supranational des réseaux et des conséquences sociales, économiques et politiques qui en découlent.	On met en évidence le fait que certains pays autorisent la mise en ligne d'informations, services ou contenus numériques dont la consultation n'est pas permise dans d'autres pays.

NOTIONS DE BASE

Introduction générale, définitions,
Topologie et Architecture, Modèle OSI

Qu'est-ce qu'un réseau ?

Définition : groupe d'ordinateurs (ou périphériques) reliés les uns aux autres afin de permettre aux utilisateurs d'échanger des informations et de partager du matériel

- **Du matériel**
  - Câbles, supports
  - Cartes réseau
  - Nœuds intermédiaires
  - Equipements terminaux
- **Du Logiciel**
  - Système d'exploitation
  - Protocoles
  - Pilotes
  - Logiciels réseaux

Utilité : communication, partage de ressources, travail en groupe

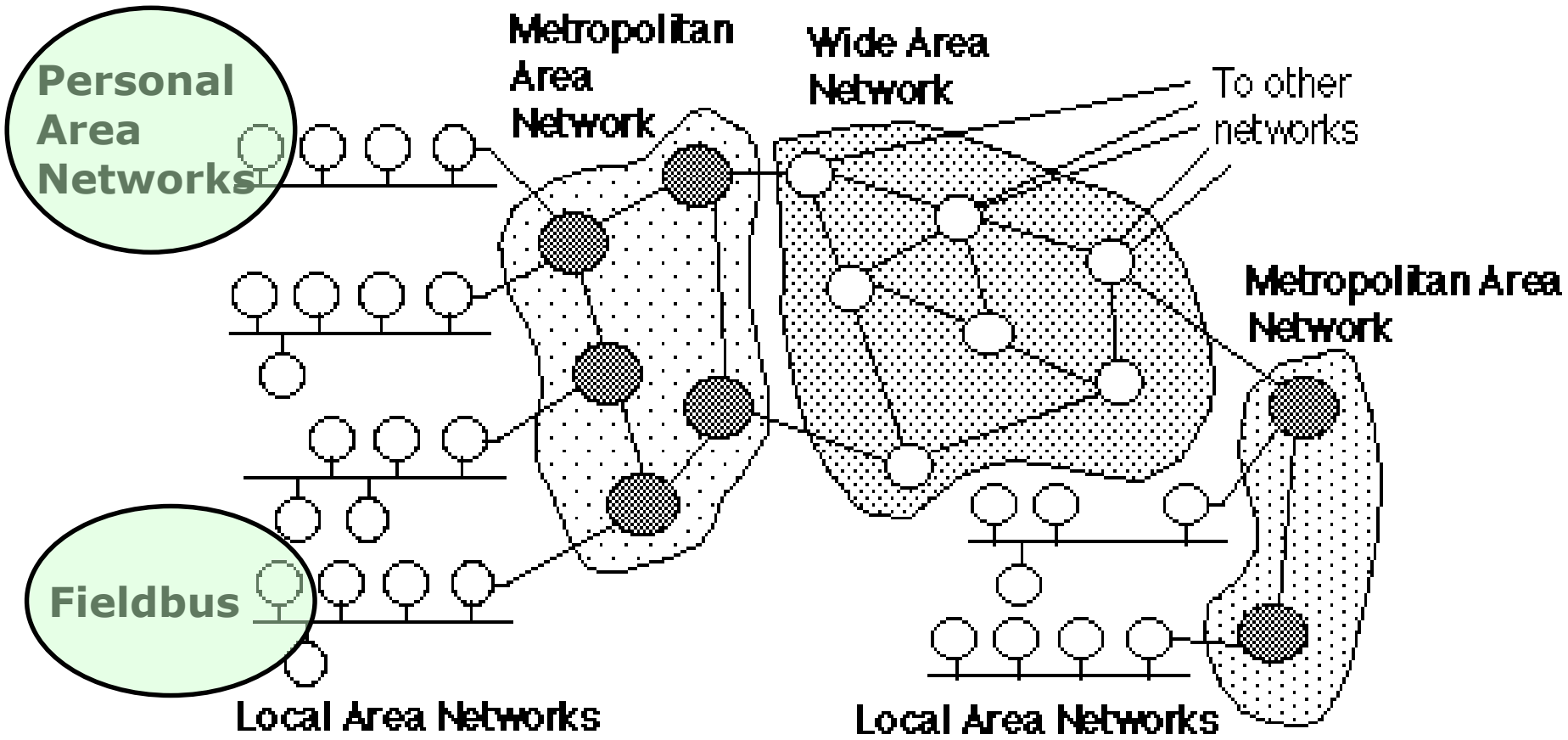
Exemples : téléphone, Ethernet, Internet

Notion de communication

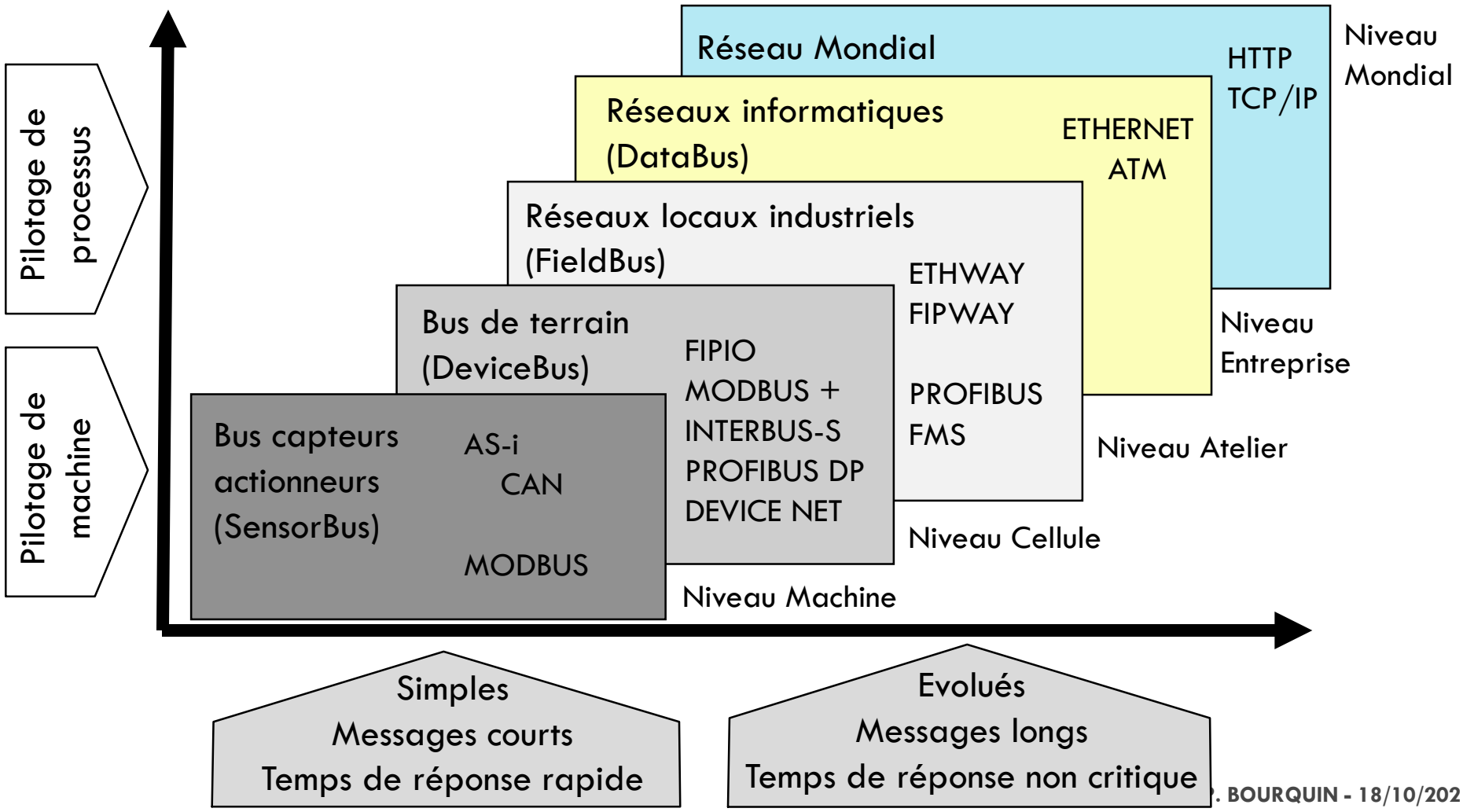
- La communication définit l'action d'échanger des informations. Cela induit un mécanisme de transmission (aspect physique) et la capacité du récepteur à recevoir et à réagir (aspect logique)
- Un système de communication complet doit permettre un transfert d'information transparent et fiable, en temps utile, d'une source vers un ou plusieurs collecteurs.
- Nécessite
  - Le respect de contraintes fonctionnelles, technologiques, et économiques

  → Un (des) langage(s) commun(s)

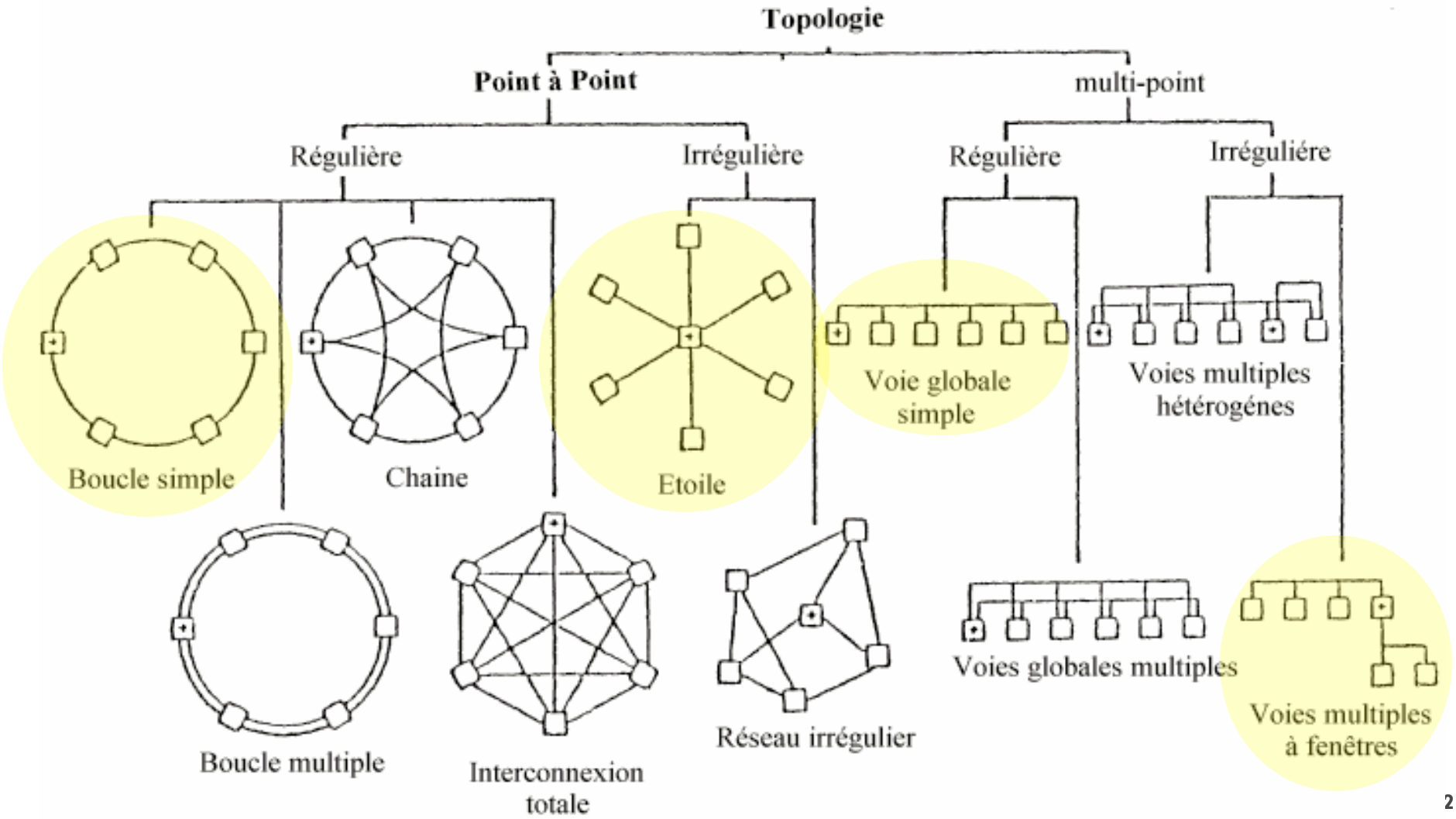
Fieldbus - PAN - LAN - MAN - WAN



Modèle de l'usine intégré (CIM)



Topologie et Architecture



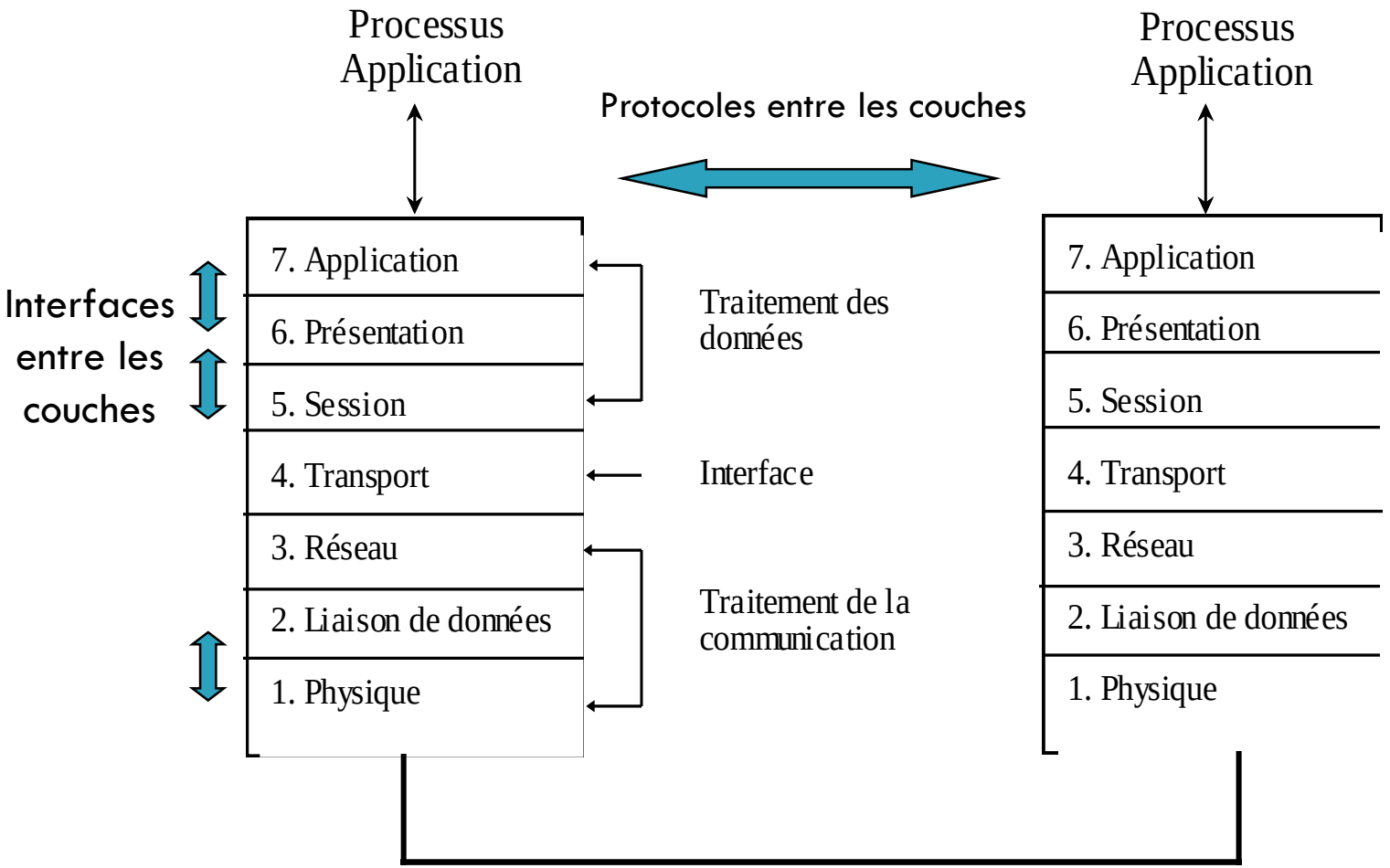
Le modèle OSI

- Le modèle d'Interconnexion des Systèmes Ouverts (*Open Systems Interconnection*) a été proposé par l'ISO (*International Standards Organization*) en 1977
- Une norme internationale pour une architecture multi-couches qui permet l'interconnexion de matériels hétérogènes.
- Pourquoi un modèle en couches ?

  1° Facilité de développement et de modification : une couche (un protocole) peut être modifiée de façon indépendante tant que l'interface avec les deux couches adjacentes reste inchangée

  2° Intéropérabilité : une couche de niveau *n+1* peut utiliser les services de couches de niveau *n* très différentes à condition que l'interface *n/n+1* soit la même

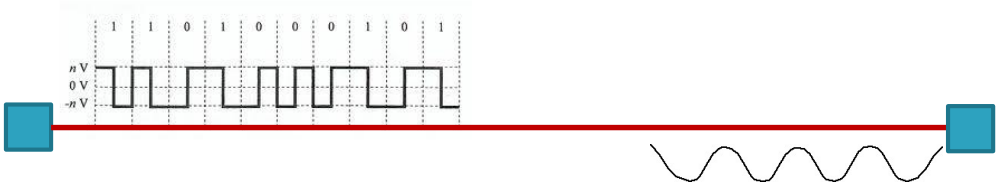
Architecture multi-couches



Architecture multi-couches

Couche Physique ou niveau 1

- "Elle décrit les interfaces mécaniques, électriques, fonctionnels et procédurales nécessaires à l'activation, au maintien et à la désactivation des **connexions physiques** destinées à la transmission de bits entre deux entités de liaison de données"
- Ce niveau est chargé de piloter le **matériel de transmission**
- C'est donc dans cette couche que sont définis le **support physique** ou médium, les signaux, les voies de transmission ou canaux, le raccordement des communicateurs, les débits, …
- Les entités principales sont le **signal analogique et le bit**



Architecture multi-couches

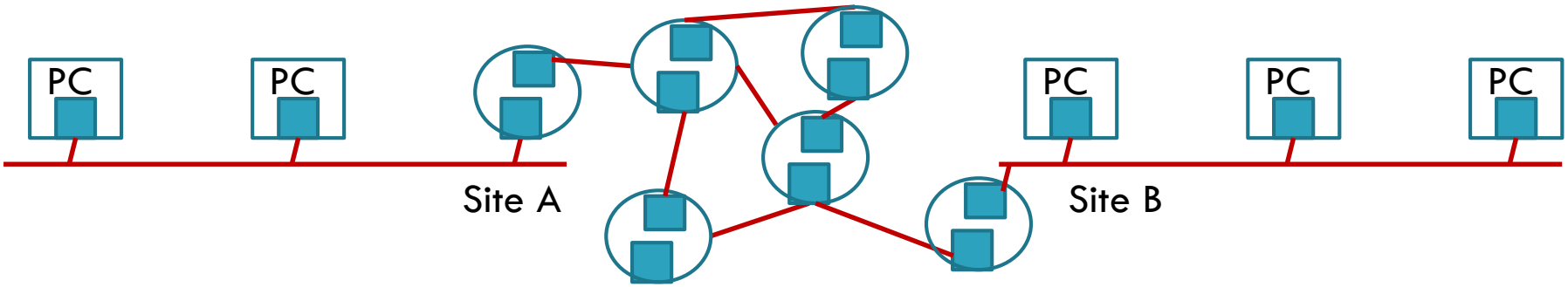
Couche liaison de données ou niveau 2

- Elle permet le transfert **fiable** d'informations entre des systèmes **directement connectés**. Elle fournit les moyens **procéduraux** nécessaires à l'établissement, au maintien et à la libération des connections de liaison de données **point à point**
- Deux sous-couches :
  - Sous-couche MAC (*Medium Access Control*) qui gère l'accès à la voie de transmission. Parfois, liée aux choix du niveau 1(une exception à l'indépendance entre les couches)
  - Sous-couche LLC (*Logical Link Control*) qui regroupe l'aspect logique de la transmission entre systèmes physiquement connectés. C'est à ce niveau que se situent la détection des erreurs et la gestion **de trames**

Architecture multi-couches

Couche réseau ou niveau 3

- Ce niveau est chargé de **l'acheminement et de la communication en paquets** d'information ainsi que de la gestion des connexions réseaux entre sites distants
- Fonctions principales : **le routage,** la recherche du chemin, la gestion de l'adressage global



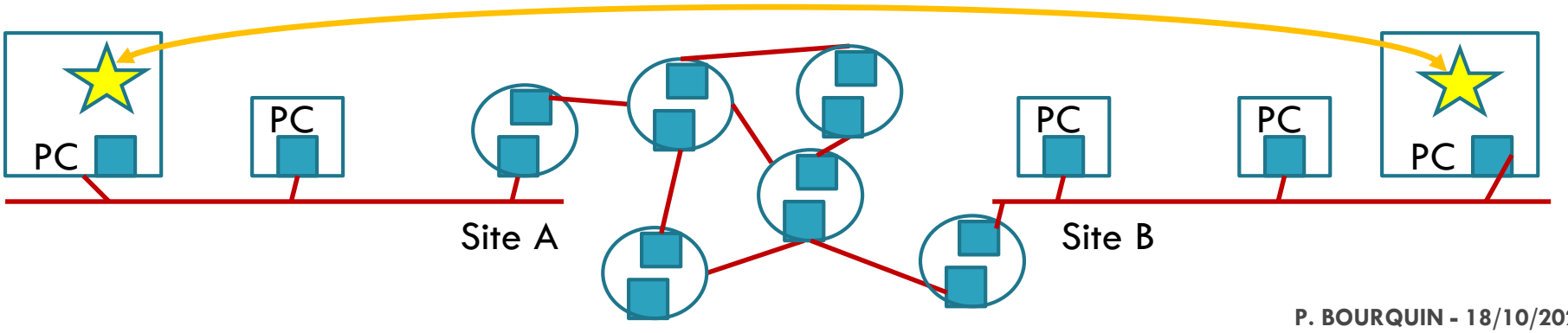
Architecture multi-couches

Couche transport ou niveau 4

Assure un transport de données transparent entre entités de session en les déchargeant complètement des détails d'exécution d'un transfert de données **fiable** et d'un bon rapport qualité/prix **de bout en bout**

Fonctions : contrôle de flux, fragmentation et réassemblage des messages, contrôle d'erreur (perte, duplicata de paquets, modifications), séquencement des messages

Au dessus de la couche transport, le message doit avoir été expurgé de sa connotation communication.



Architecture multi-couches

Couche session ou niveau 5

- Elle fournit des outils de synchronisation et de gestion du dialogue entre entités communicantes
- Fonctions principales : gestion des interruptions, des reprises, checkpoints,…

Couche présentation ou niveau 6

- Elle se charge de la **représentation des informations** que des entités s'échangent, ou auxquelles elles se réfèrent au cours de leur dialogue
- Elle est chargée de décrire de manière cohérente les données et de les coder sous une forme universelle dans le réseau
- Gère une partie des problèmes de sécurité, en particulier ceux relatifs à la sûreté du contenu des messages. Le **cryptage/décryptage** est donc un des services présents dans cette couche

Architecture multi-couches

Couche application ou niveau 7

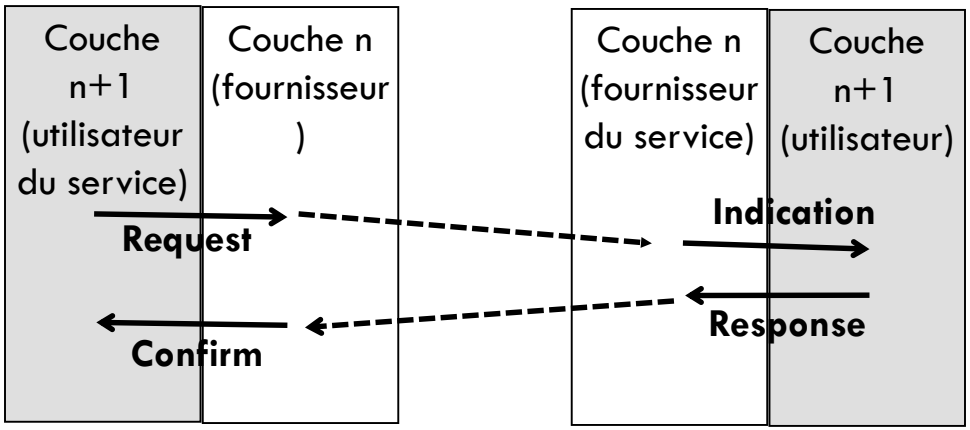
- C'est la couche chargée de la communication entre les processus application et le modèle OSI.
- Elle définit les formats de données spécifiques à une application (mail, ftp, web, …)
- C'est la seule couche ouverte vers l'extérieur. Toute normalisation est donc très difficile.

Modèle OSI : Les Services

- La couche N+1 peut accéder aux services de la couche N via des **points d'accès** au service (**SAP** : *Service Access Point*). Chaque SAP est identifié par une « adresse » unique

- <u>4 types de primitives de service :</u>

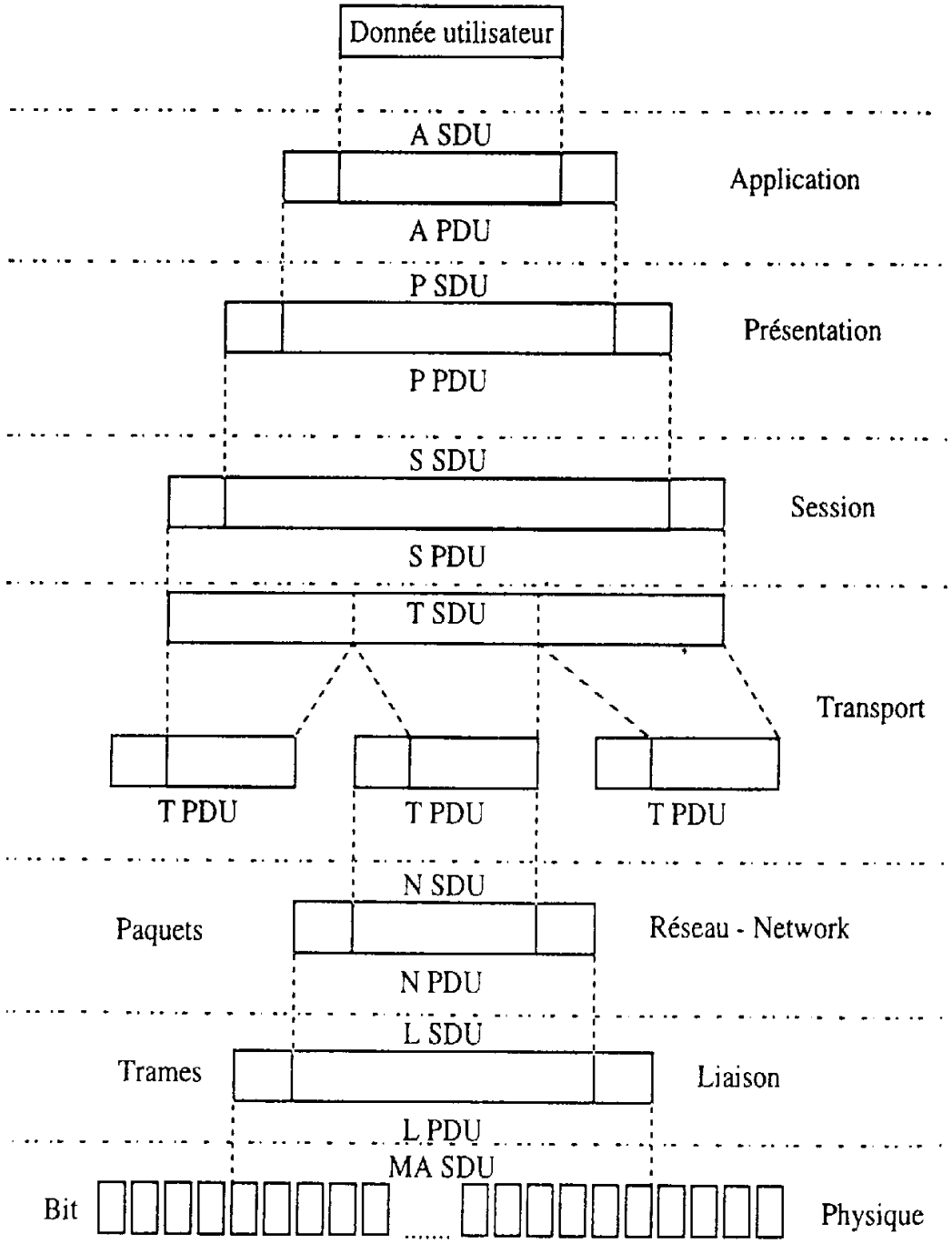
Request	requête (1 entité sollicite un service)
Indication	indication (1 entité est informée d'un événement)
Response	réponse (une entité répond à un événement)
Confirm	confirmation (demande de service bien reçue)



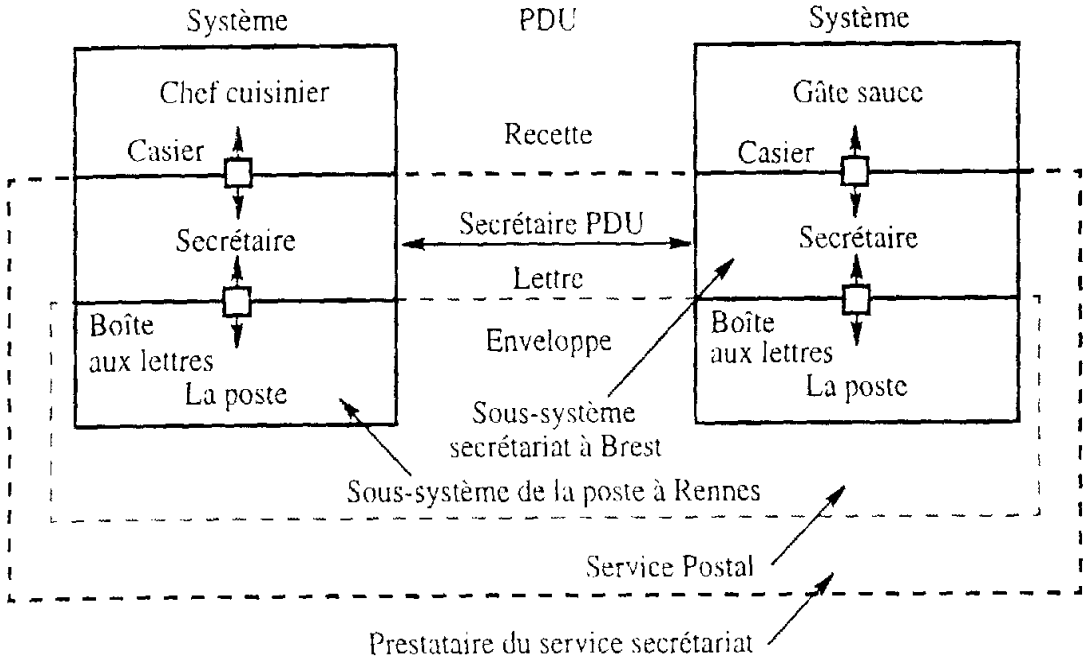
La syntaxe générale est :
Niveau.Fonction.Primitive

Fonction = Connexion, Libération (ou Données.

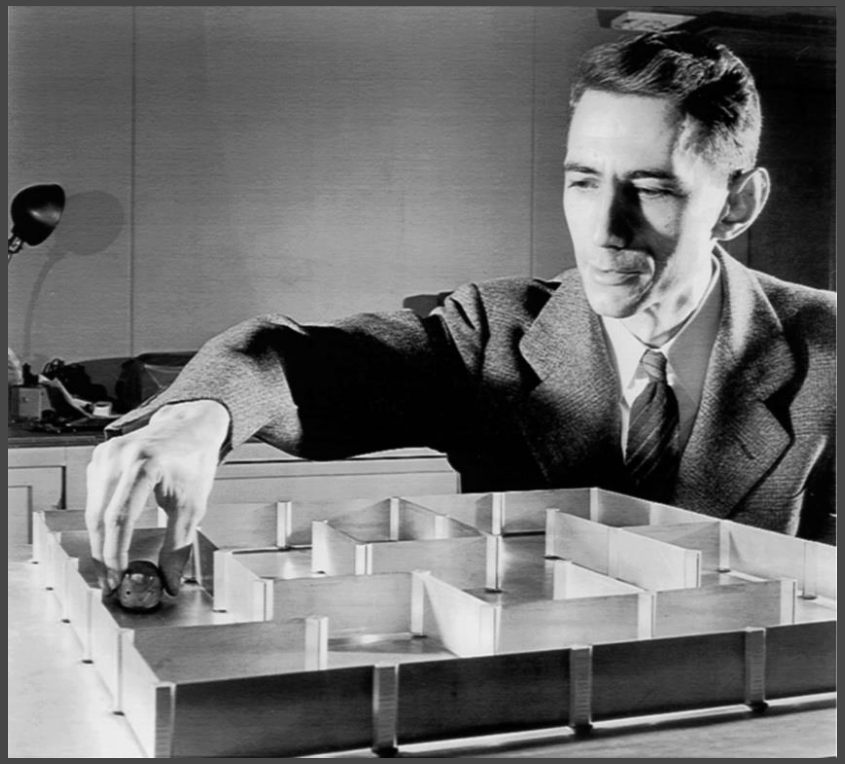
Services confirmés ou non


Donnée utilisateur
A SDU
Application
A PDU
P SDU
Présentation
P PDU
S SDU
Session
S PDU
T SDU
Transport
T PDU
T PDU
T PDU
N SDU
Paquets
Réseau - Network
N PDU
L SDU
Trames
Liaison
L PDU
MA SDU
Bit
Physique

La métaphore



Service	Entités protocolaires homologues	PDU échangées	Points d'accès
Cuisine	Chef cuisinier Gâte-sauce	Recettes	Restaurant
Secrétariat	Secrétaires	Lettres	Casier
Postal	Postiers	Enveloppes	Boîtes à lettres

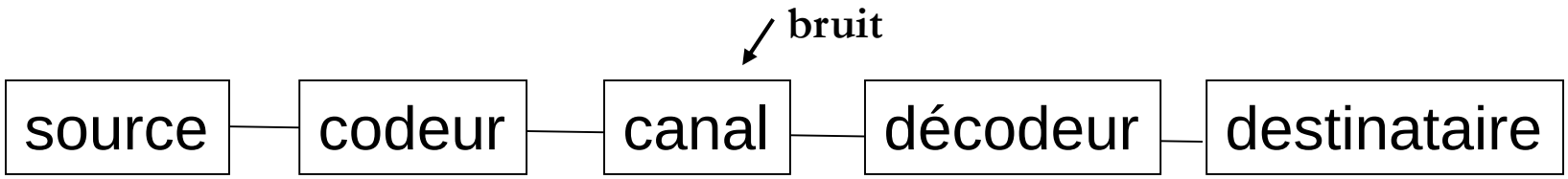


Claude Elwood Shannon (1916-2001)

OSI-1, TRANSMISSION DE L'INFO

OSI-1, théorie de l'information, codage, compression, transmission de l'information, …

Théorie de l'information

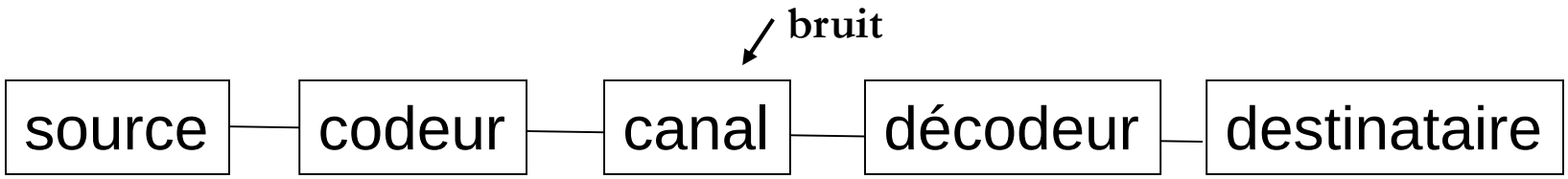


Vision probabiliste :

- Source d'information : siège d'événements *aléatoires* qui constituent le message
- Signal : phénomène physique pouvant représenter des données porteur d 'une information
- Message : lot d 'informations formant un tout intelligible et exploitable transmis en 1 seule fois. Séquence finie de signaux (lettres) qu'une source transmet à un destinataire

- Un constat : la transmission d'un message certain est inutile
- Quantité d 'information : mesure quantitative de l 'incertitude d 'un message en fonction du degré de probabilité de chaque signal composant le message

Modèle d'un système de comm.

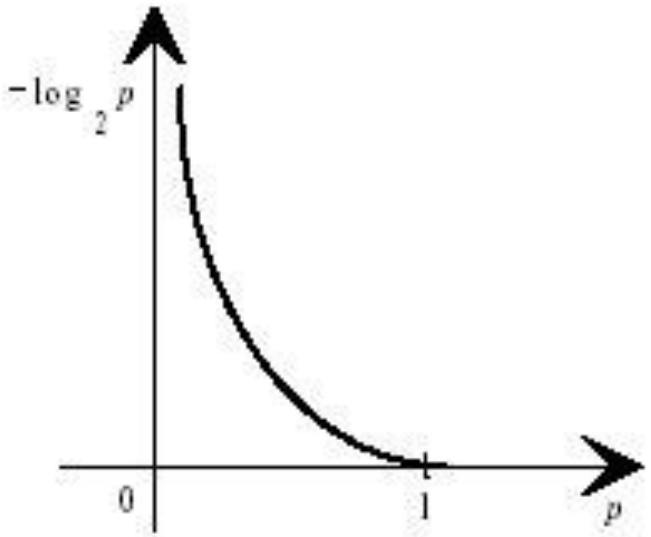


- Caractèrisation quantitative
- Source : générateur de message → Entropie
- Canal : transmet et dégrade le message →Capacité
- Des messages différents portent la même information, le **codage** cherche le message avec les meilleurs propriétés :
  - Codage de source → supprime la redondance, réduit le coût
  - Codage de canal → protège contre les perturbations
  - Cryptage, chiffrement → protège contre les curieux

Information, grandeur mesurable ?

- Selon Shannon,
- Quantité d'information de x :

$$I(x) = \log(1/p(x)) = -\log(p(x))$$



- Si log base 2, alors I(x) s 'exprime en bit
- $I(x_k)$ est aussi appelé Self-information de la source

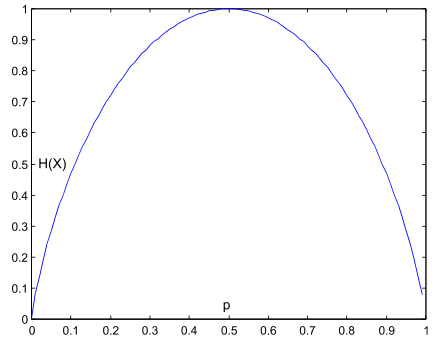
Entropie d'une source d'information

source d'information

$$X = \{x_1, \ldots, x_k, \ldots, x_K\}$$

$$\mathbf{p} = \{p(x_1), \ldots, p(x_k), \ldots, p(x_K)\}$$

$$H(X)$$



Source binaire

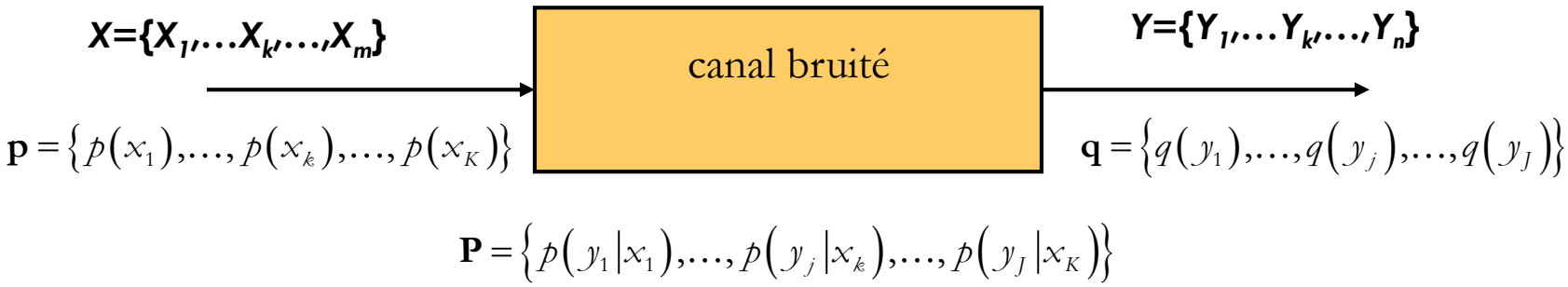
Entropie : Quantité d'information moyenne de la source

$$H(X) = E(I(X)) = \sum_{i=1}^{N} p_i.\log(1/\ p_i) = -\sum_{i=1}^{N} p_i.\log(\ p_i)$$

Hyp : source discrète finie stationnaire sans mémoire

Emission = Variable aléatoire - $\Sigma p_i = 1$

Capacité d'un canal



Capacité d 'un canal : la plus grande quantité d'information moyenne qu'il est capable de transmettre de son entrée à sa sortie

Autres définitions :

Quantité maximum d'information que l'on peut transmettre via un canal avec une probabilité d'erreur arbitrairement faible

Le maximum de l'information mutuelle moyenne I(X;Y) avec X en entrée, Y en sortie

Capacité d'un canal

- **Information mutuelle**

$$I(x_k ; y_k) = \log(1/\, p(x_k ; y_k)) = \log(p(x_k / y_k)/\, p(x_k))$$

Exprime le lien qui existe entre un symbole émis et un symbole reçu

  - H(X) caractérise la source
  - I(X;Y) dépend de la source → p(x)
  - I(X;Y) dépend du canal → p(x/y) = P
  - I(X;Y) varie entre 0 ≤ I(X;Y) ≤ H(X) → on définit C

- **Cas extrêmes**
  - I(X;Y) = H(X) → canal non bruité (parfait)
  - I(X;Y) = 0 → canal bruité (aléatoire)

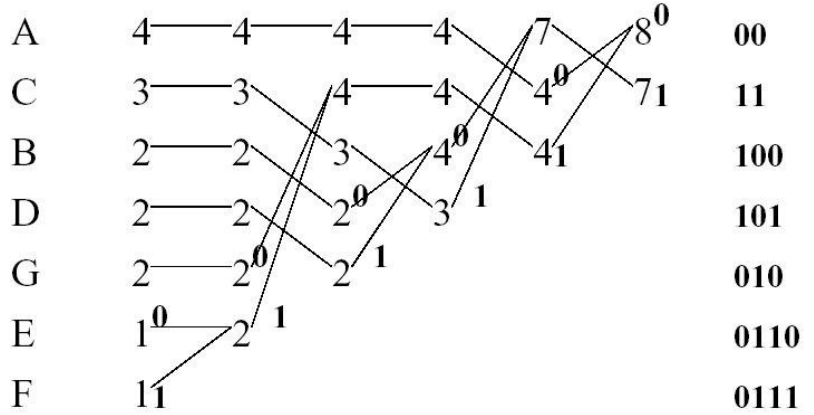
Codage Source

- Adapter la source au canal :
  - l'alphabet
  - le débit
- Utiliser la capacité du canal → maximiser I(X,Y)
- Codeur de source → Supprimer la redondance
- Compression

Codage Source

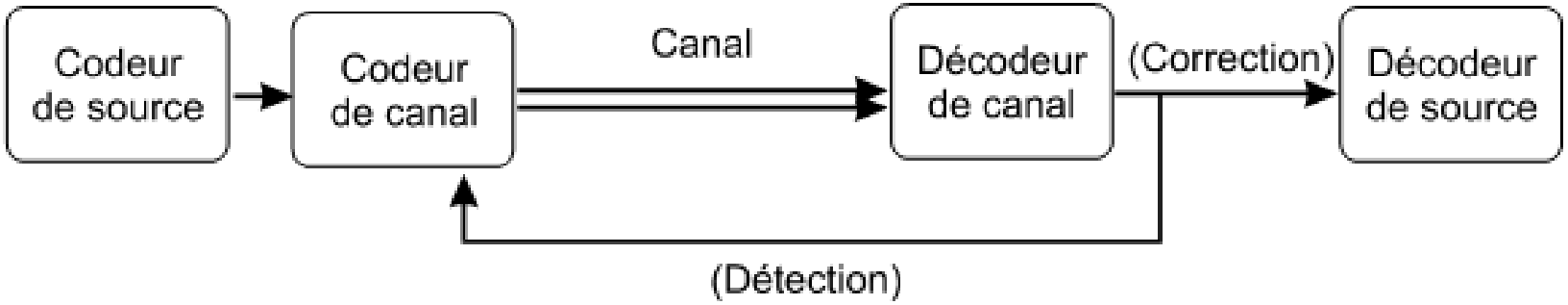
- Génération d'un **codage optimal absolu,** pour des sources divisibles récursivement (jusqu'à un symbole par ensemble) en deux sous-ensembles équiprobables
- Codage de Huffman
- Codage de Shannon-Fano

Symboles s_k	Proba $p(s_k)$	Mots-codes c_k	Longueur l_k
s_1	0.25	00	2
s_2	0.25	01	2
s_3	0.125	100	3
s_4	0.125	101	3
s_5	0.0625	1100	4
s_6	0.0625	1101	4
s_7	0.0625	1110	4
s_8	0.0625	1111	4

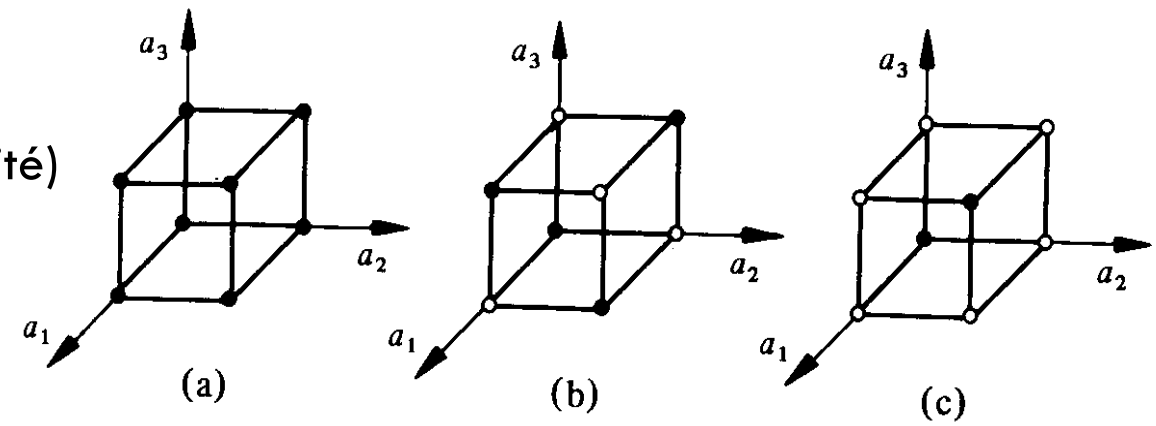


Symbole	Fréquence
A	4
C	3
B	2
D	2
G	2
E	1
F	1

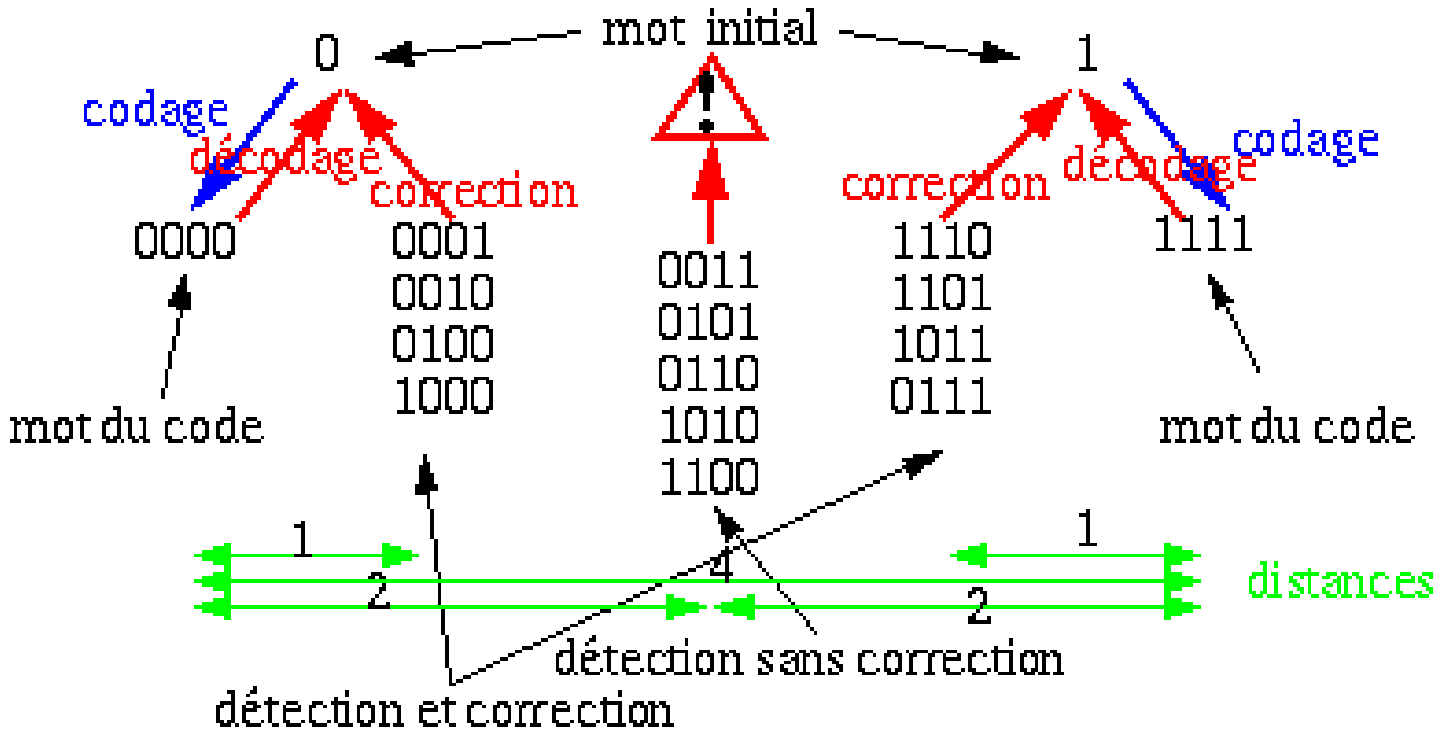
Codage Canal



- Détecter et/ou corriger les erreurs de transmission
- Introduire de la redondance utilisable
- Détection et correction
- Codes linéaires
  - Checksum (bit de parité)
  - Codage de Hamming
- Codes cycliques
- Codes convolutifs

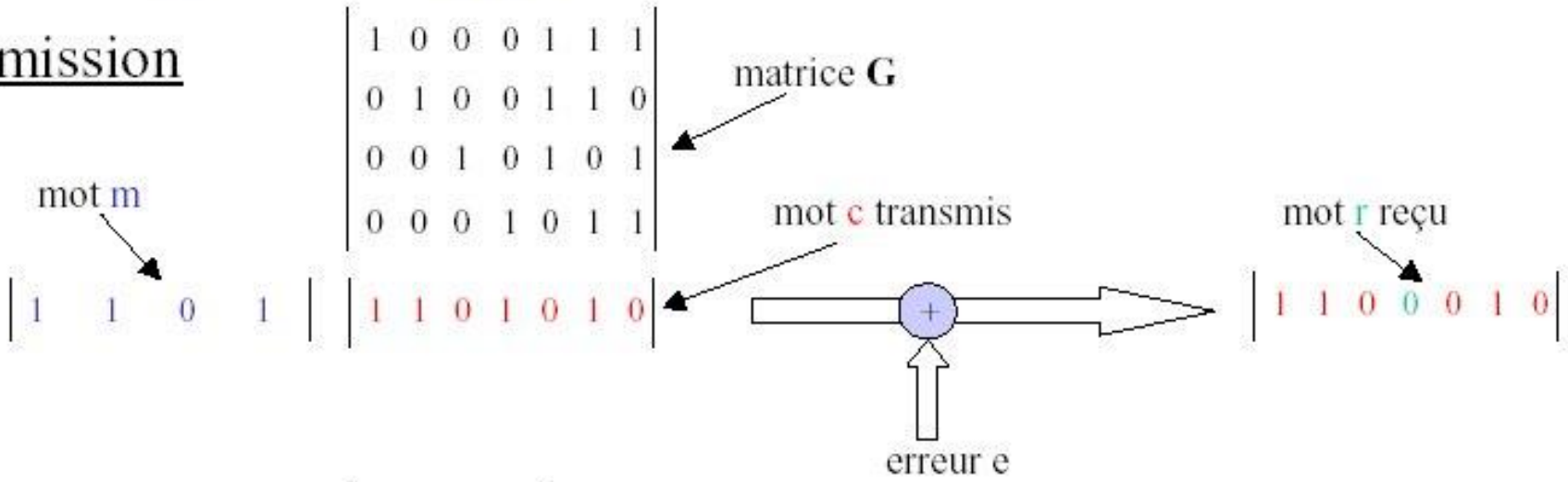


Détection et correction des erreurs

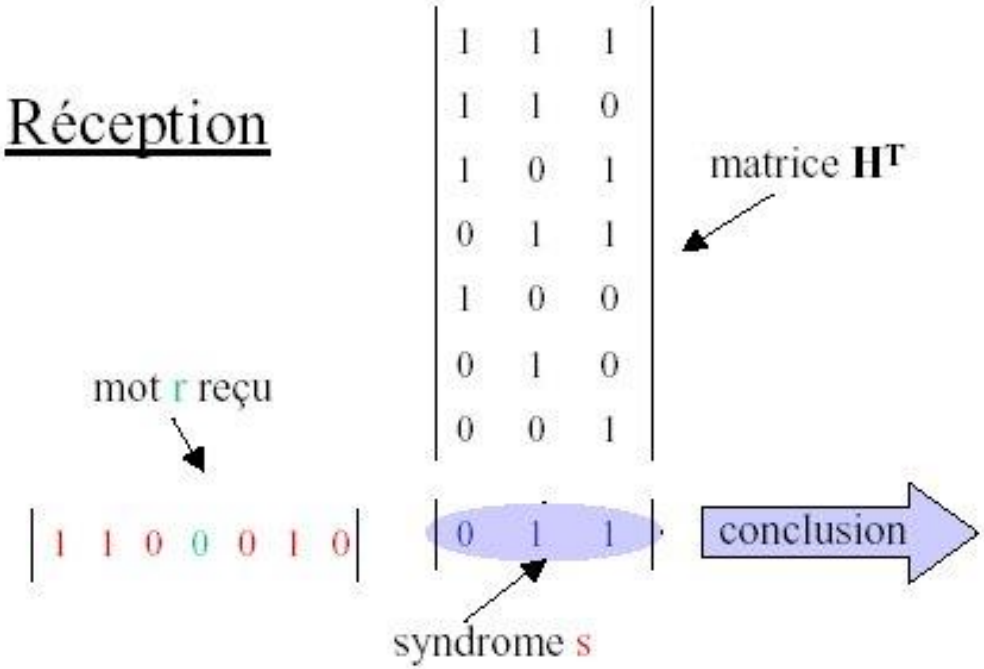


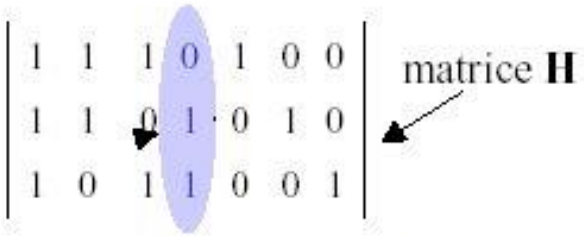
Exemple : mot m à transmettre ‘1 1 0 1’

Emission



Réception



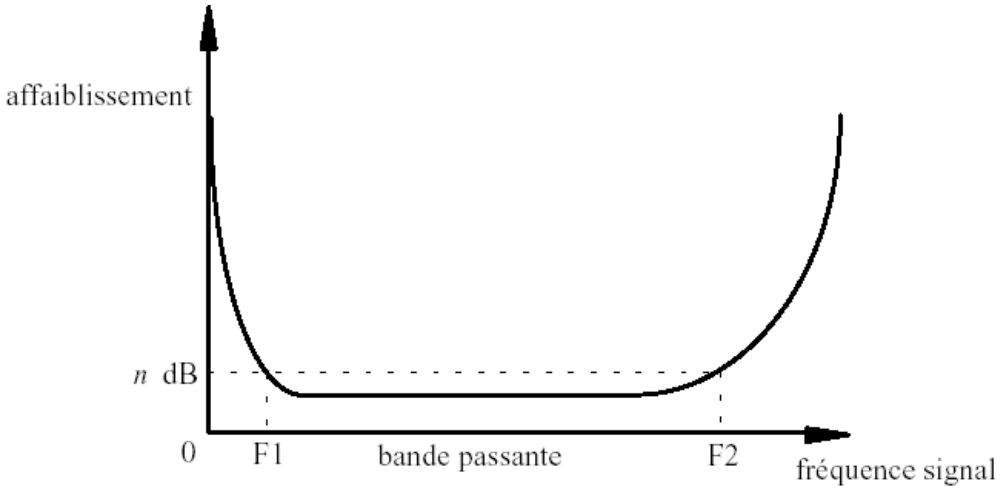


L’erreur est en *quatrième* position : le mot de **n** bits corrigé s’écrit : ‘1 1 0 1 0 1 0’, et le mot de **k** bits est : ‘1 1 0 1 ‘; ce sont les quatre premiers bits.

Voies de transmission et codage

- **Bande passante W :** La bande passante d'un canal de transmission est la bande de fréquence dans laquelle les signaux sont soumis à un affaiblissement inférieur ou égal à n db.

- Ps > 0.5 x Pe ➔ n = 3 db

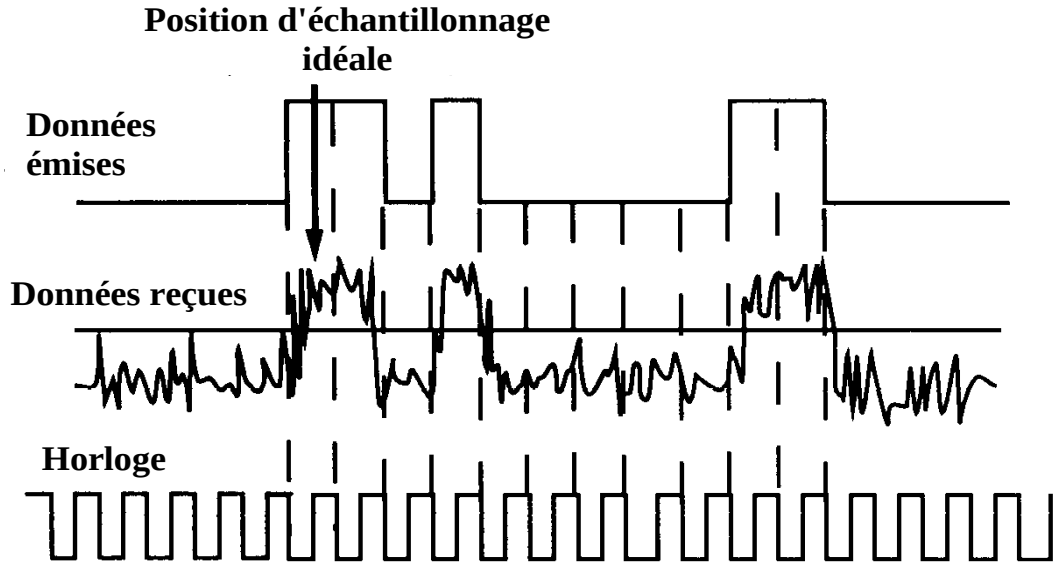


- **Capacité :** quantité d'information que peut transmettre un canal par unité de temps (débit binaire max.)

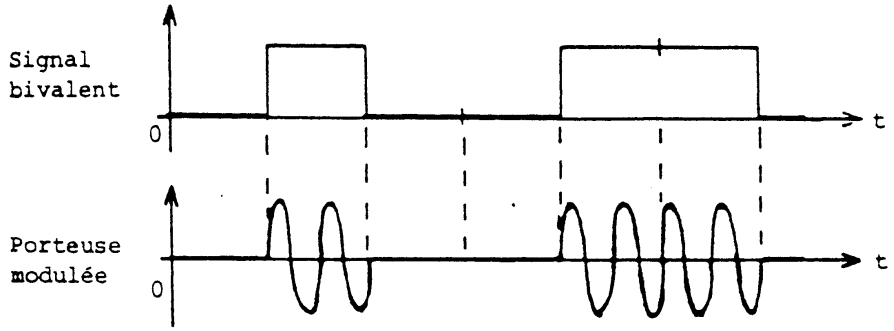
$$C = W \log_2 (1 + S/N) \text{ en bits/s}$$

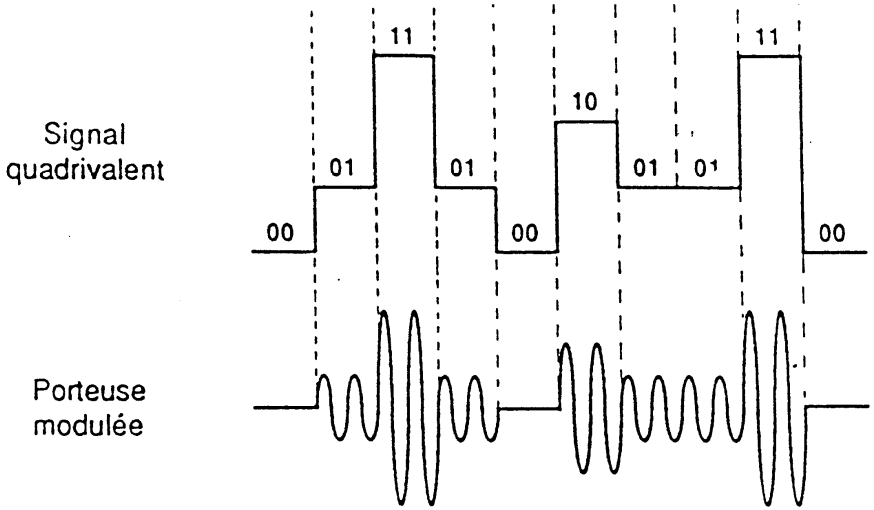
Voies de transmission et codage

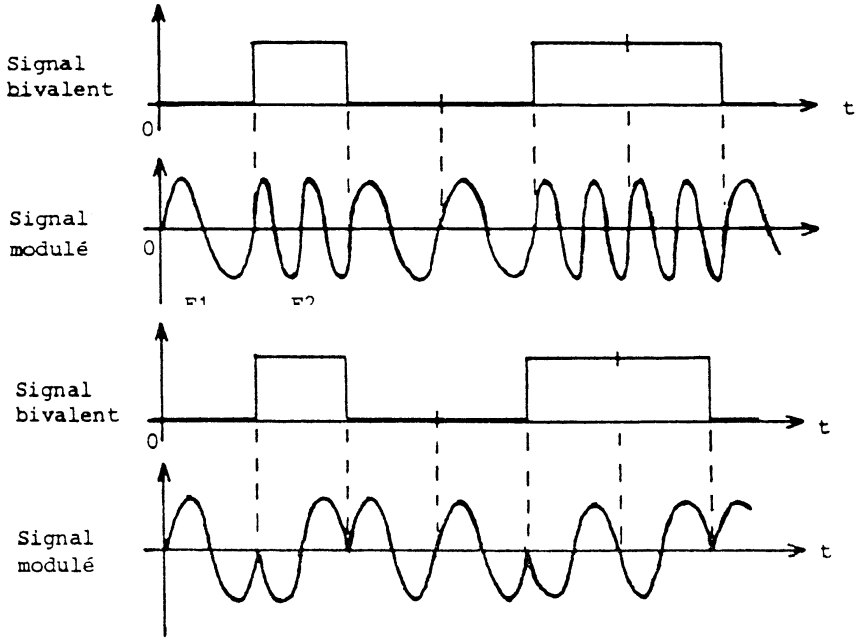
- **Intervalle significatif** Δ : intervalle entre 2 instants significatifs
- **Valence** V : nombre d'états significatifs distincts pour caractériser les états du signal à transmettre
- **Rapidité de modulation :** $R = 1 / \Delta$ Bauds (Δ en s)
- **Débit binaire** : quantité d'information émise, $D = R.\log_2 V$ (bit/s)

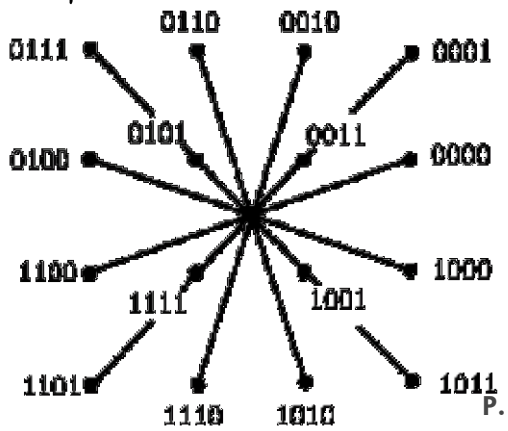


Codage par modulation

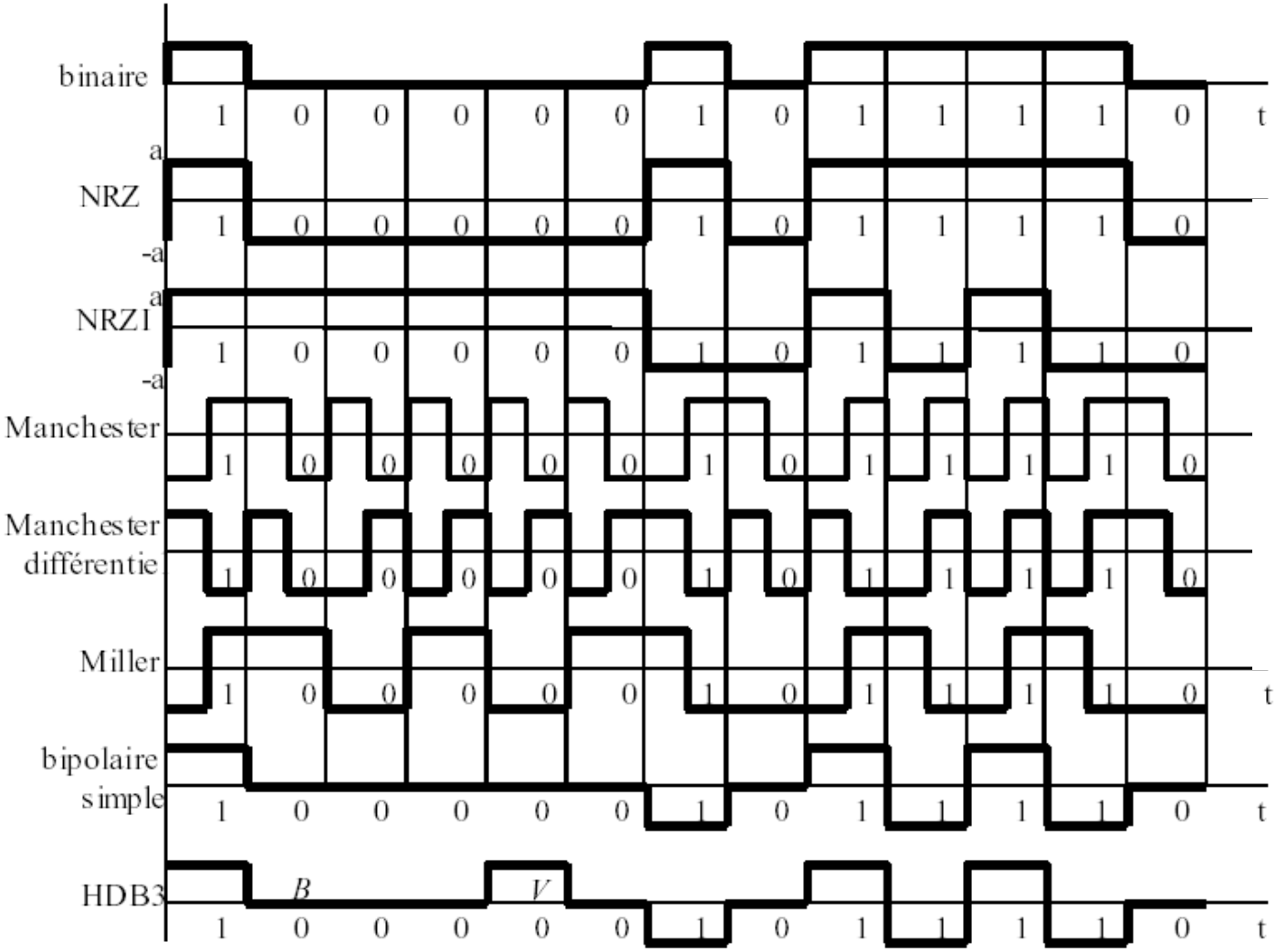


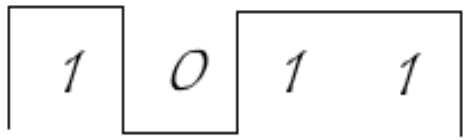


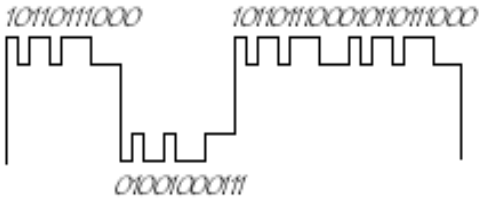




Codage en bande de base



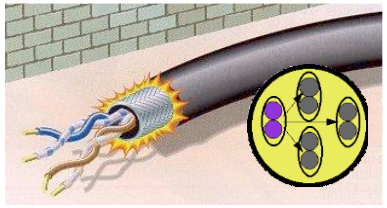


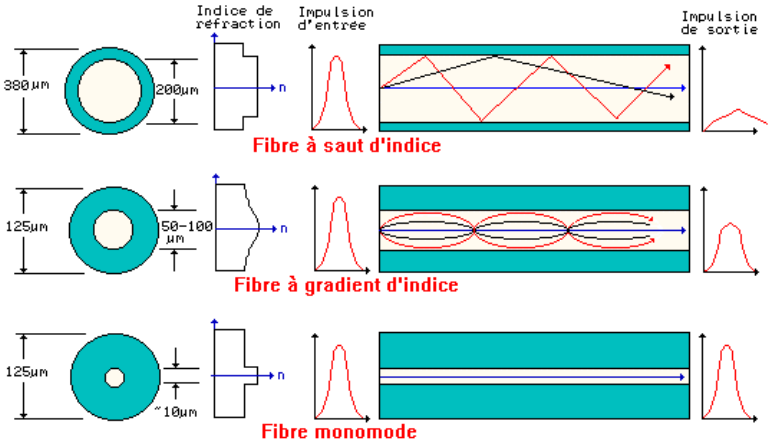


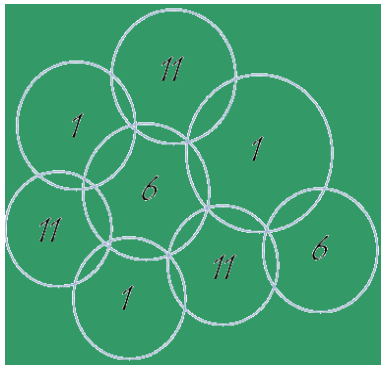
Direct Sequence Spread Spectrum

Le support physique ou Médium

- **Différents types :**
  - Paires torsadées
  - Coaxial
  - Fibre optique
  - Radio
  - . . .







- **Critère de choix :**
  - Coût
  - Extensibilité
  - Fiabilité
    - Immunité électromagnétique
    - Résistance mécanique
    - Souplesse
    - Résistance thermique
    - Facilité de localisation des coupures

Débit/Distance/Coût

SOUS COUCHE MAC

CSMA/CD - CSMA/CA (DCF, PCF) - Jeton
Adresse MAC - Contrôle d'erreur

Plan

- Méthodes d'accès au support
  - A contention : CSMA/CD (Ethernet)
  - CSMA/CA (WiFi)
    - A contention : DCF (aléatoire)
    - A réservation centralisée : PCF (polling)
  - A réservation décentralisée
    - Anneau à jeton (Token Ring - FDDI)
- Adresse MAC
- Contrôle d'erreur de transmission

CSMA/CD (Ethernet)

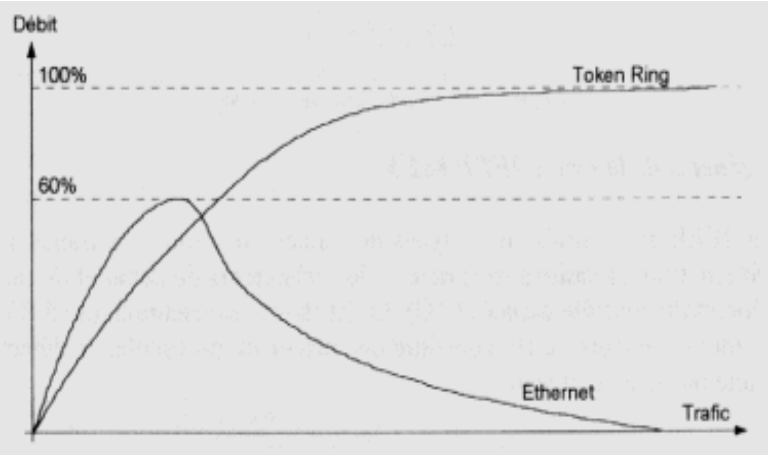
- Bus → collisions !!!
- Résoudre ce problème : CSMA/CD
- Multiple Access, Carrier Sense (écoute)
  - Une station n'émet sa trame que si le support est libre
  - « J'envoie sur le réseau et j'espère que ça passe »
  - Ce système n'évite pas totalement les collisions…
  - … car le support physique possède un temps de propagation !
  - Exemple
    - Paire torsadée temps de propagation = 5 ns/m, longueur du bus = 100 m
    - Stations A et B situées aux extrémités du bus
    - T0 : A commence à émettre sa trame
    - T0 + 400 ns : B commence à émettre sa trame
    - → Collision !

CSMA/CD (Ethernet)

- Collision Detection
  - Une station qui émet **écoute en même temps**
  - « Ce que j'entends doit être équivalent à ce que j'émets, sinon c'est qu'il y a une collision »
- Résolution de la collision
  - Réémission après une attente **aléatoire**
  - Algorithme du « BEB backoff »
    - i = numéro de la tentative d'émission
    - Tirage aléatoire du temps d'attente T
    - $0 < T < 2^i$

Conclusion CSMA/CD (Ethernet)

- Performances CSMA/CD
  - Se dégradent très vite si le nombre de stations augmente
- Avantages
  - Algorithme simple à implémenter
  - Accès équitable au support
- Inconvénients
  - Non déterministe
- Aujourd'hui : commutateurs Ethernet = zéro collision
  - Bufferisation des trames (toujours non déterministe)
  - Point à point full duplex

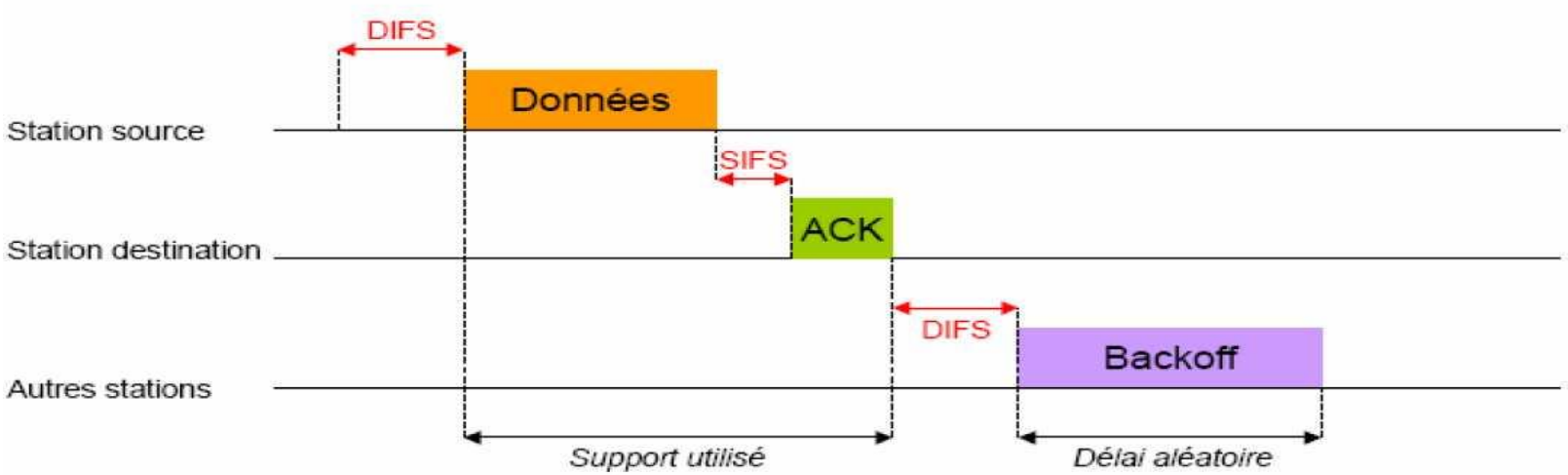


CSMA/CA DCF (WiFi)

- Radio = modulation d'une porteuse
  - mais multiplexage (FDMA, TDMA, CDMA) impossible car tout le monde doit pouvoir entendre tout le monde
- DCF = Distributed Coordination Function
  - La plus utilisée, adaptée pour les données asynchrones
  - Les utilisateurs ont une chance égale d'accéder au support
- Carrier Sense
  - Écoute du support avant émission
- Collision Avoidance
  - En radio il n'est pas pertinent d'écouter ce qu'on émet pour détecter une collision
  - Problème de la « station cachée »
- Acquittements
  - Récepteur envoie un acquittement
  - Emetteur attend acquittement
  - Si émetteur ne reçoit pas d'acquittement alors ***collision probable***

CSMA/CA DCF (WiFi)

- Constantes temporelles
  - Timeslot = 10µs, SIFS = 10µs, DIFS = 30 µs
- Pour chaque tentative de retransmission i, le temporisateur de backoff croît de la façon suivante
  - $[2^{2+i}$ x randf()] x Timeslot
- Echange type :

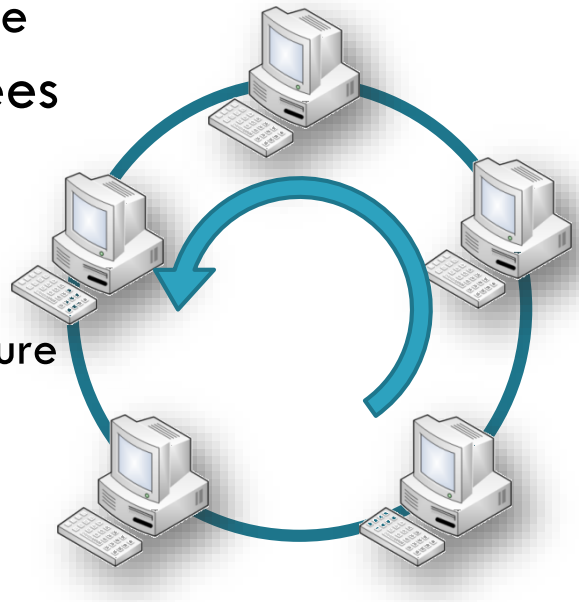


CSMA/CA PCF (WiFi)

- PCF = Point Coordination Function
  - Accès géré par un « chef d'orchestre »
    - Point d'accès WiFi
  - Interrogation à tour de rôle des stations
    - « Polling »
    - Schéma « Maître – Esclave »
  - Données synchrones (voix, vidéo)
  - Déterministe !
  - Rarement implémenté dans les points d'accès et les cartes WiFi…

Token Ring (IBM, 1981)

- Topologie en anneau, tous les équipements sont égaux
- Algorithme d'accès au support
  - Jeton = trame sans données
  - Une station qui reçoit le jeton
    - Le réémet tel quel si elle n'a pas de données à émettre
    - Ou bien émet sa trame de données à la place
  - Une station qui reçoit une trame de données
    - Regarde si elle est émettrice
      - Si oui la retire, et émet le jeton
    - Regarde si elle est destinataire
      - Si oui la transmet à la couche réseau supérieure
      - Emet la trame reçue



Token Ring (suite)

- Trames Token-ring

Jeton vide

Marqueur Début	Access CTRL (J = 0)	Marqueur fin
1 octet	1	1

Trame

Marqueur Début	Access CTRL (J = 1)	Frame Control	Adresse destination	Adresse source	Données	CRC	Marqueur Fin	Status
1 octet	1	1	2 ou 6	2 ou 6	quelconque	4	1	1

5000 max.

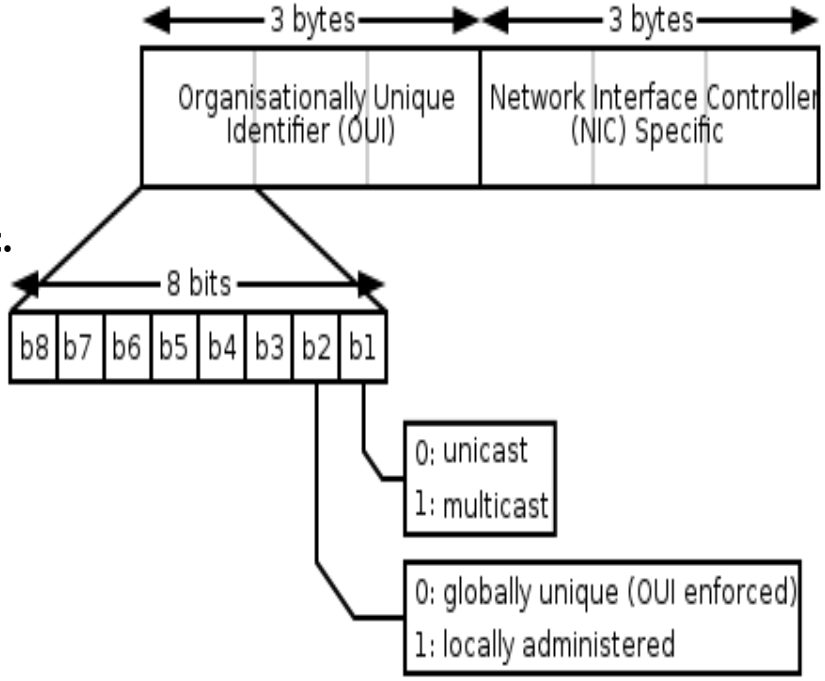
- Avantages
  - Déterministe
  - Délai d'attente de transmission garanti
- Inconvénients
  - Une seule panne → l'anneau est arrêté !
  - Equipements onéreux

FDDI (1986)

- Fiber Distributed Data Interface
- 100 Mb/s sur fibre optique
- Topologie en anneau
  - Double anneau pour la sécurité
- Gestion par jeton (cf Token Ring)
- Assez utilisé pour les « backbone »

Adresse MAC

- Désigne de manière unique une station dans le réseau
  - Ethernet, WiFi, etc. : adresse fixée en usine par le fabricant de la carte réseau
  - Adressage « à plat » : adresse unique au monde mais impossible de déterminer la position géographique de la station
    - Routage MAC inter-réseaux impossible !
- Format normalisé IEEE
  - Objectif : garantir l'unicité des adresses
  - Longueur standard = 48 bits (6 octets)
  - Ethernet, WiFi, Bluetooth, Token-Ring, etc.
  - ATM : 6 derniers octets des 20 octets d'une adresse ATM
- Types d'adresses MAC
  - Unicast
  - Broadcast
  - Multicast



Contrôle d'erreur

- La couche MAC réalise un contrôle d'intégrité
- A l'émission
  - CRC32 ajouté en fin de trame
- En réception
  - Rejet des trames erronées
  - Pas de reprise sur erreur

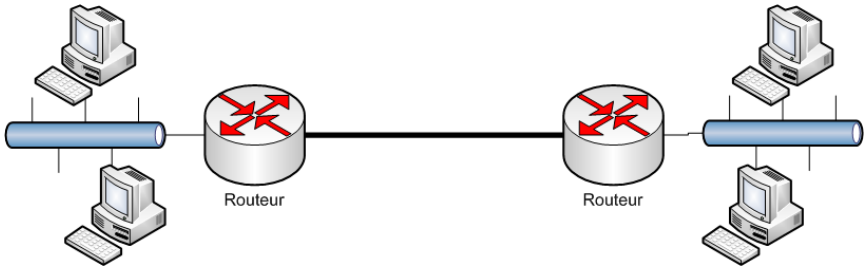
RESEAUX A COMMUTATION

Commutation de circuits

Commutation de paquets

Introduction

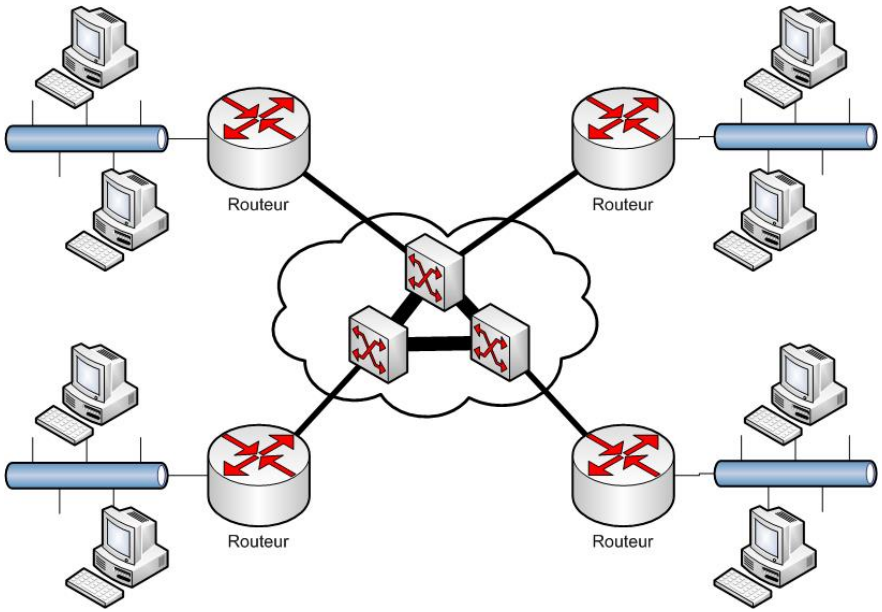
- Interconnexion de réseaux locaux
  - Objectif
    - Pouvoir mettre en relation une machine d'un réseau local avec n'importe quelle autre machine d'un autre réseau local
  - 2 réseaux à interconnecter : tout va bien !
    - 1 liaison point à point, Multiplexage traditionnel



  - N réseaux à interconnecter : problème !
    - Il faut **Nx(N – 1)/2** liaisons point à point !

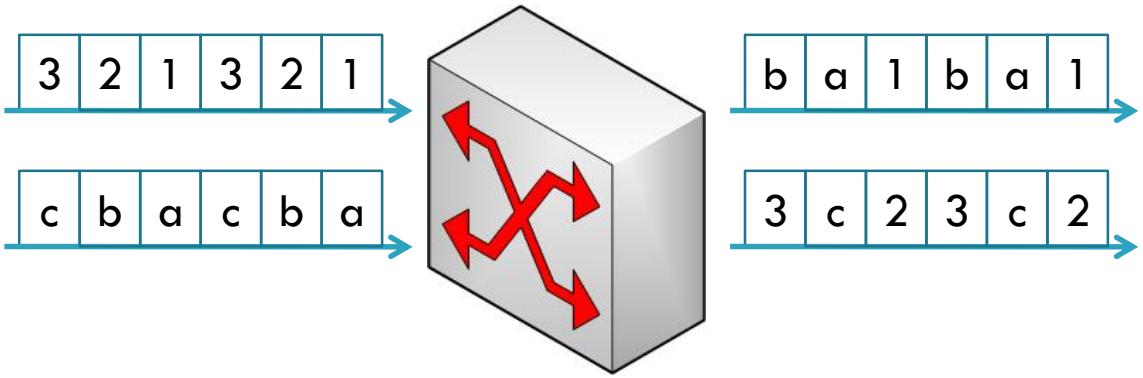
Commutation

- Réseau à commutation
  - Implique le routage !
- Commutateurs
  - De circuits
  - De messages
    - Message entier
    - Paquets
      - « Morceaux » du message
      - Environ 1000 octets
      - Taille variable
    - Cellules
      - Tout petits paquets
      - Taille fixe
- Table de commutation
  - Interne à chaque commutateur



Commutation de circuits

- Etablissement d'un lien physique entre source et destination
  - Exemple : RTC ancien
  - C'est une liaison point à point temporaire
  - Toute la ressource est monopolisée durant la connexion, même si elle est sous-utilisée
  - Technique moderne = commutation temporelle
    - Cœur numérique des réseaux téléphoniques actuels
    - Commutation par intervalle de temps

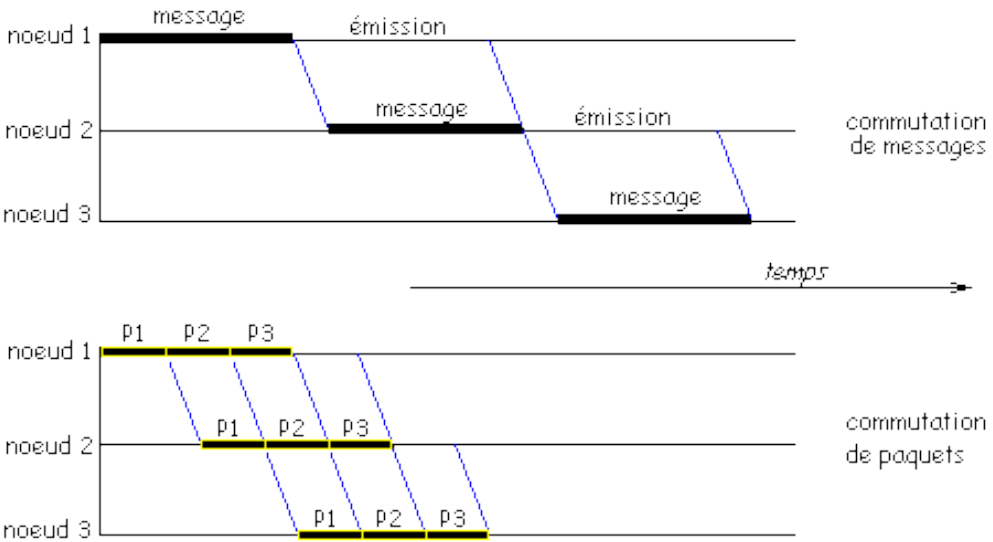


Commutation de messages, de paquets

- Commutation de messages
  - Le message à transmettre est transmis de proche en proche jusqu'au destinataire
  - Il n'est réémis qu'après avoir été entièrement reçu
  - Pendant sa transmission, le message occupe la totalité de la ressource (liaison)
  - Une fois le message transmis, la ressource est libérée
- Commutation de paquets
  - Découpage du message en petits paquets
  - Transmission des paquets selon la technique ci-dessus

Commutation de messages, de paquets

- Avantage de la commutation de paquets
  - Temps de transmission total
- Inconvénient
  - Séquencement non garanti si l'itinéraire change entre 2 paquets du même message

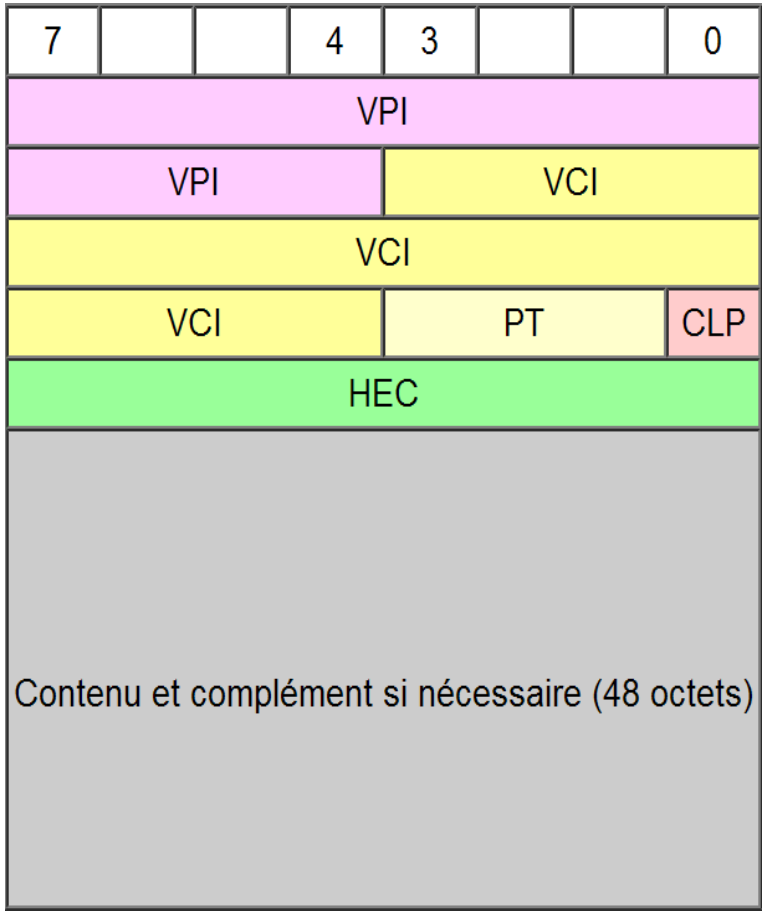


Commutation de paquet : 2 modes

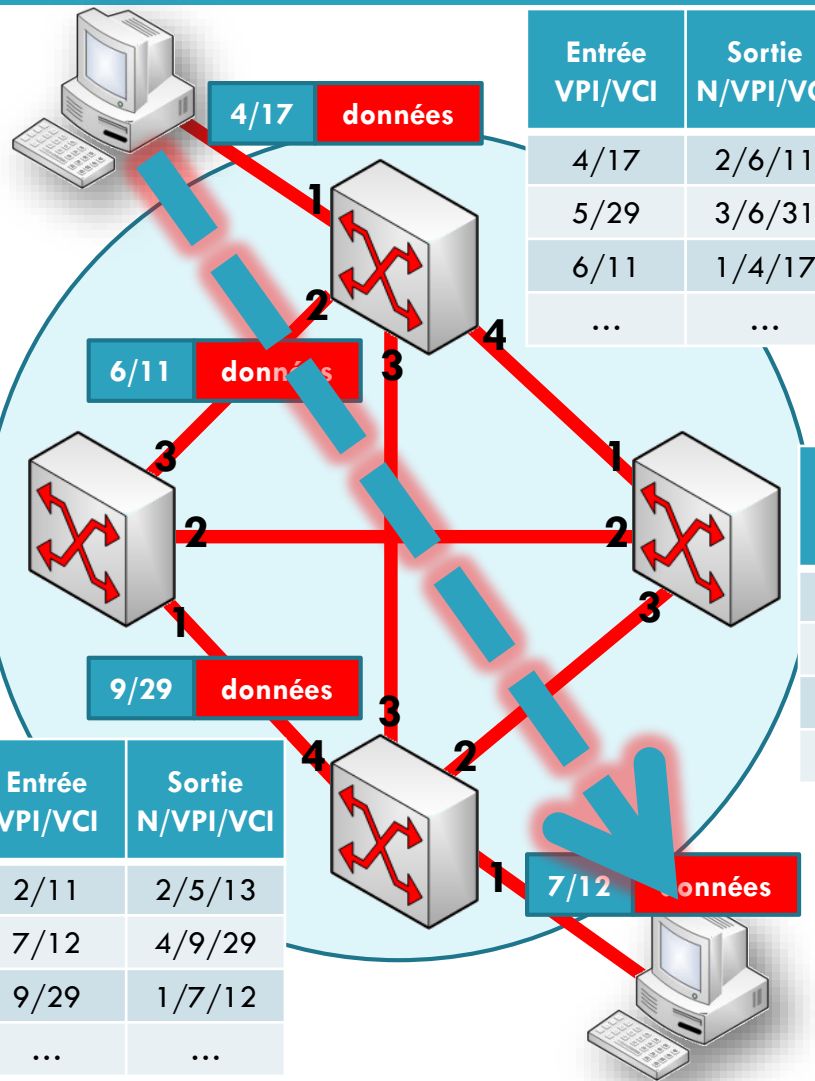
- Mode circuit virtuel
  - Tous les paquets d'un message suivent le même chemin
  - Séquencement garanti
  - Nécessité d'une signalisation préalable pour « préparer » les nœuds intermédiaires
- Mode datagramme
  - Pour chaque paquet, le nœud source choisit le nœud destination sans tenir compte des paquets précédents
  - Cette décision peut être différente d'un paquet à l'autre
    - Avantage ?
      - Routage adaptatif selon la charge du réseau
    - Inconvénient ?
      - Table de commutation très longue
      - Prise de décision difficile
      - Chaque paquet doit transporter des informations pour le routage (adresses src/dst)

ATM : commutation de cellules

- Cellule
  - 5 octets de contrôle
    - Information de routage sur 28 bits
    - Contrôle d'erreur
  - 48 octets de données



ATM : exemple



Entrée VPI/VCI	Sortie N/VPI/VCI
4/17	2/6/11
5/29	3/6/31
6/11	1/4/17
…	…

Entrée VPI/VCI	Sortie N/VPI/VCI
3/19	2/14/22
6/11	1/9/29
9/29	3/6/11
…	…

Entrée VPI/VCI	Sortie N/VPI/VCI
3/16	2/11/12
11/36	3/5/9
18/25	2/5/7
…	…

Entrée VPI/VCI	Sortie N/VPI/VCI
2/11	2/5/13
7/12	4/9/29
9/29	1/7/12
…	…

PROTOCOLE LAPB

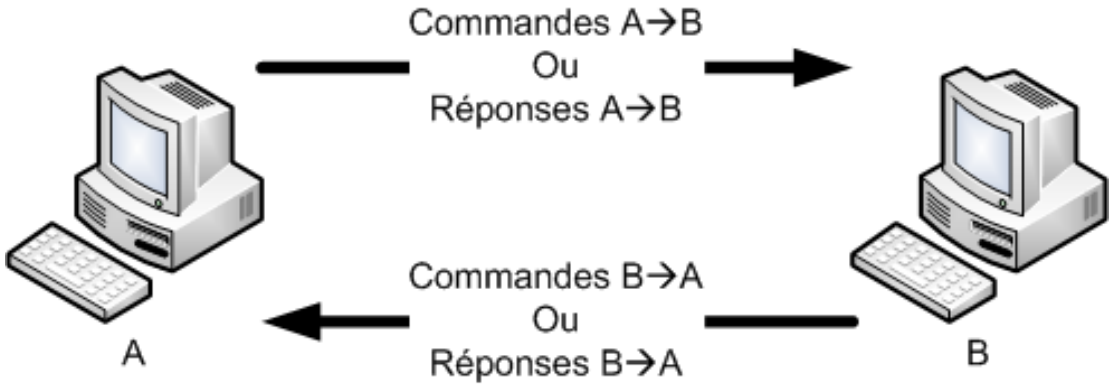
Mécanismes fondamentaux

Plan

- LAPB
  - Messages possibles (« PDU »)
    - Messages de données
    - Message de contrôle
  - Règles d'échanges des messages
    - Mécanismes communs à beaucoup de protocoles
    - Calcul d'efficacité
    - Automate d'états finis

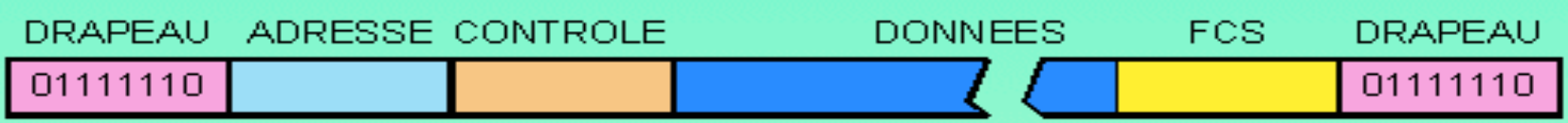
LAPB

- Link Access Protocol Balanced
- Dérivé proche de HDLC → OSI n°2 (LLC classe 2)
  - Connexion, déconnexion
  - Acquittements
  - Séquencement
  - Contrôle de flux
  - Contrôle d'erreur
- Exploite une liaison point à point full-duplex
- Pas de hiérarchie entre les 2 communicants



LAPB : les PDU

LAP B



CHAMP CONTROLE									
TYPE DE TRAMES		Rang du bit (ordre d'envoi) 1	2	3	4	5	6	7	8
I	Information	0	N(S)			P	N(R)		
S	Supervision	1	0	S	S	P/F	N(R)		
U	Non numérotées	1	1	M	M	P/F	M	M	M

N(S) N(R) : Numéro de séquence d'émission (S) ou de réception (R)

P/F : Invitation à transmettre (P/F=1) dans une commande.
Fin de transmission (P/F=1) dans une réponse.

M : Element binaire d'une fonction de modification

S : Element binaire d'une fonction de supervision

P : Invitation à émettre (P=1)

LAPB : les PDU (suite)

- Drapeau 8 bits « 01111110 »
  - Indique le début et la fin de trame dans le train binaire
  - Et si les données contiennent « 01111110 » ???
    - Bit stuffing → assure la **transparence** du protocole vis-à-vis des données transportées
    - Algorithme
      - Emission
        - Insertion systématique de « 0 » après « 11111 »
      - Réception
        - Suppression systématique du « 0 » suivant « 11111 »
- Adresse 8 bits
  - Indique le destinataire
  - En mode point à point : quasiment inutilisé
    - Exemple Transpac : 0x03=Station, 0x01=Commutateur Transpac

LAPB : les PDU (suite)

- FCS
  - 16 bits, CRC16 → cours « cryptographie »
- Compteurs N(S) et N(R)
  - Chaque équipement maintient de son côté 2 compteurs
    - Numérotation des trames émises
    - Numérotation des trames reçues
  - Dans les PDU émis, on retrouve ces compteurs
    - N(S) = numéro de la trame émise
    - N(R) = numéro de la prochaine trame attendue
      - Implique l'acquittement des trames N(R)-1, N(R)-2, etc.

LAPB : PDU de supervision (trames « S »)

4 fonctions implémentées par 4 PDU :

RR

RNR

REJ

SREJ

Nom	Description
RR (Receiver Ready)	Prêt à recevoir la trame n° N(R) •Acquittement simple •Reprise après RNR
RNR (Receiver Not Ready)	Non prêt à recevoir, mais bien reçu jusqu'à N(R)-1 •Saturation du récepteur (contrôle de flux)
REJ (Reject)	Les trames >= N(R) ont été rejetées (erreur de transmission)
SREJ (Selective Reject)	La trame N(R) a été rejetée

LAPB : PDU spéciaux (trames « U »)

Principales fonctions :

SABM

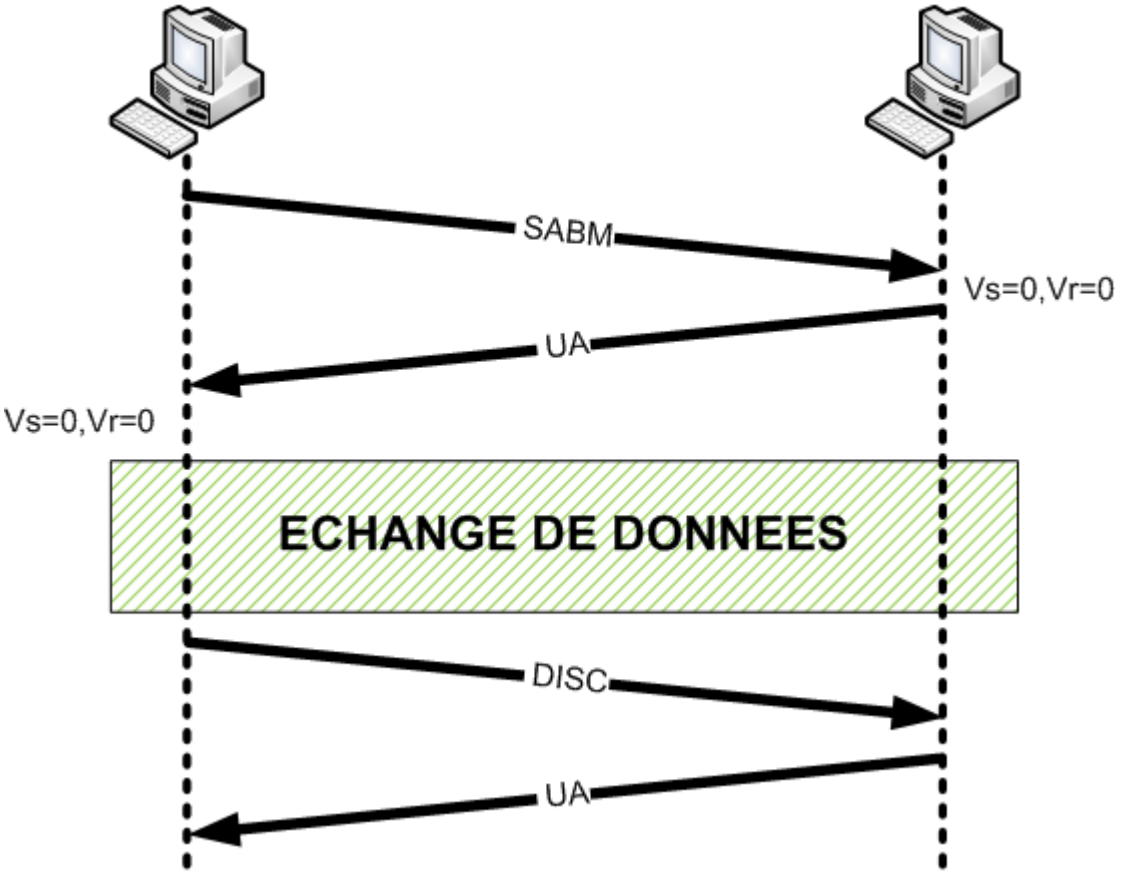
UA

DISC

Nom	Description
SABM (Set Asynchronous Balanced Mode)	Demande de connexion
UA (Unumbered Acknowledgment)	Acquittement des PDU spéciaux (SABM, DISC, etc.)
DISC (Disconnected)	Demande de déconnexion

LAPB : connexion et déconnexion

Initialisation des compteurs d'émission et réception

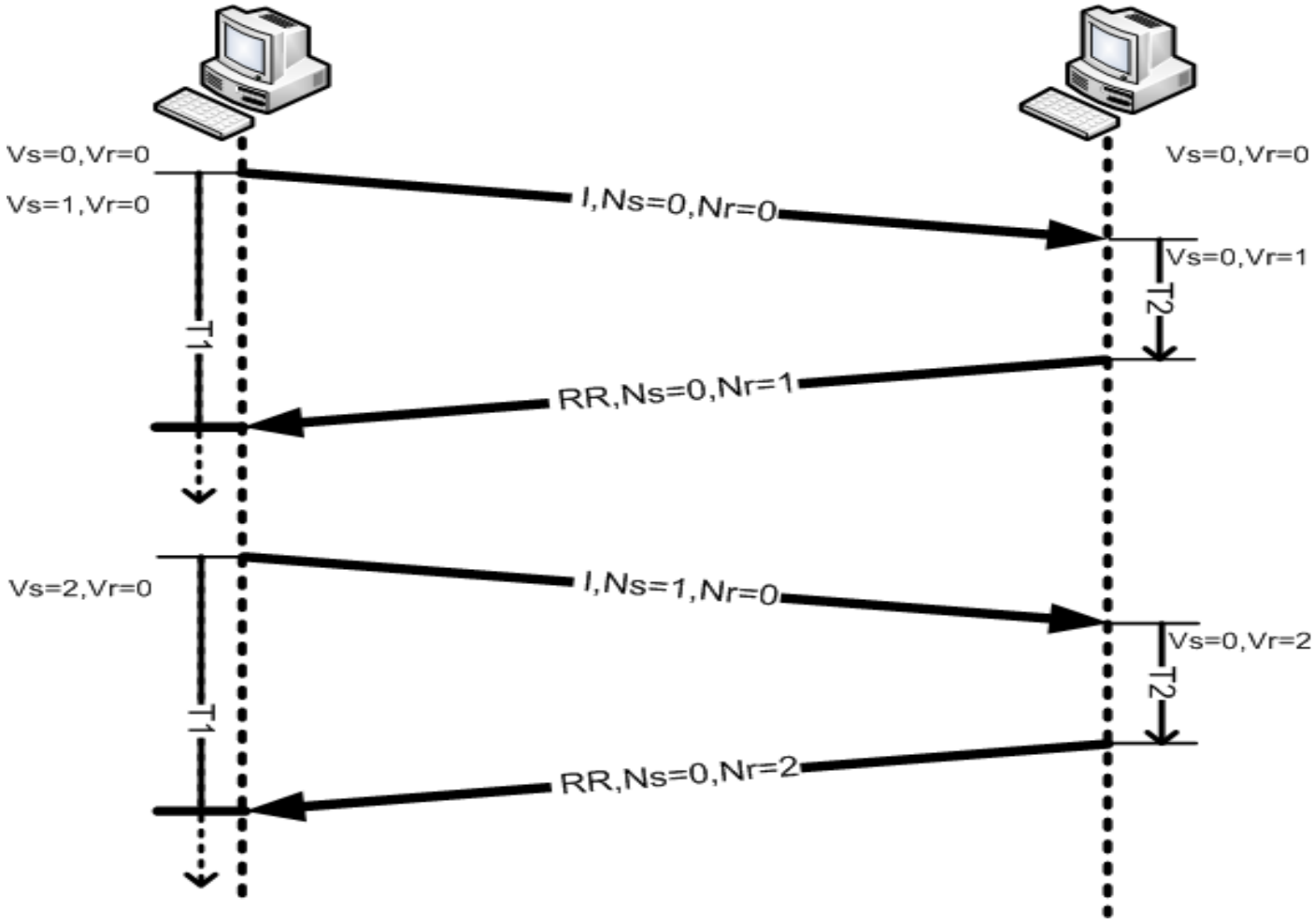


LAPB : échange de base

Timer T1 : un acquittement doit être reçu avant expiration !

Timer T2 : un acquittement doit être envoyé avant expiration, mais on attend éventuellement l'envoi de données pour acquitter en même temps.

Règle simpliste :
T2 < T1+D(INFO)+D(RR)



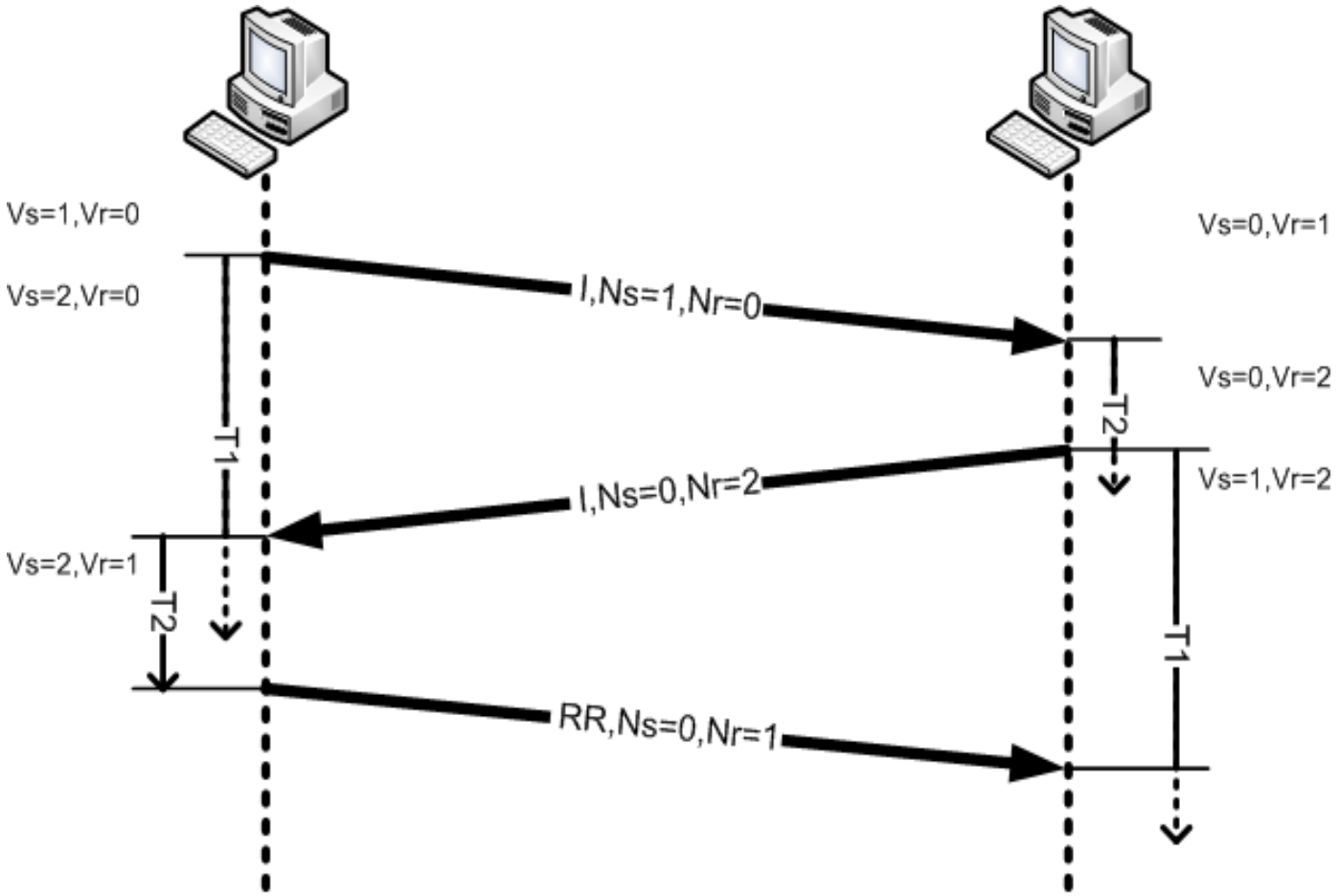
LAPB : échange de base (2)

Timer T1 : un acquittement doit être reçu avant expiration !

Timer T2 : un acquittement doit être envoyé avant expiration, mais on attend éventuellement l'envoi de données pour acquitter en même temps.

Règle améliorée :
T2 < T1+2.D(INFO)

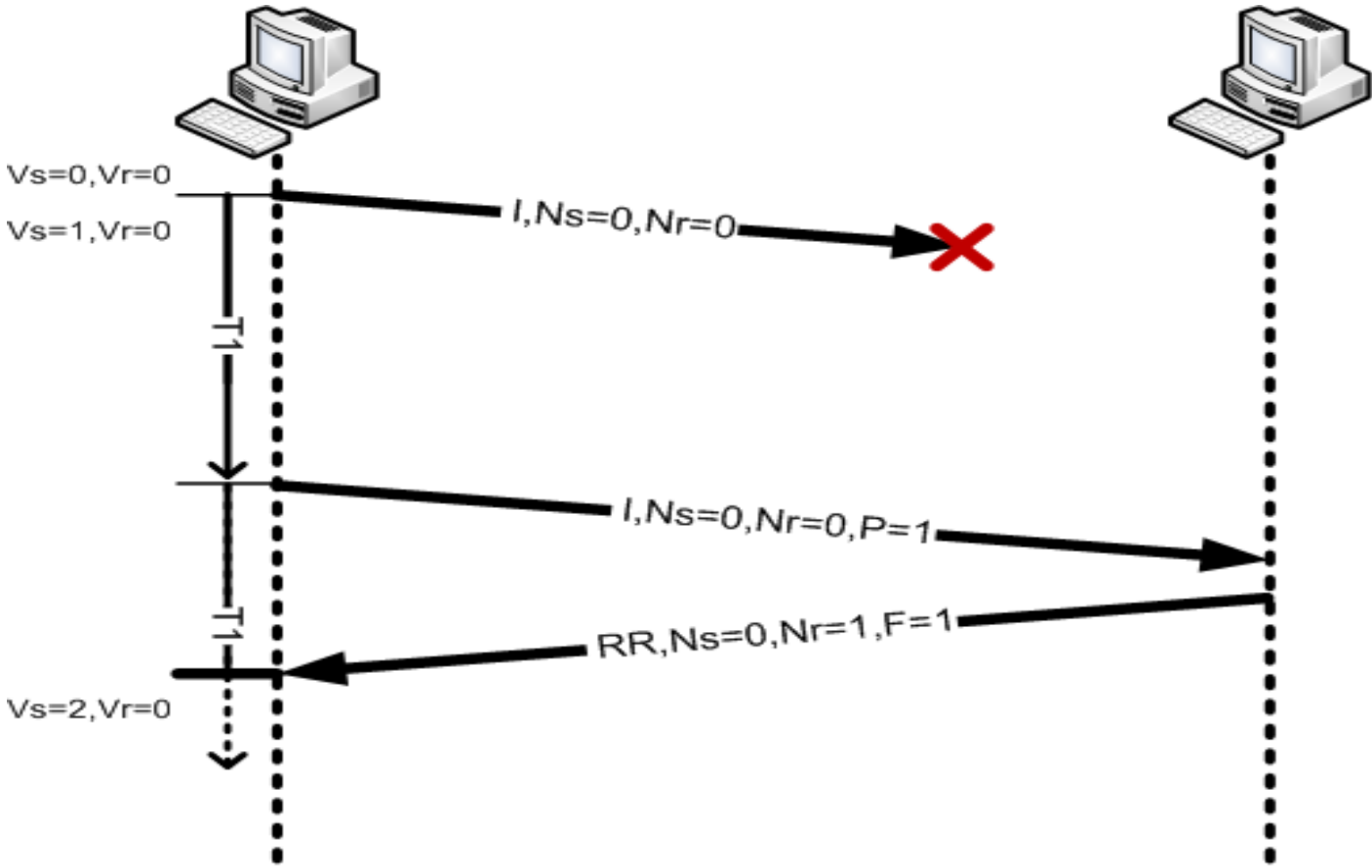
Etude du pire cas :
T2 > D(INFO)
T1 > 3.T2



LAPB : erreur de transmission

Timer T1 : un acquittement doit être reçu avant expiration !

Réémission si expiration T1, avec bit P=1 (acquittement immédiat demandé)



Echange de base : calcul d'efficacité

- Echange de base = « Stop and Wait »
- Calcul d'efficacité
  - Notations
    - D (bits) = taille des données, H (bits) = taille de l'en-tête (ex. LAPB : 6x8 = 48 bits)
    - A (bits) = taille de l'acquittement
    - Tp (s) = temps de propagation, B (b/s) = débit
  - Temps total réel pour une transaction (INFO + ACK) : Treel = (D+H)/B + A/B + 2.Tp
  - Temps minimal théorique pour l'envoi des données : Ttheorique = D/B
  - Efficacité = Ttheorique/Treel = D/(D+H+A+2.Tp.B)
    - Généralement H,A << D → Efficacité = 1 / (1 + 2a) avec a = B.Tp/D
    - D ↗ implique efficacité ↗ (mais la probabilité d'une trame erronée ↗ ☹)
    - Tp ↗ implique efficacité ↘ ☹ - B ↗ implique efficacité ↘ (mais les utilisateurs sont ☺)
- Conclusion : Stop and Wait est inefficace
  - Pour liaisons longues (ex : liaison par satellite, liaison ADSL)
  - Pour liaisons très haut débit
- Exemples
  - LAN (B=100Mb/s, D=10kb, Tp=500ns) : efficacité = 99% ☺
  - ADSL (B=4Mb/s, D=10kb, Tp=25µs) : efficacité = 1% ☹

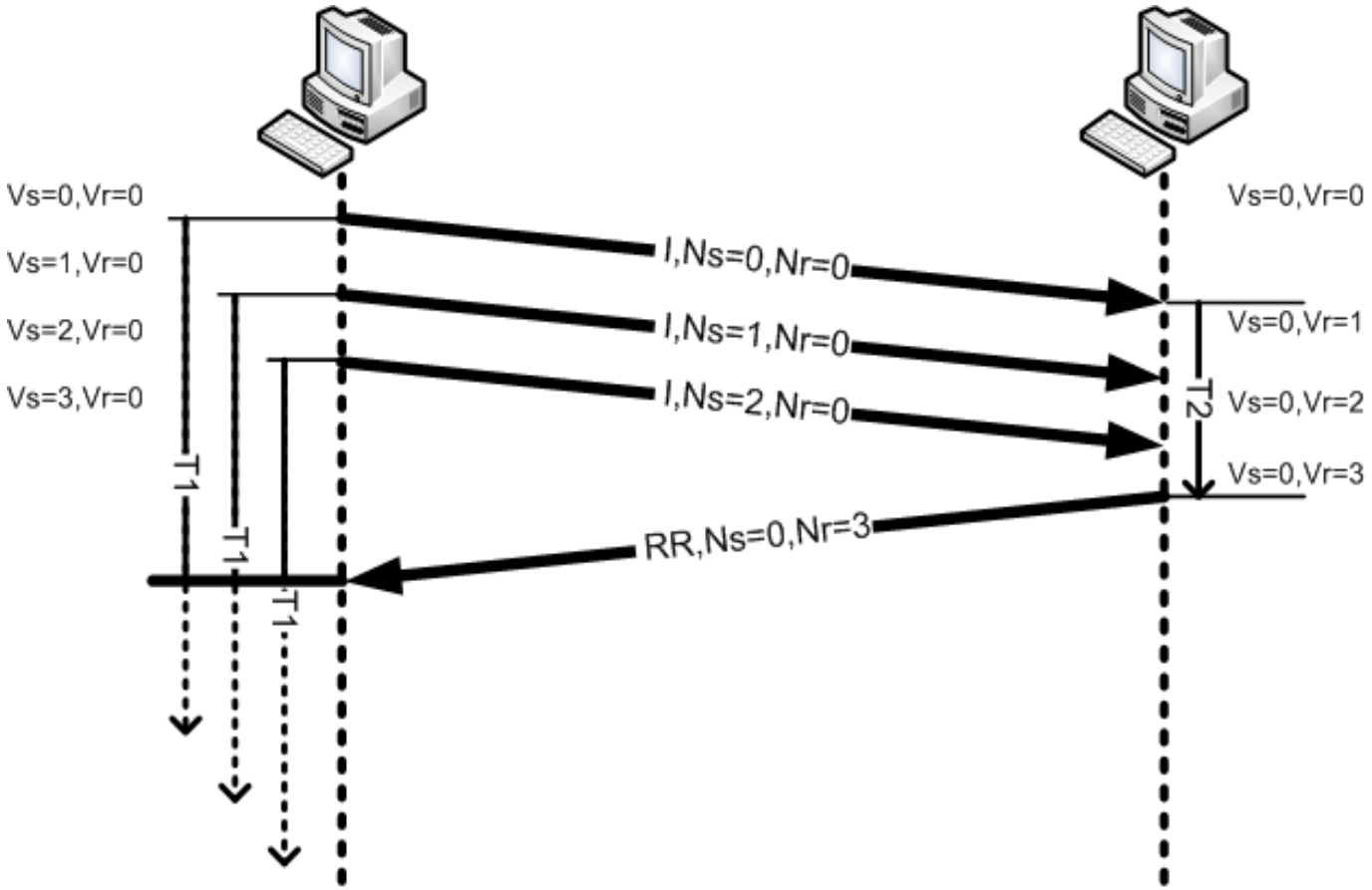
LAPB : fenêtre d'émission

Anticipation d'acquittement = plusieurs émissions sans attendre d'acquittement

K = fenêtre d'émission

LAPB :
Kmax = 7

Efficacité :
K / (1 + 2.a)



LAPB : fenêtre d'émission et erreurs

Anticipation d'acquittement = plusieurs émissions sans attendre d'acquittement

K = fenêtre d'émission

LAPB :

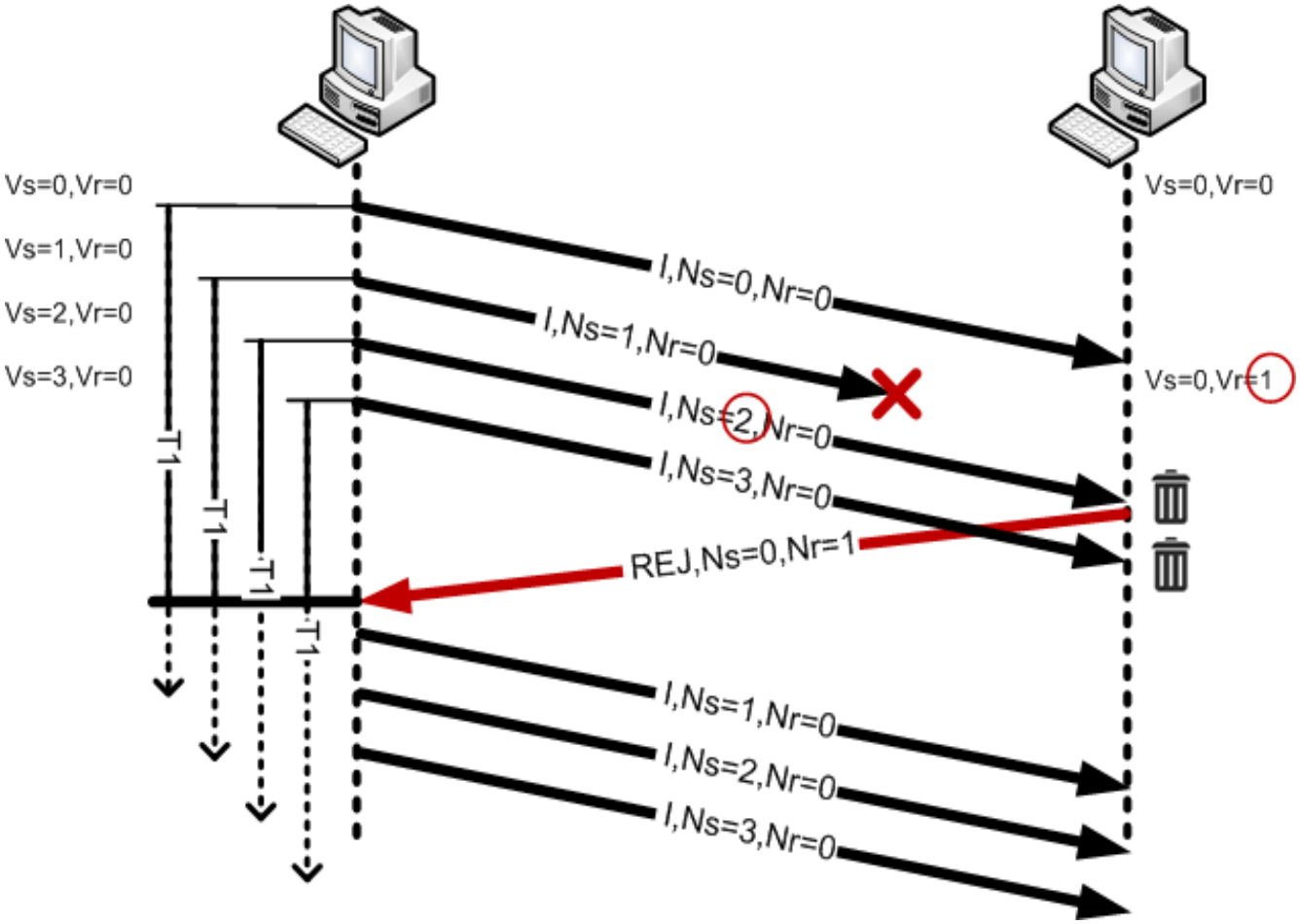
Kmax = 7

Bonne liaison :

K = 5..7

Mauvaise liaison :

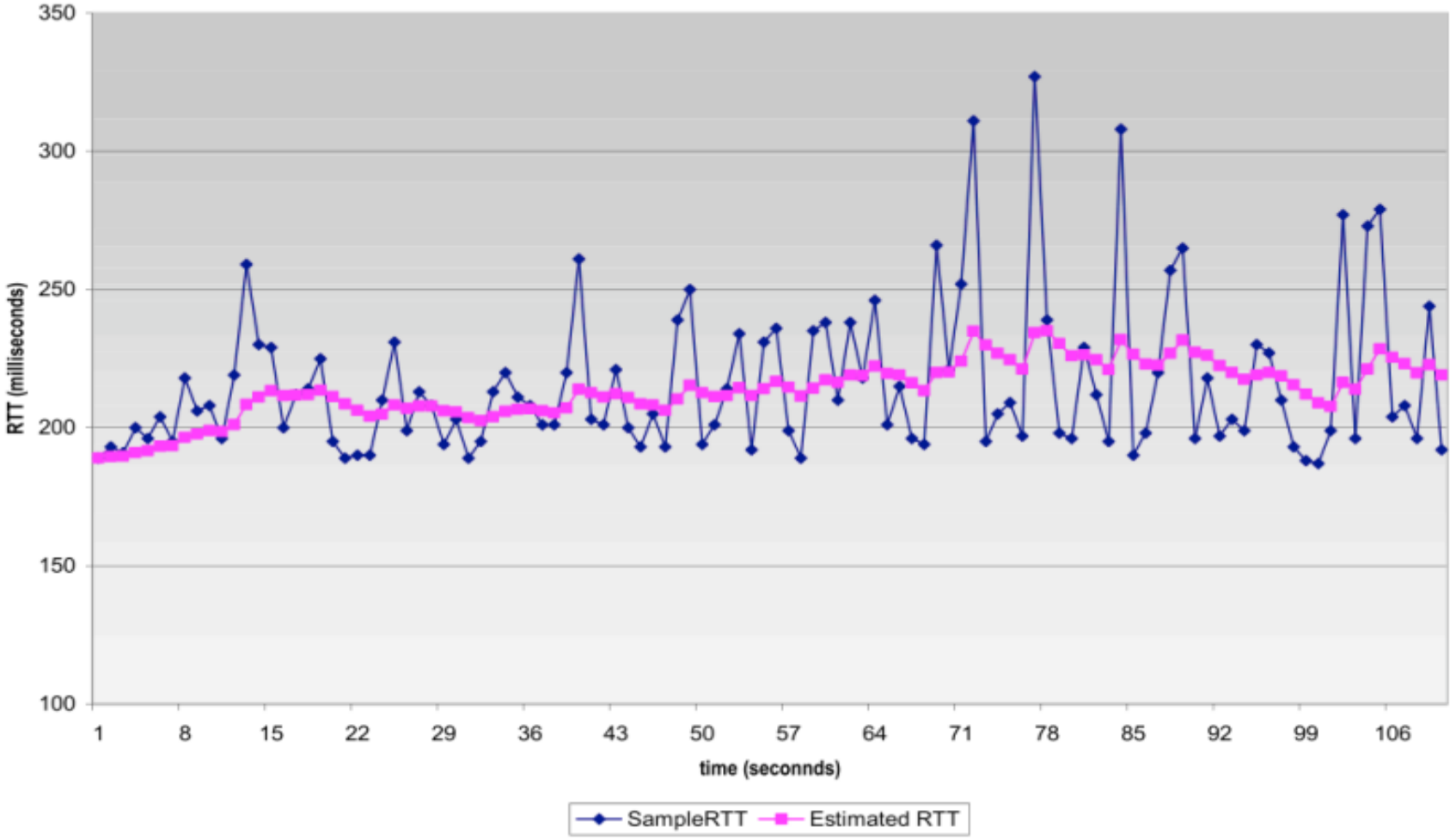
K = 1..4



Timers et fenêtre d'émission dans un réseau inconnu : TCP

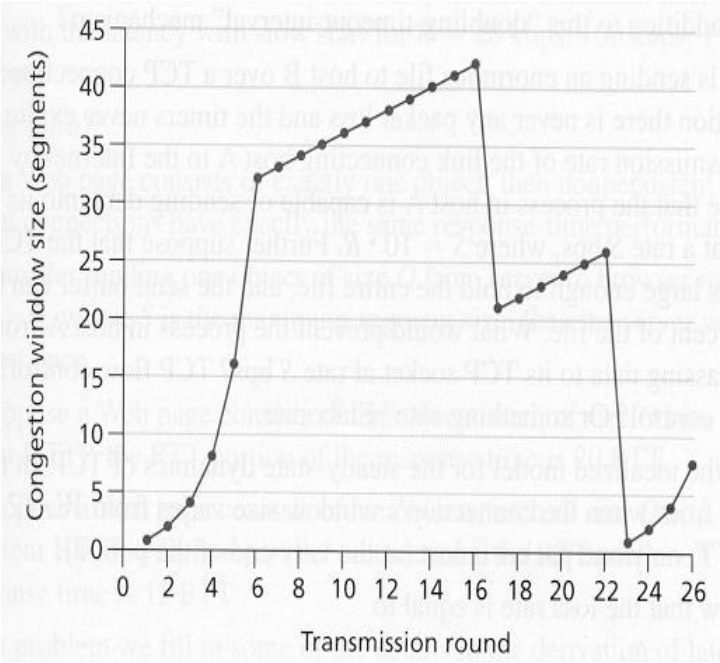
- Timer « T1 » = RTO (Retransmission Time Out)
  - RTO trop petit → réémissions inutiles
  - RTO trop grand → attentes inutiles avant retransmission
  - Ajustement dynamique du RTO
    - Estimation permanente du RTT (Round Time Trip) avec lissage
      - $RTT_{n+1} = RTT_n + r.(RTT_{mesuré} - RTT_n)$ avec ($0 \leq r \leq 1$, typiquement : $r = 0.125$)
    - Ajustement du RTO
      - $\Delta RTT_{n+1} = s.\ \Delta RTT_n + (1 - s).(RTT_{mesuré} - RTT_n)$ avec ($0 \leq s \leq 1$, typiquement : $s = 0.75$)
      - $RTO = RTT_{n+1} + 4.\Delta RTT_{n+1}$
    - →Adaptation **<u>en temps réel</u>** aux conditions du réseau !
    - « acknowledgment ambiguity »
      - Un ACK reçu correspond-il au paquet initial ou au paquet retransmis ?
      - Algorithme de Karn
        - Ne pas tenir compte d'un $RTT_{mesuré}$ suite à une retransmission
        - Si retransmission, $RTO = 2^n.RTO$ jusqu'à réception correcte ACK
        - Ensuite reprise de l'algorithme normal

Timers et fenêtre d'émission dans un réseau inconnu : TCP



Timers et fenêtre d'émission dans un réseau inconnu : TCP

- TCP = fenêtre d'émission (« congestion window » = NxMSS)
  - Trop grande → TCP engorge les nœuds intermédiaires avant de recevoir le moindre acquittement…
  - Trop petite → performances médiocres (réseaux « fat pipe »)
  - Solution : elle doit être dynamique !
    - « slow start »
      - **cwnd = 1, threshold = 65536**
      - A chaque ACK reçu, **cwnd = 2.cwnd**
      - Si cwnd > threshold
        - « congestion avoidance »)
      - Si paquet perdu → **threshold = threshold/2**
    - « congestion avoidance »
      - A chaque ACK reçu, **cwnd = cwnd + 1**
      - Si paquet perdu (expiration Timer)
        - « slow start »
      - Si paquet perdu (3 ACK identiques)
        - **cwnd = cwnd/2**



Contrôle de flux

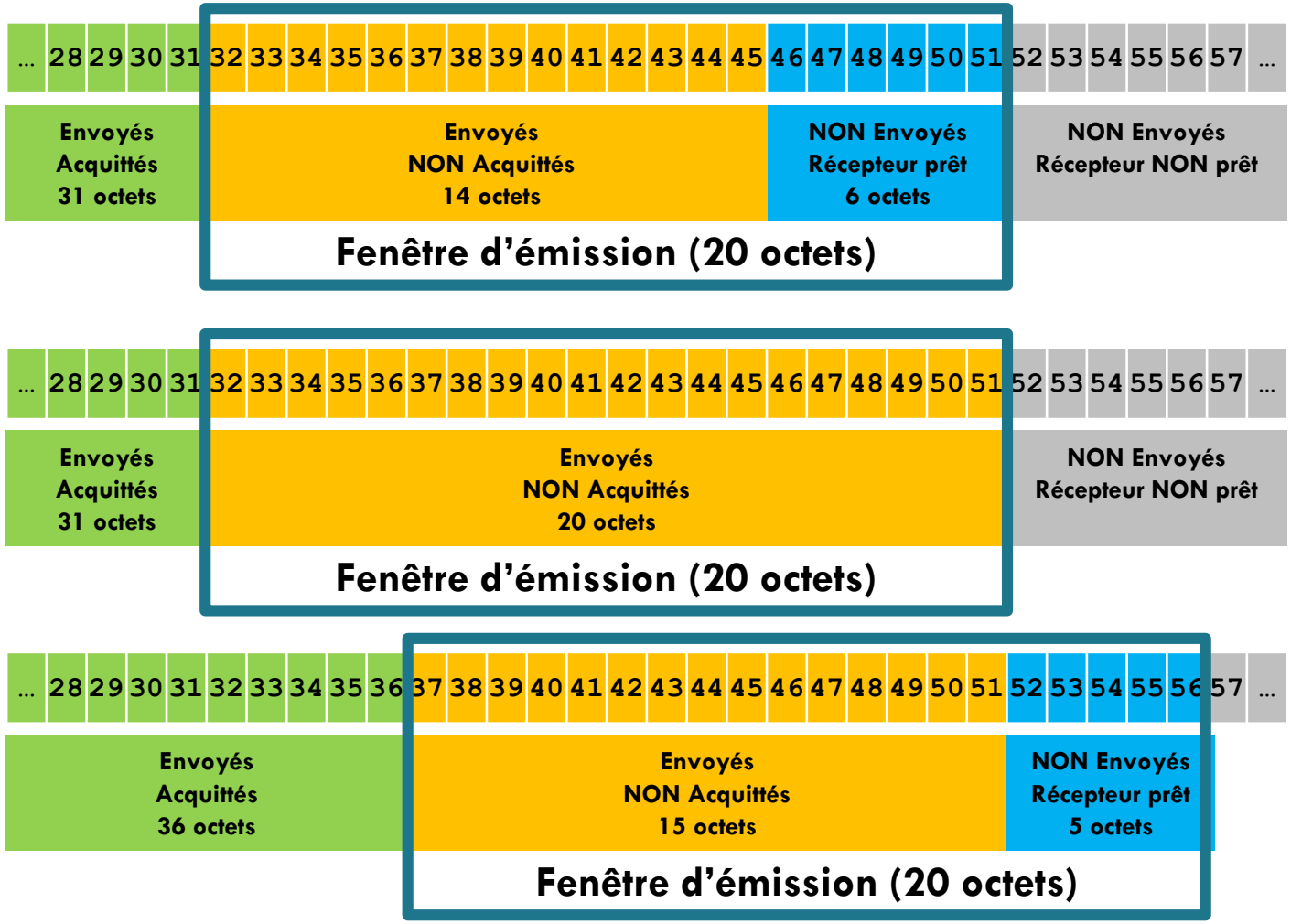
- LAPB : comportement binaire
  - « STOP » : RNR
  - « ENCORE » : RR
- TCP : comportement plus fin
  - Récepteur indique dans chaque ACK le nombre maximal d'octets qu'il peut recevoir : **rwnd**
  - L'émetteur peut donc envoyer
    - Min(**rwnd**, cwnd)
  - « Silly Window Syndrome »
    - Emetteur rapide, récepteur lent
    - Dès que le récepteur consomme quelques octets, il va indiquer dans ses ACK des rwnd très faibles
    - L'émetteur envoie aussitôt le maximum possible : rwnd
    - L'équilibre est trouvé mais transmission de nombreux tout petits paquets
    - Solution → grouper les données à l'émission (algorithme de Nagle)

Fenêtre glissante TCP : émission

Situation initiale

SEND(46,6)

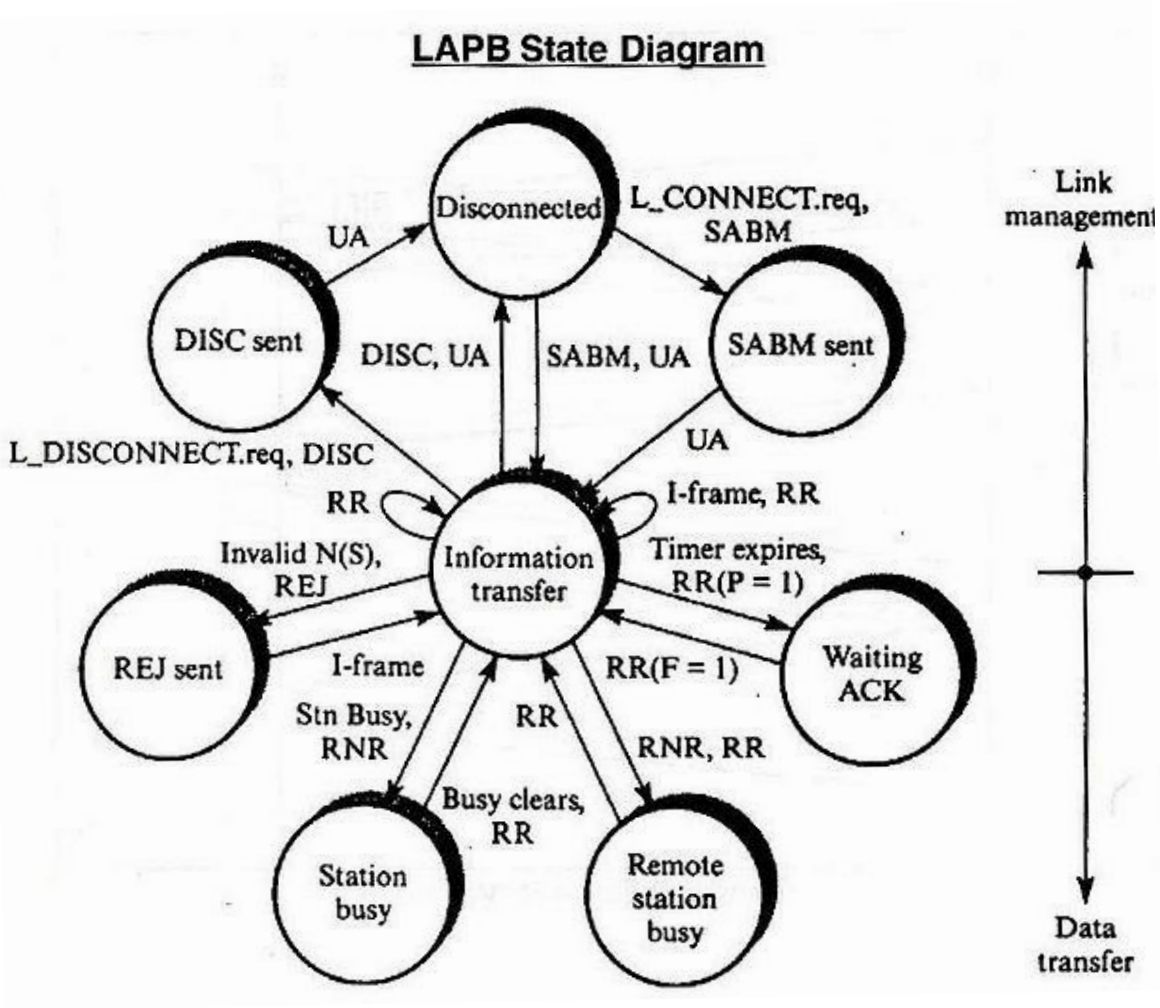
ACK(36)



LAPB : automate d'états

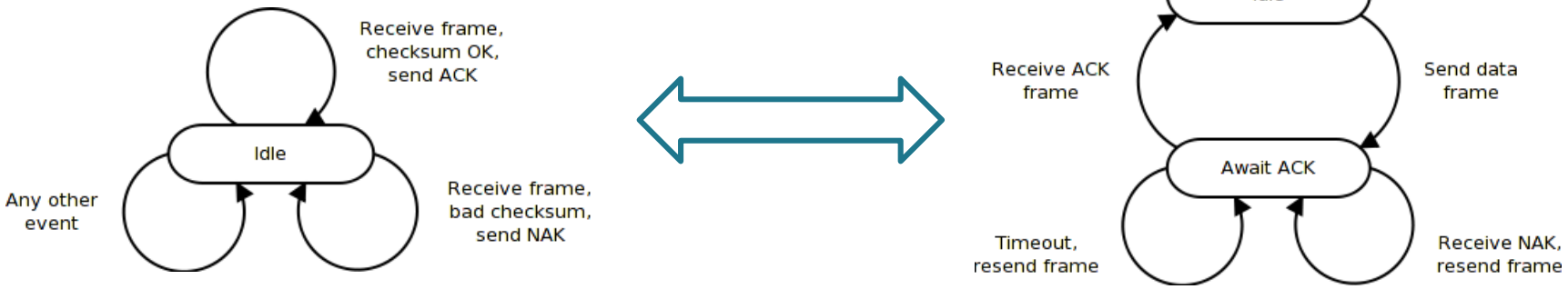
Notation pour les transitions :

« Evénement, Action »



Protocoles et automates d'états finis

- Machine à états finis = graphe orienté
- Sommet = état de l'entité communicante
  - Exemple « attente ACK »
- Arcs orientés = {événement, action à réaliser}
  - Exemple « réception REJ, réémission DATA »
- Intérêts
  - Spécifier de manière non ambiguë le comportement d'une entité communicante qui utilise les PDU du protocole modélisé
  - Pouvoir effectuer des preuves de la cohérence du protocole
    - Détection d'interblocages entre entités communicantes
    - Détection d'états non atteignables
  - Faciliter l'implémentation en langage procédural



ROUTAGE

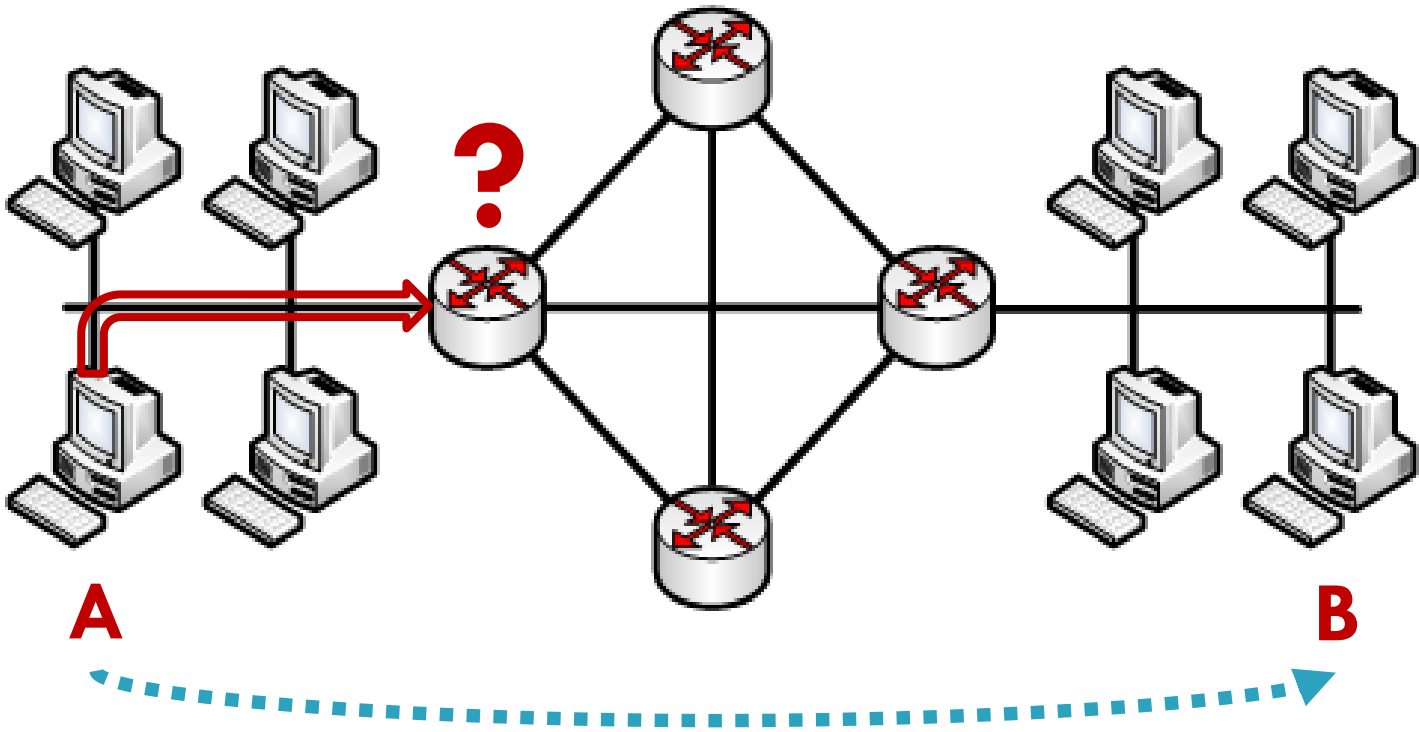
Principes - Algorithmes

Plan

- Rappels
- Algorithmes utilisés pour le routage
  - Bellman-Ford
  - Dijkstra
- Implémentations pour les réseaux
  - RIP
  - OSPF

Problématique du routage

Par où passer ?



Acheminer un paquet dans un réseau

- Routage
  - Décision d'acheminement prise en fonction d'une adresse **destination**
    - Mode datagramme : **pour chaque paquet**
    - Mode circuit virtuel : **paquet d'établissement**
  - Table d'acheminement = **table de routage**
    - Contient des groupes d'adresses, un coût et la direction à prendre
  - Nœud intermédiaire = **routeur**
  - Aucun contexte mémorisé → tolérance aux pannes
  - Mais prise de décision complexe pour chaque traversée de nœud
  - Protocole type : IP
- Commutation
  - Décision d'acheminement indépendante de l'adresse destination
  - Table d'acheminement = **table de commutation**
    - Contient des identifiants de flux et la direction à prendre
  - Nœud intermédiaire = **commutateur**
  - Protocoles types : Ethernet commuté, Frame Relay, ATM, MPLS

Différents modes de routage

- Statique
  - Remplissage **manuel** des tables de routage des nœuds
  - Simple à gérer, séquencement garanti
  - Pas de bouclage (si l'administrateur est compétent ! ☺)
  - Pas de solution de secours si rupture d'un lien ☹
  - Utilisation : réseaux avec peu de nœuds, sans redondance entre les routes
- Par diffusion
  - Aucune table de routage
  - Un nœud réémet le paquet reçu sur **toutes ses interfaces de sortie**
  - Surcharge du réseau et bouclage → « TimeToLive » dans chaque paquet
    - TTL = N au départ du paquet
    - TTL = TTL – 1 à chaque traversée de nœud
    - Si TTL = 0 Alors destruction du paquet
  - Très robuste, garantit de trouver le meilleur chemin
  - Mais engorgement du réseau et duplication inutile de paquets

Routage au moindre coût

- Mode de routage le plus utilisé
- Chaque nœud tient **à jour** une table indiquant quel est le « plus court » chemin (**coût minimal**) pour atteindre une adresse de destination
  - Adaptation en temps réel aux conditions de trafic
- Coût ou **métrique**
  - Nombre de sauts, distance réelle, temps de latence, délai de transmission, etc.
- Algorithmes pour calculer les coûts
  - Algorithmes basé sur les graphes (sommets = nœuds, arcs = liens réseau)
  - Peut-on utiliser les algorithmes classiques sur les graphes ?
    - Personne ne supervise le comportement des nœuds du réseau (pas d'ordinateur central qui calcule la solution optimale et envoie le résultat à tous les nœuds du réseau)
    - Généralement, personne ne connait la topologie exacte du réseau car cette topologie change à chaque instant (pannes, engorgements, etc.)
    - Au mieux, un nœud connaît ses voisins immédiats
    - → les algorithmes « classiques » sont donc inutilisables

Routage au moindre coût (2)

- Solution : algorithmes répartis
  - Algorithme réparti
    - Chaque nœud exécute le même « mini-algorithme »
  - Protocole d'échange d'informations entre nœuds
    - Convergence vers la solution optimale
- Deux grandes classes d'algorithmes répartis pour calculer les coûts
  - Algorithmes à vecteur de distance (ex: RIP)
  - Algorithmes à état de liens (ex: OSPF)

Algorithme de Bellman-Ford : principe

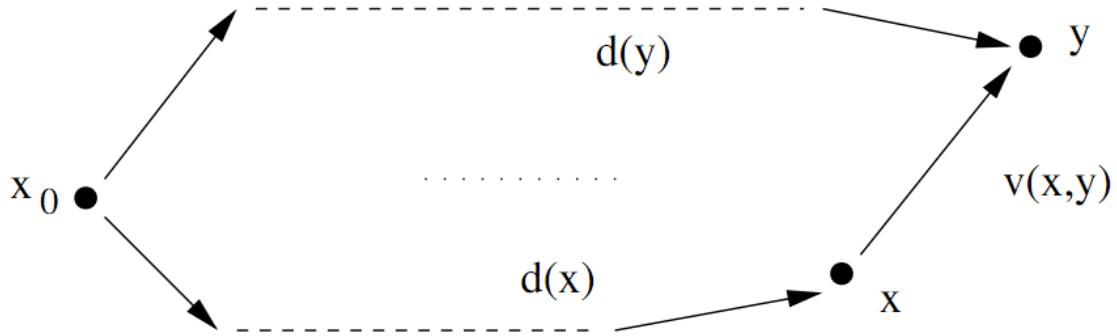
G=(S, A, v)

v(x,y) = valeur de l'arc (x,y)

x_0= sommet de « départ »

Pour chaque sommet s, on mémorise un score d(s) et son père P(s)

- A chaque étape, on considère un sommet x et un successeur y de x. On compare la valeur $d(y)$ à celle que l'on obtiendrait en passant par x, c'est-à-dire $d(x) + v(x, y)$.



- Si cette deuxième valeur est plus petite que $d(y)$, on remplace l'estimation $d(y)$ par $d(x) + v(x, y)$ et le père $P(y)$ par x.

Algorithme de Bellman-Ford : détails

G=(S, A, v)

v = valeur de chaque arc

x_0= sommet de « départ »

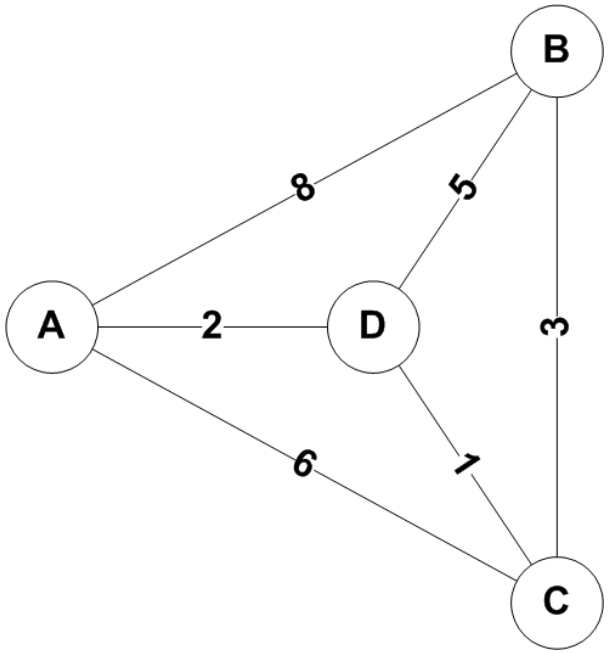
Pour chaque sommet s, l'algorithme détermine le plus court chemin de x_0 à s, et la distance de x_0 à s

Initialisation :
$d(x_0) = 0,\ P(x_0) = nul$
pour tout $s \in S,\ s \neq x_0$ **répéter**
$d(s) = +\infty,\ P(s) = nul$
$fin = FAUX$

tant que $fin = FAUX$ **répéter**
$fin \leftarrow VRAI$
pour tout $x \in S$ **répéter**
pour tout $y \in G(x)$ **répéter**
si $d(x) + v(x, y) < d(y)$ **alors**
$d(y) \leftarrow d(x) + v(x, y)$ (màj distance)
$P(y) \leftarrow x$ (màj père)
$fin \leftarrow FAUX$ (pas encore stabilisé)

fin tant que

Algorithme de Bellman-Ford : exemple



Itération	$d(A)$	$d(B)$	$d(C)$	$d(D)$	$P(A)$	$P(B)$	$P(C)$	$P(D$
Initialisation	0	$+\infty$	$+\infty$	$+\infty$	nul	nul	nul	nul
1	0	7	3	2	nul	D	D	A
2	0	6	3	2	nul	C	D	A
3	0	6	3	2	nul	C	D	A

Algorithme de Dijkstra : principe

- Dans un graphe, on cherche à déterminer les plus courts chemins d'un sommet $\mathbf{x_0}$ vers tous les autres sommets **s** du graphe
  - On construit petit à petit, à partir de $\{x_0\}$, un ensemble M de sommets **marqués**. Pour tout sommet marqué s, l'estimation $d(s)$ est égale à la distance $d(x_0, s)$.
  - A chaque étape, on sélectionne le (un) sommet non marqué x dont la distance estimé $d(x)$ est la plus petite parmi tous les sommets non marqué.
  - On marque alors x (on rajoute x à M), puis on met à jour à partir de x les distances estimées des successeurs non marqués de x.
  - On recommence, jusqu'à épuisement des sommets non marqués.

Algorithme de Dijkstra : détails

Initialisation

$d(x_0) = 0$, $P(x_0) = nul$

aucun sommet n'est marqué

$min\_dist\_M = 0$ (minimum des distances estimées des sommets non marqués)

pour tout $s \in S$, $s \neq x_0$ **répéter**

$d(s) = +\infty$, $P(s) = nul$

répéter

chercher x non marqué tel que $d(x) = min\_dist\_M$

marquer x

pour tout $y \in G(x)$, y non marqué **répéter**

si $d(x) + v(x, y) < d(y)$ **alors**

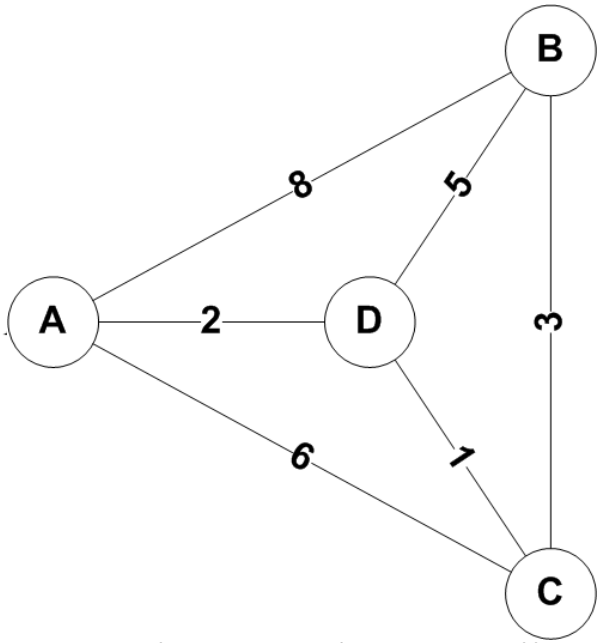
$d(y) \leftarrow d(x) + v(x, y)$ (màj distance)

$P(y) \leftarrow x$ (màj père)

$min\_dist\_M = \min\{d(s),\ s \notin M\}$

jusqu'à ce que $min\_dist\_M = +\infty$

Dijsktra : exemple



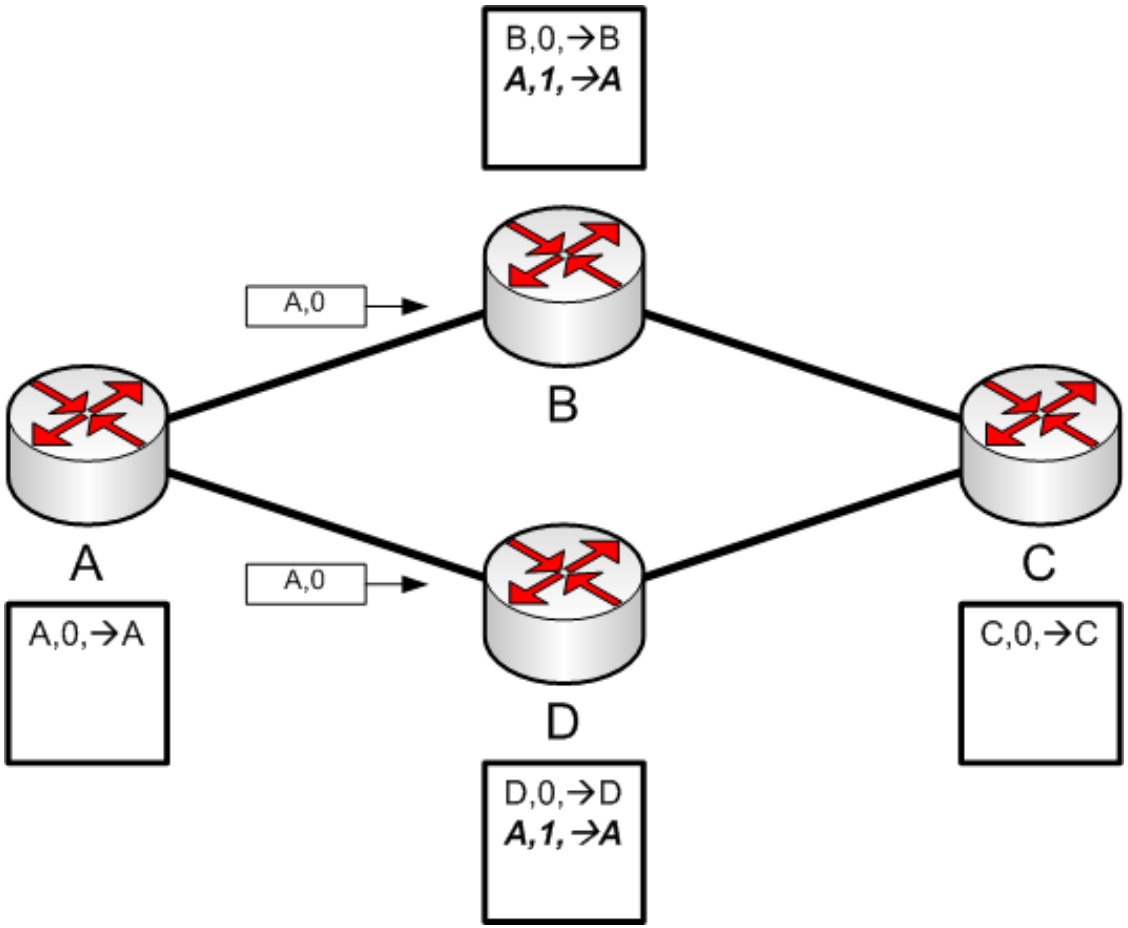
M	$d(A)$	$d(B)$	$d(C)$	$d(D)$	$P(A)$	$P(B)$	$P(C)$	$P(D$
$\emptyset$	0	$+\infty$	$+\infty$	$+\infty$	nul	nul	nul	nul
$\{A\}$		8	6	2		A	A	A
$\{A, D\}$		7	3			D	D	
$\{A, D, C\}$		6				C		
$\{A, D, C, B\}$	0	6	3	2	nul	C	D	A

Routage à vecteur de distance : RIP

- RIP = variante **distribuée** de l'algorithme de Bellman-Ford
- Initialisation
  - Chaque nœud ne se connaît que lui-même, et maintient une table
  - Sa table ne contient que {« moi », 0, → « moi »} qui signifie :
    - pour joindre « moi », cela coûte 0, et il faut passer par « moi »
- **Périodiquement,** chaque nœud envoie sa table à ses voisins
  - Liste de couples {nœud, coût}
- A la réception de ces informations, le nœud récepteur compare avec le contenu de sa propre table
  - Si nœud inconnu
    - Ajout {nœud, **1+coût,** → nœud émetteur}
  - Si nœud déjà connu **mais coût mémorisé > (1 + coût)**
    - Modification {nœud, **1+coût, → nœud émetteur**}
  - Si nœud déjà connu, **même route mais avec coût mémorisé < (1 + coût)**
    - Modification {nœud, **1+coût,** → nœud émetteur}
    - Prise en compte des changements de topologie

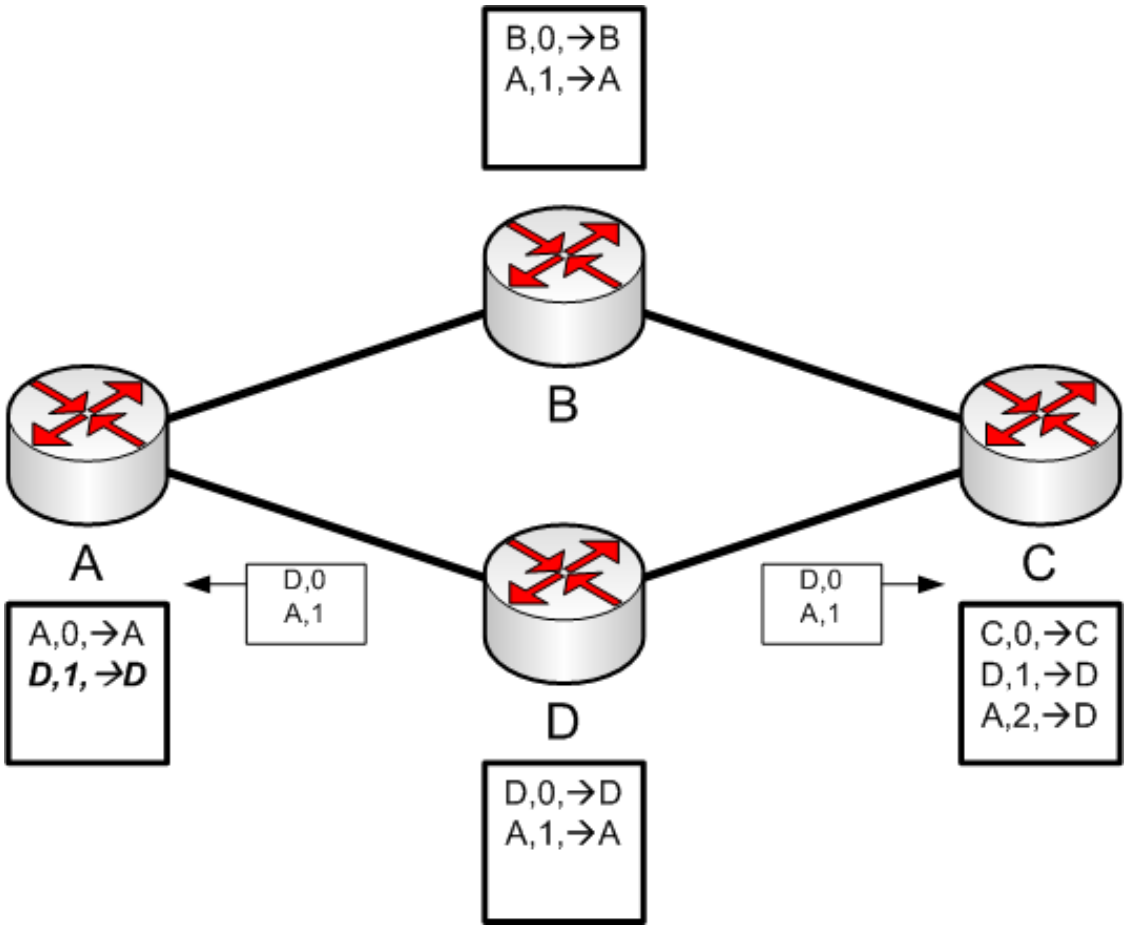
Principes de RIP : exemple (1)

1. Envoi
A→B,D



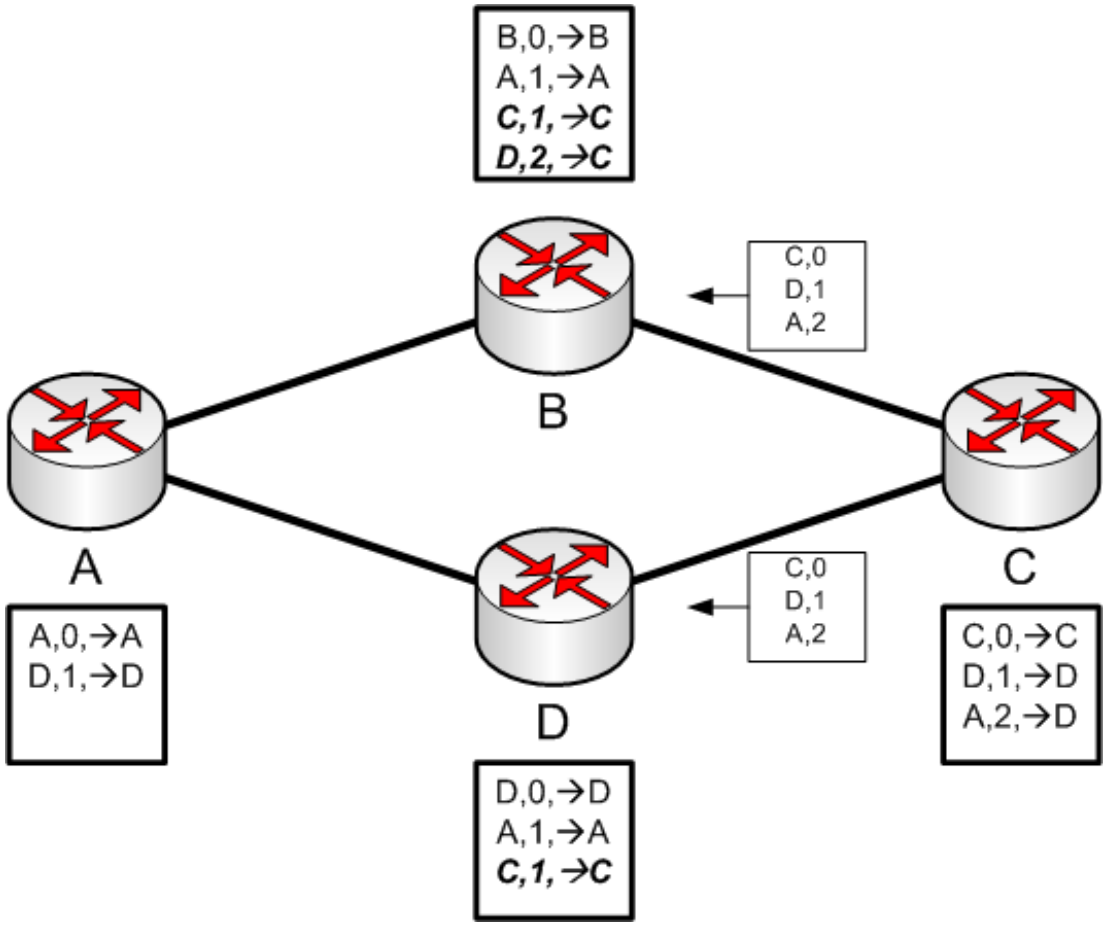
Principes de RIP : exemple (2)

2. Envoi
D→A,C



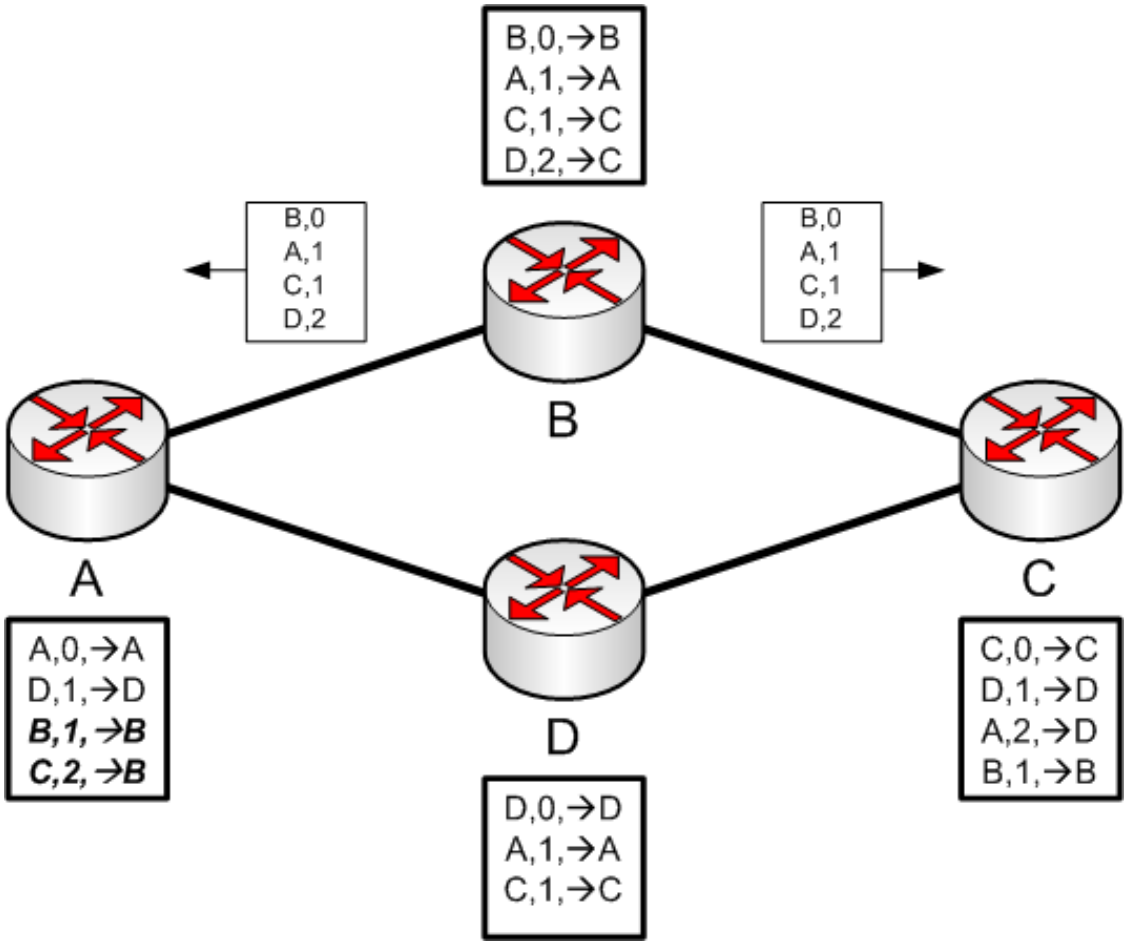
Principes de RIP : exemple (3)

3. Envoi
C→B,D



Principes de RIP : exemple (4)

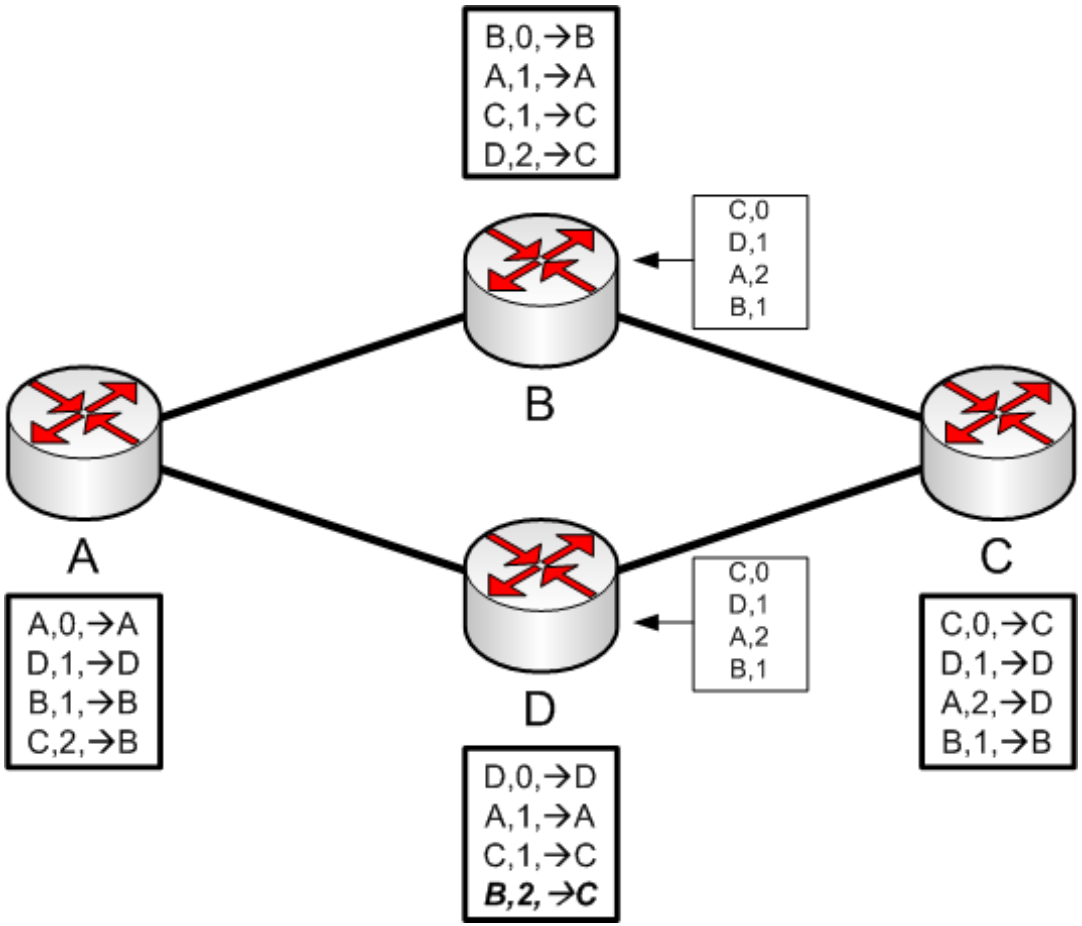
4. Envoi
B→A,C



Principes de RIP : exemple (5)

5. Envoi C→B,D

Convergence atteinte !

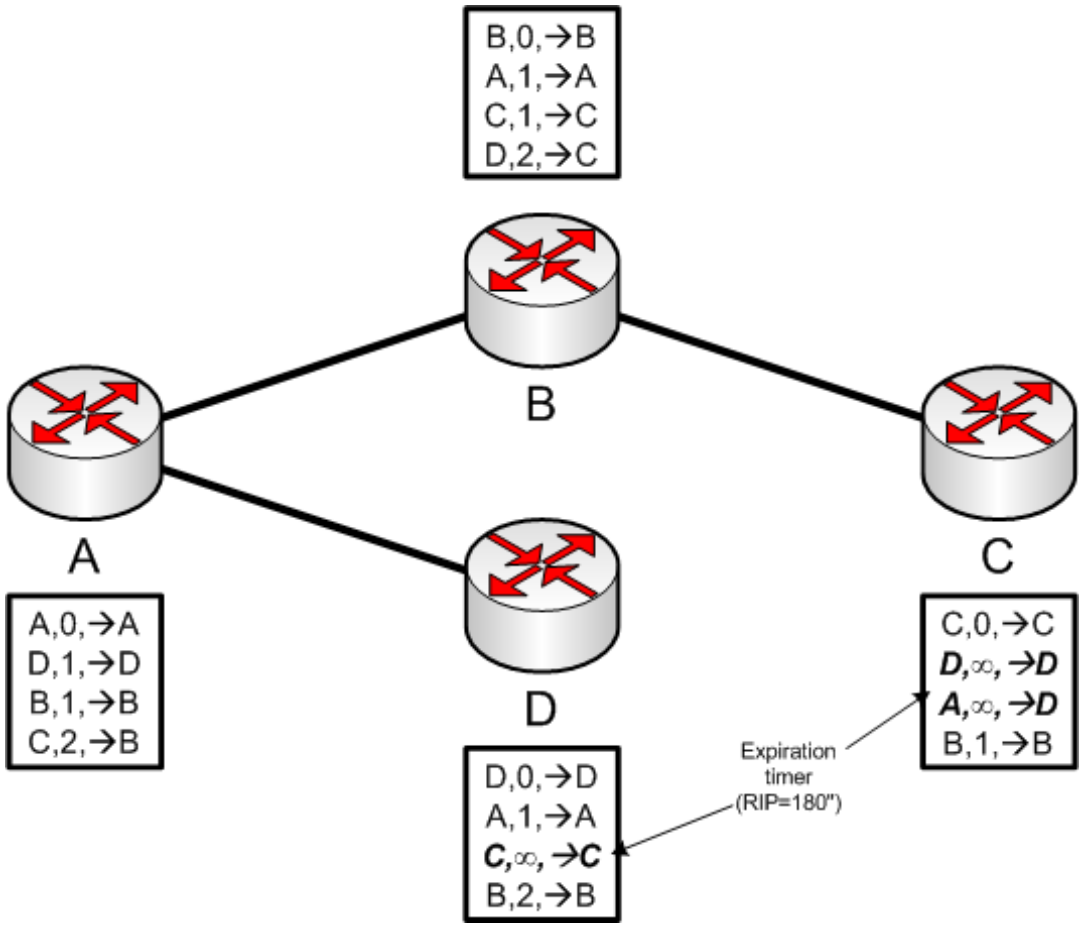


RIP : tolérance aux pannes (1)

D-C se coupe

A l'expiration d'un timer, les routes non rafraichies sont marquées « inatteignables »

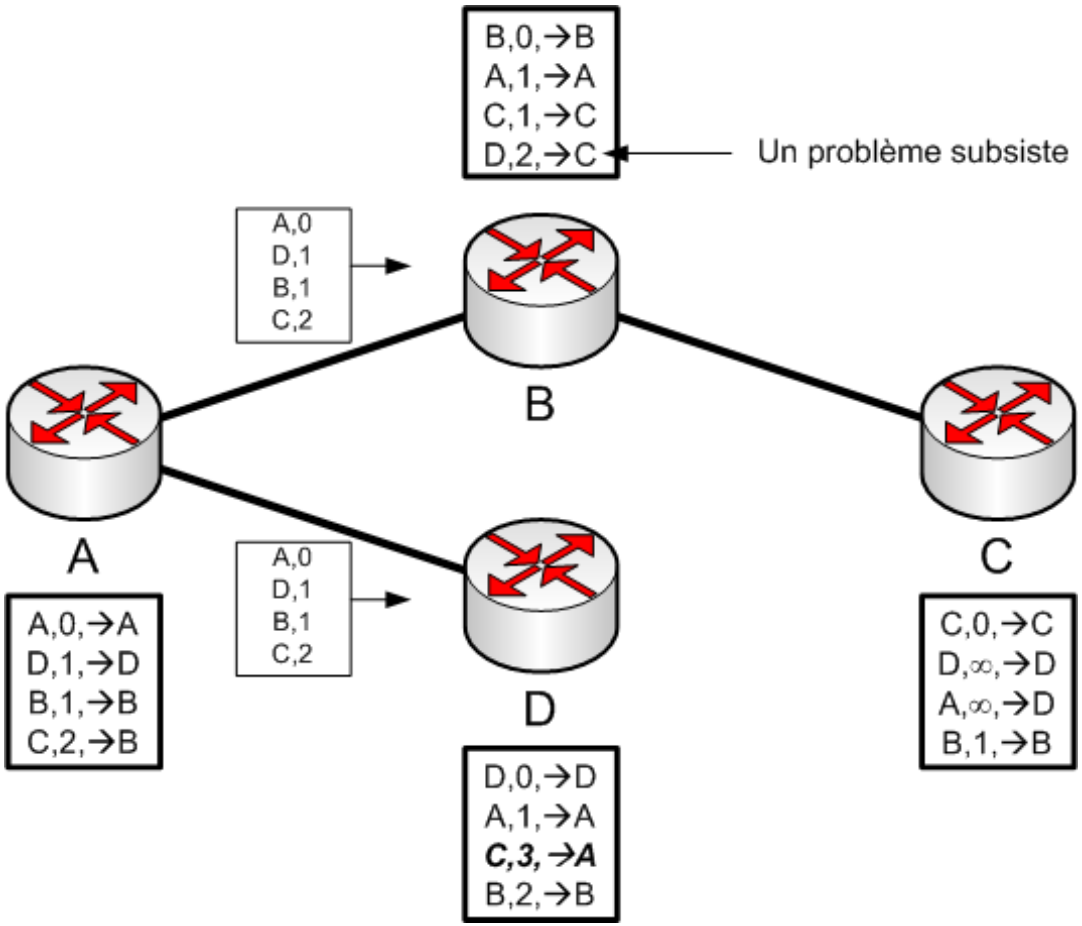
Une route devenue inatteignable est ***prise en compte*** !



RIP : tolérance aux pannes (2)

La convergence reprend d'elle-même vers une nouvelle solution

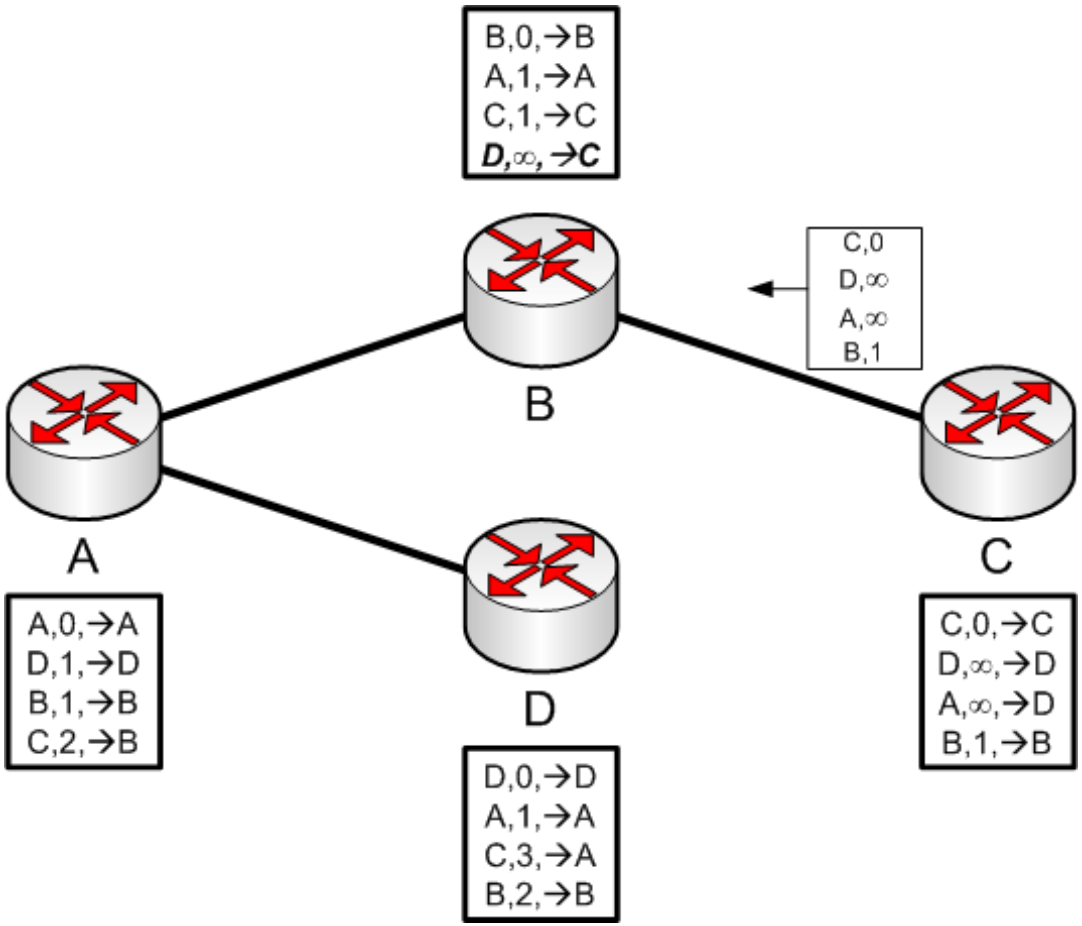
Une nouvelle route a été trouvée pour aller de D à C



RIP : tolérance aux pannes (3)

Durant cette nouvelle convergence, des métrique infinies peuvent se propager de proche en proche.

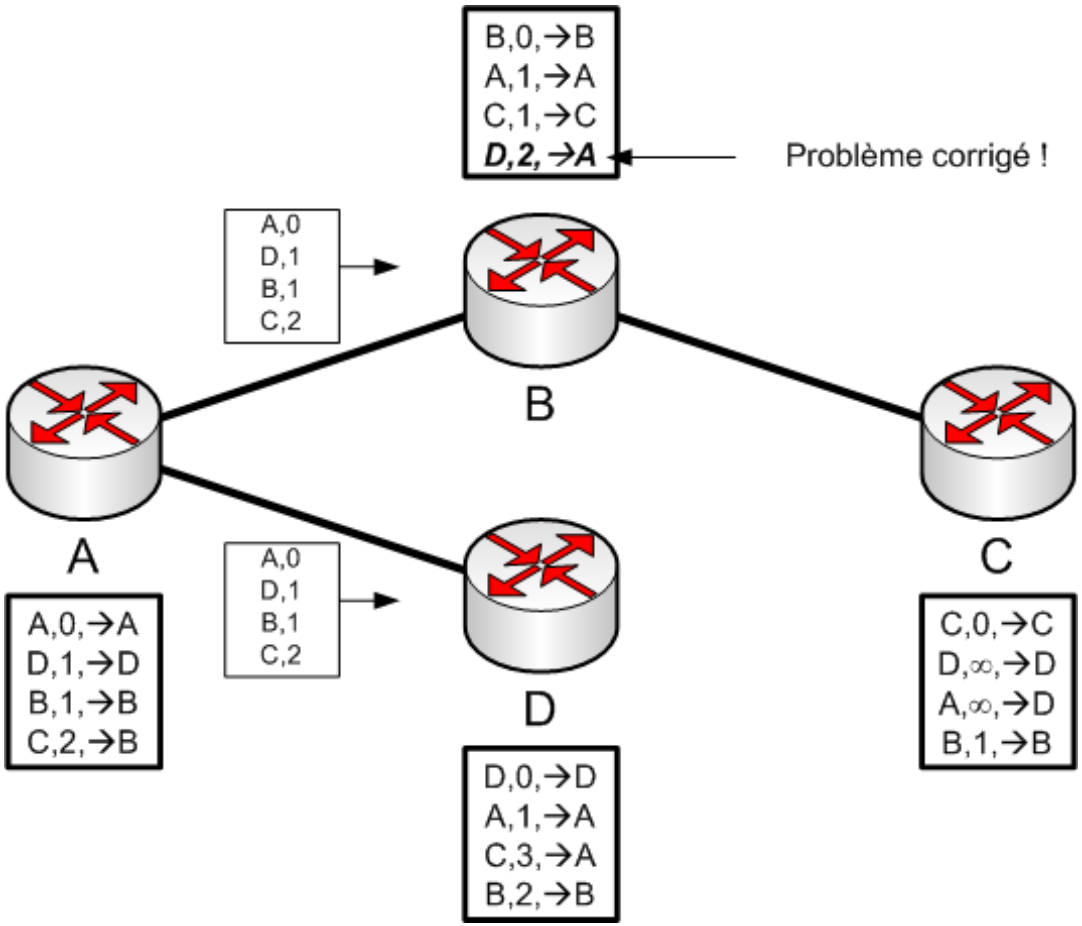
Cela ne pose aucun problème particulier (dans cet exemple ☺)



RIP : tolérance aux pannes (4)

Une nouvelle convergence est atteinte.

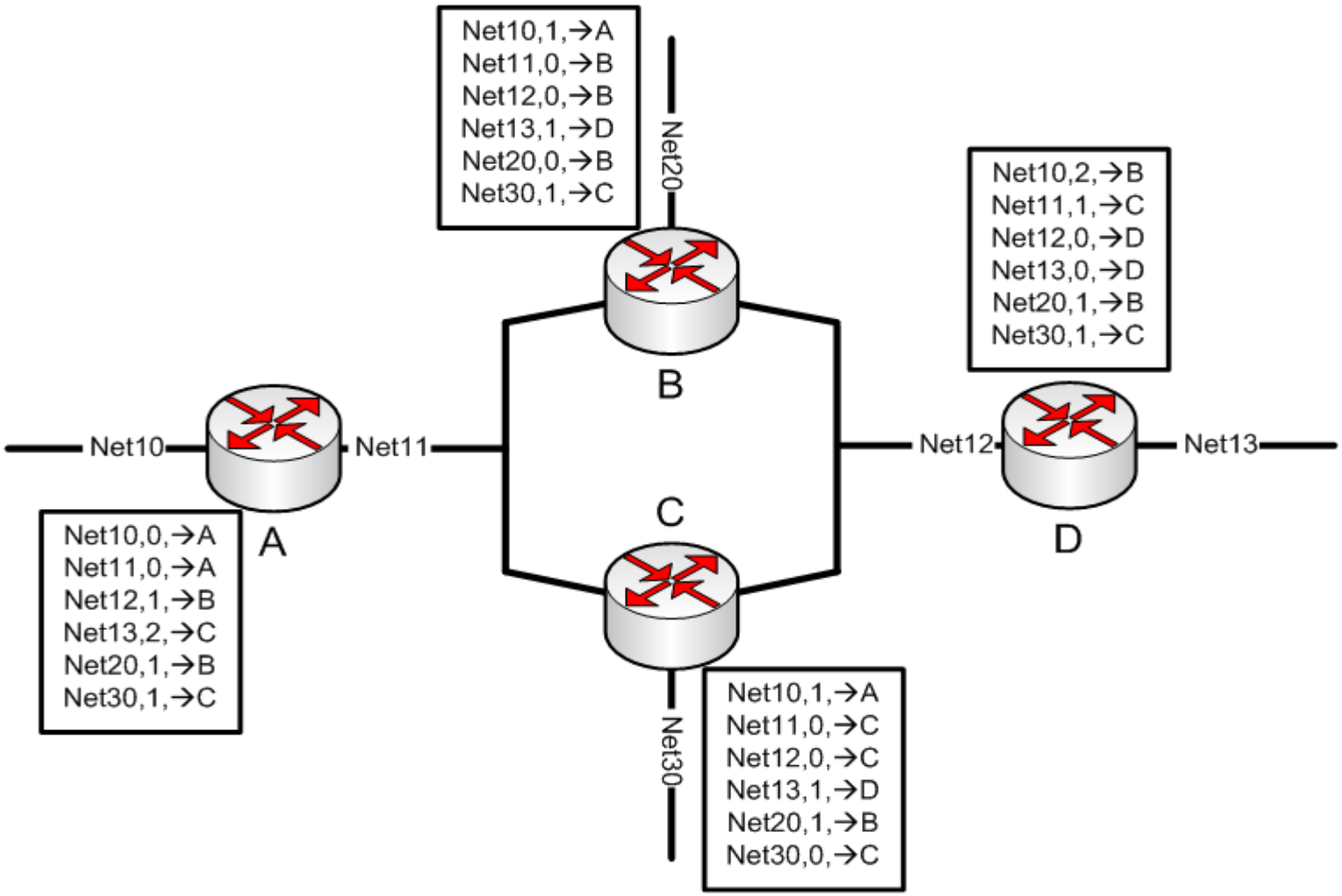
La liaison coupée a été contournée automatiquement.



RIP en pratique

- Connaître les meilleurs chemins vers des routeurs n'intéresse personne ! ☺
- RIP propage des routes vers des réseaux !
  - Exemple : « 101.0.0.0/8, m=3, gw=194.252.10.3 »
  - Ces réseaux sont attachés directement à certains routeurs de l'ensemble du graphe
- Le graphe n'est pas vraiment un graphe « normal »
  - Il n'y a pas que des liaisons point-à-point : plusieurs routeurs peuvent être reliés au même réseau !
  - Ceci peut générer des effets indésirables car RIP travaille en diffusion
    - L'émission d'une table RIP n'est pas destinée à un routeur voisin précis, mais à **l'ensemble des routeurs voisins**
    - Plusieurs routeurs peuvent donc être joints par une même émission de message RIP

RIP en pratique : exemple de réseau

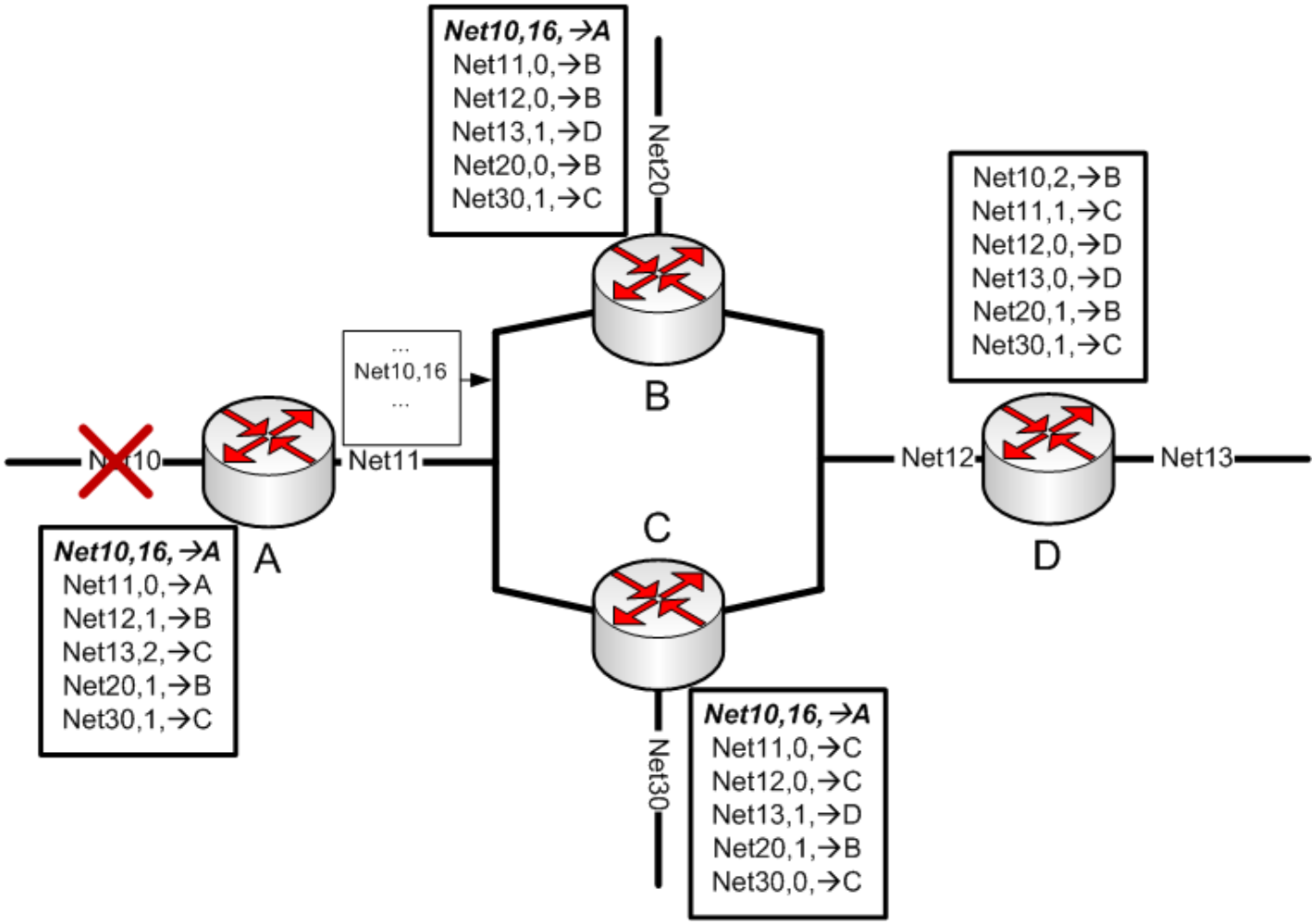


RIP en pratique : problème (1)

Net10 tombe en panne

A diffuse cette information sur Net11.

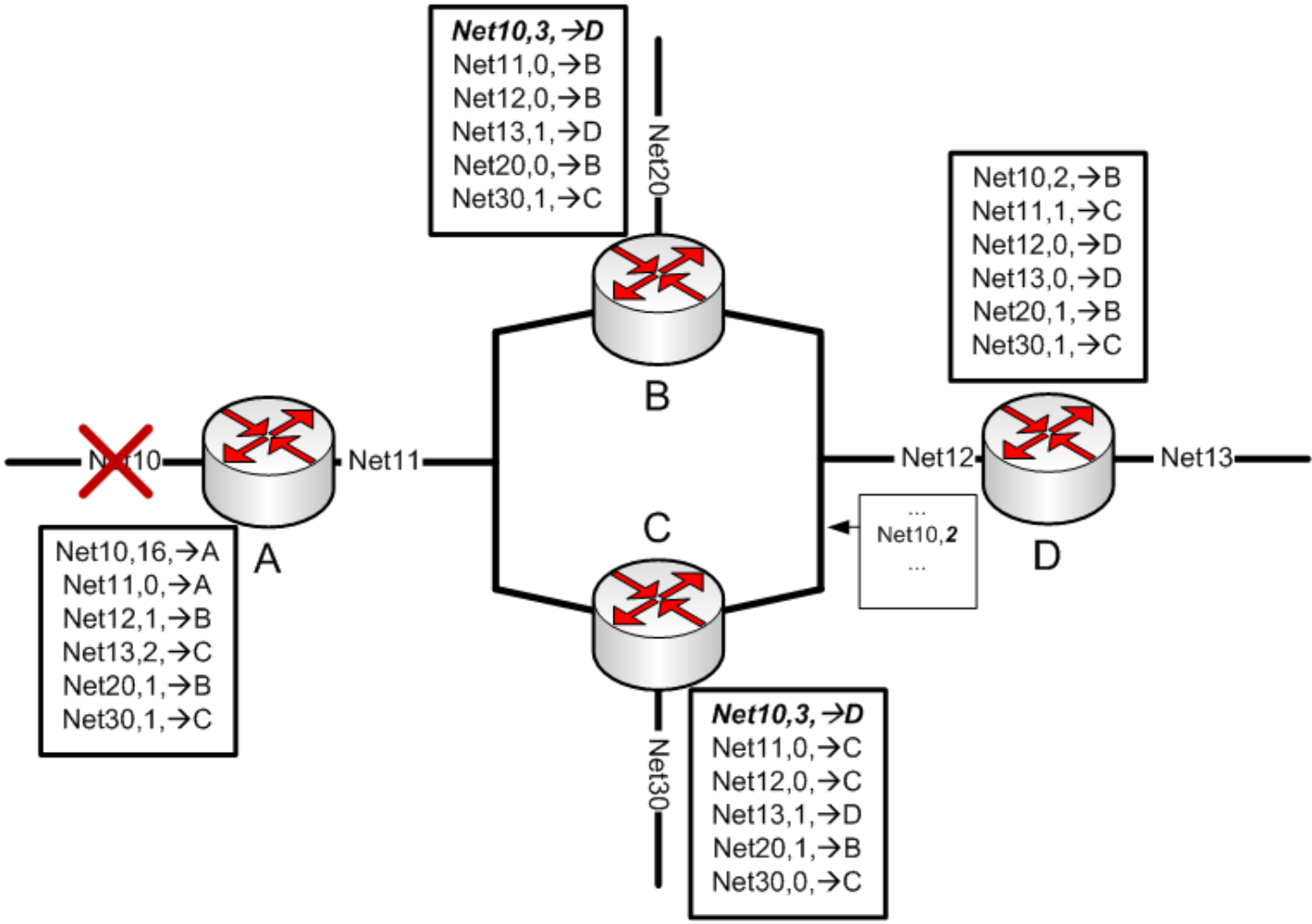
B et C captent cette information et mettent leur table à jour.



RIP en pratique : problème (2)

D continue sont travail normal, mais il diffuse sans le savoir une information erronée…

B et C captent cette information et en tiennent malheureusement compte !

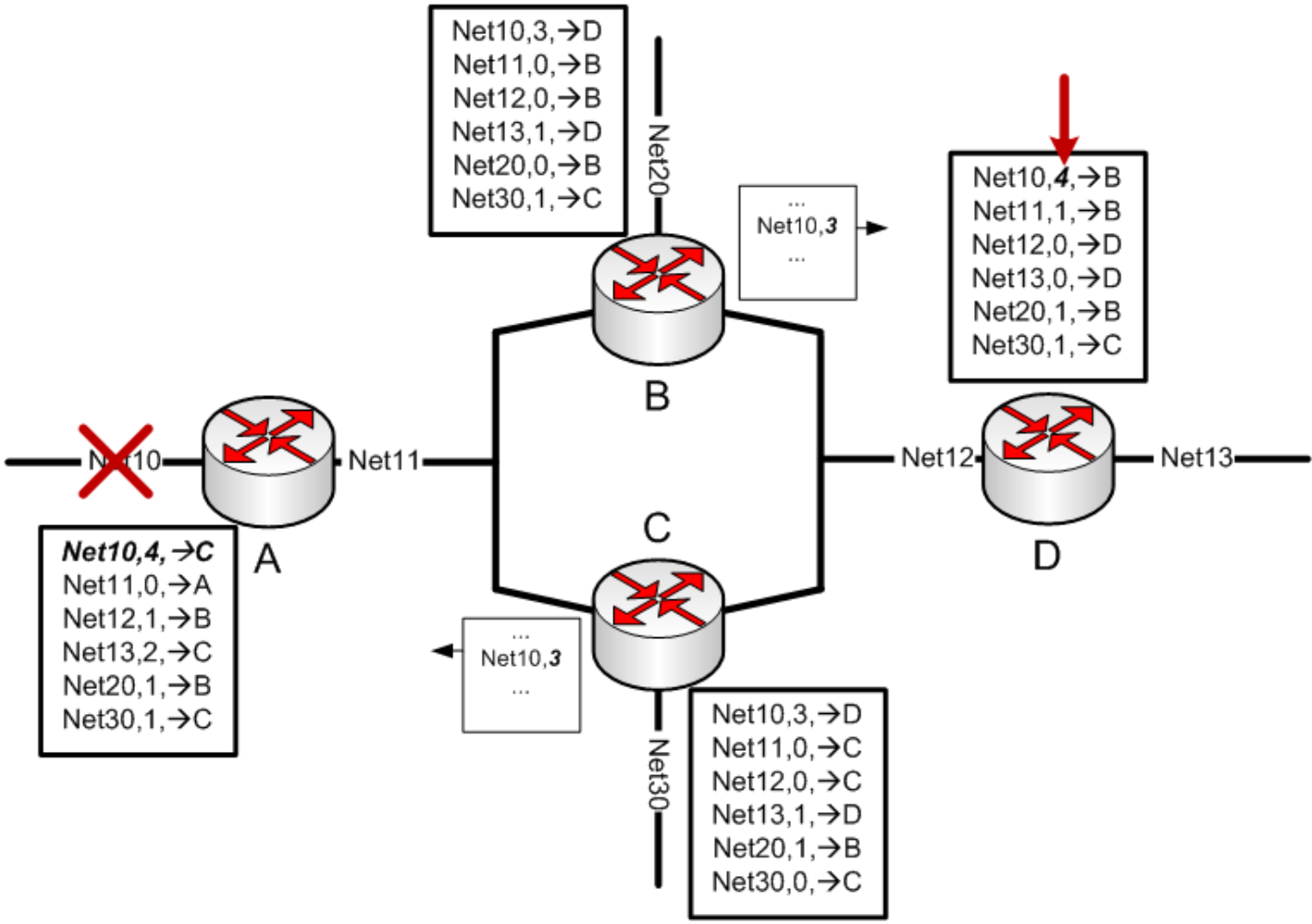


RIP en pratique : problème (3)

B et C propagent aussi sans le savoir une mauvaise information, etc.

Les routeurs vont se répliquer l'information avec un coût incrémenté à chaque fois (règle de prise en compte des changements de topologie)

A terme, tous les routeurs auront « 16 » comme métrique vers « Net10 » mais la convergence aura été très longue !



RIP : « patchs »

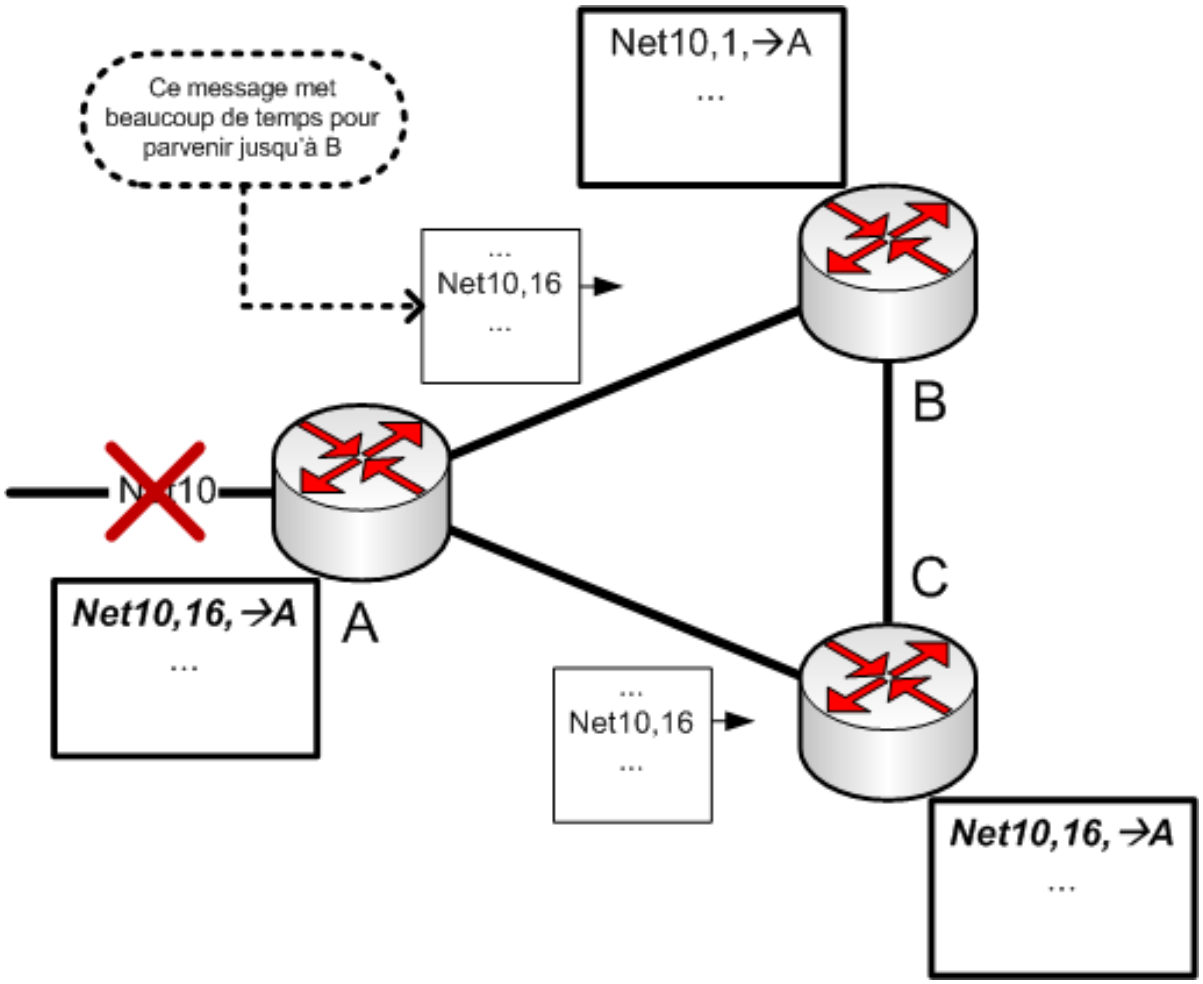
- Métrique max = 15
  - 16 = « infini » ou « route coupée »
  - Conséquence : diamètre du réseau = 15 routeurs maxi
- Règle du « Split-Horizon »
  - Non-retransmission vers « NetX » des routes apprises depuis « NetX »
  - Permet d'éviter les problèmes de bouclage (exemple précédent)
- Timers : assurent la **stabilité** de l'algorithme
  - « Hello Time » (30s)
    - Fréquence d'émission des messages RIP
  - « Expiration Timer » (T1=180s)
    - Pas de message RIP depuis T1 secondes en provenance du voisin X → panne sur la liaison !
    - Toutes les routes apprises depuis X sont notées « 16 »
  - « Garbage collection » (T2=240s)
    - Elimination des routes de métrique « 16 » existant depuis plus de T2"

RIP : split-horizon pris en défaut

A détecte un défaut

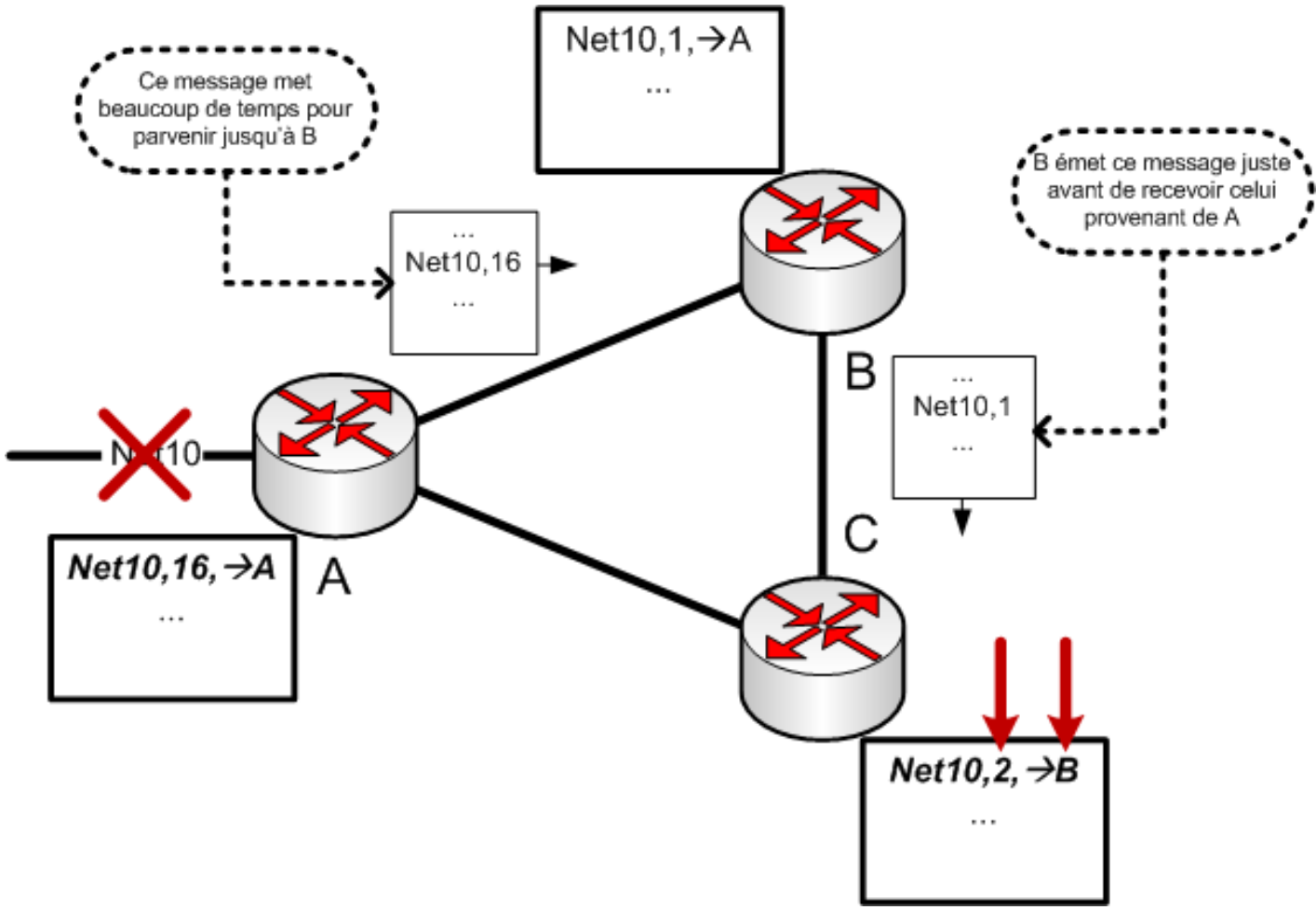
Ce défaut est propagé aux voisins

MAIS…



RIP : split-horizon pris en défaut

C se retrouve en possession d'une information temporairement erronée… qu'il va propager vers A car cela ne viole plus la règle du split-horizon !

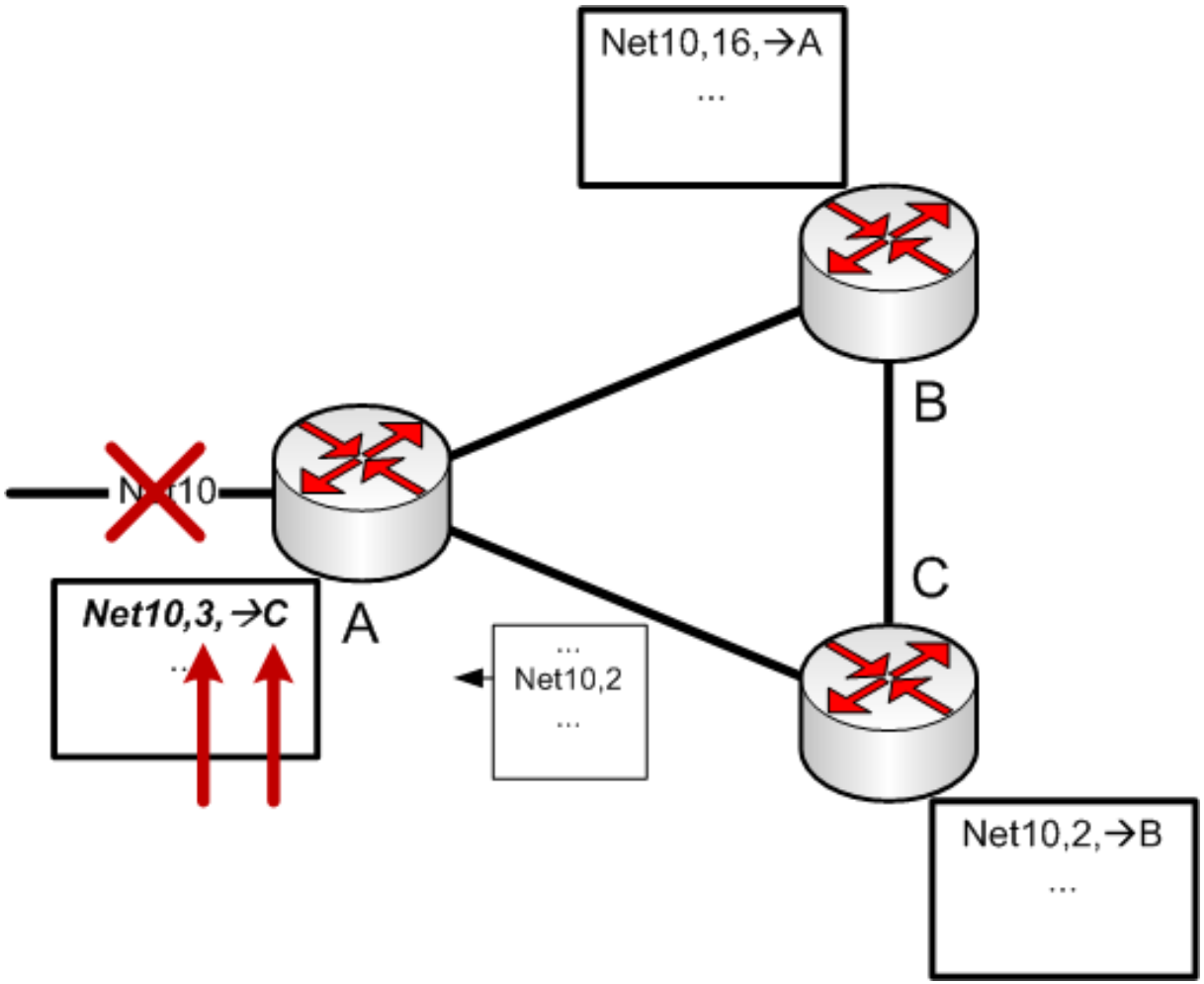


RIP : split-horizon pris en défaut

Et le « count-to-infinity » surgit à nouveau… ☹

Solution : « hold-down » timer

Une route en défaut (métrique = 16) sera conservée pendant un **certain temps** même si un message invalidant ce défaut est reçu



RIP : conclusion

- Défauts
  - **Convergence lente** dans les grands réseaux (propagation des défauts)
    - Hello Time = 30" - **Expiration Timer = 180"** - **Hold Down Timer = 120"**
  - Notion de coût simpliste (nombre de sauts)
    - RIP préfère passer par 2 routeurs reliés par une liaison 64Kb/s plutôt que par 3 routeurs reliés par des liaisons 2 Mb/s !!!
  - Utilisation importante de bande passante
- Avantages
  - Simple
    - À comprendre ☺ - À configurer - À Implémenter
  - Peu d'administration, peu gourmand en ressources CPU et mémoire
  - Implémenté sur **tous** les routeurs
- Conclusion
  - RIP est toujours **très utilisé** dans les réseaux de petite taille (< 10 routeurs)
  - Certaines variantes propriétaires (ex: CISCO « EIGRP ») réduisent ses défauts

Routage à états de liens : OSPF

- Apports d'OSPF
  - Notion de coût assouplie
    - Un coût est affecté à chaque interface sortante
    - Ex: coût faible pour un débit élevé, un engorgement faible, etc.
  - Equilibrage de charge
    - Utilisation de plusieurs routes vers la même destination
    - → impact sur l'algorithme de routage
  - Réaction rapide aux changements de topologie
    - « Update » émis dès qu'un changement (panne) est détectée
    - Seul est émis le changement d'état !

OSPF : algorithme

- 3 phases principales
  - Phase 1 : cartographie du réseau
    - Chaque nœud
      - Découvre ses voisins et détermine le coût du lien qui les relie
      - Diffuse les informations précédentes au reste du réseau
      - Relaie les diffusions reçues
    - Après un certain temps, chaque nœud connaît la cartographie complète du réseau
      - Matrice de coûts pour aller d'un nœud X à un nœud Y
  - Phase 2 : calcul du plus court chemin
    - Chaque nœud détermine le plus court chemin de lui-même vers tous les autres nœuds
    - OSPF utilise l'algorithme de Dijsktra pour ce calcul
    - Les résultats sont stockés dans une base de données topologique
  - Phase 3 : mise à jour de la table de routage
    - Utilise la base de données topologique

OSPF : conclusion

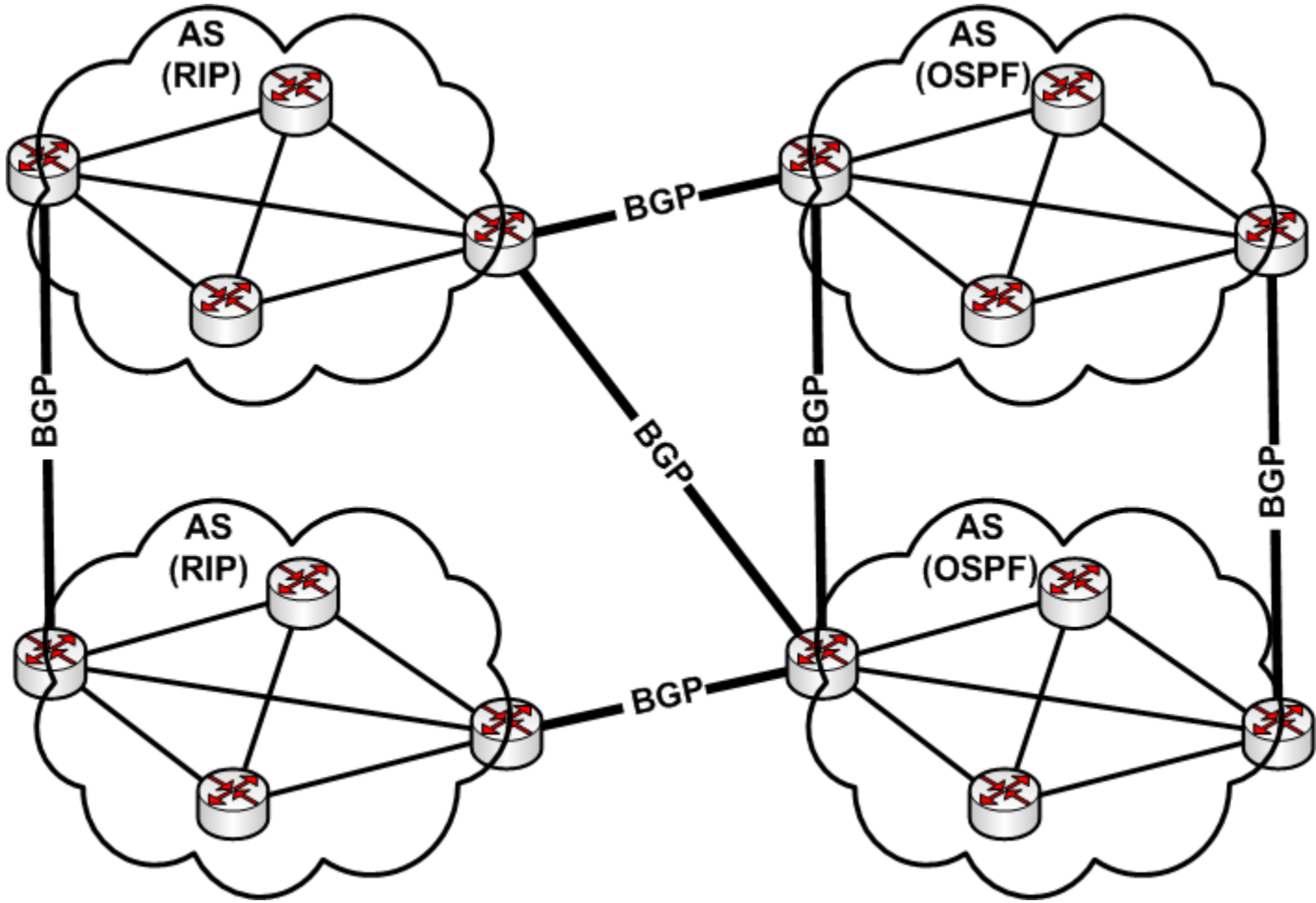
- Avantages
  - Résout tous les problèmes posés par RIP
- Inconvénients
  - Protocole complexe
    - Compréhension par les administrateurs ☺
    - Paramétrage des routeurs
    - Gourmand en ressources CPU et mémoire
- Conclusion
  - OSPF n'est pas utilisé dans les petits réseaux

Routage hiérarchique

AS = Autonomous System (ex: opérateur télécom, grande entreprise)

BGP = Border Gateway Protocol

- Métrique unifiée
- Résumés de routes
- Routes interdites
- Etc.



TCP / IP

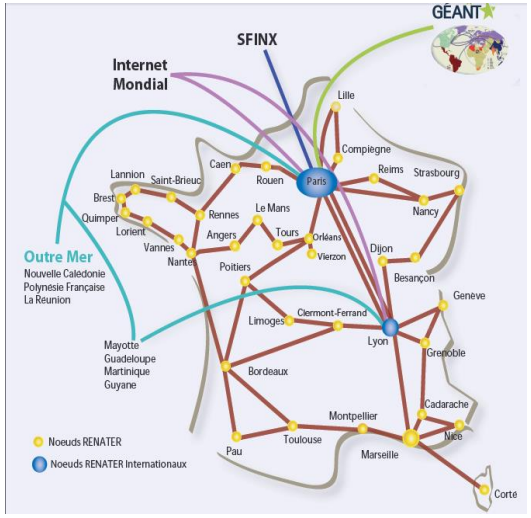
Notion de base, adressage IP

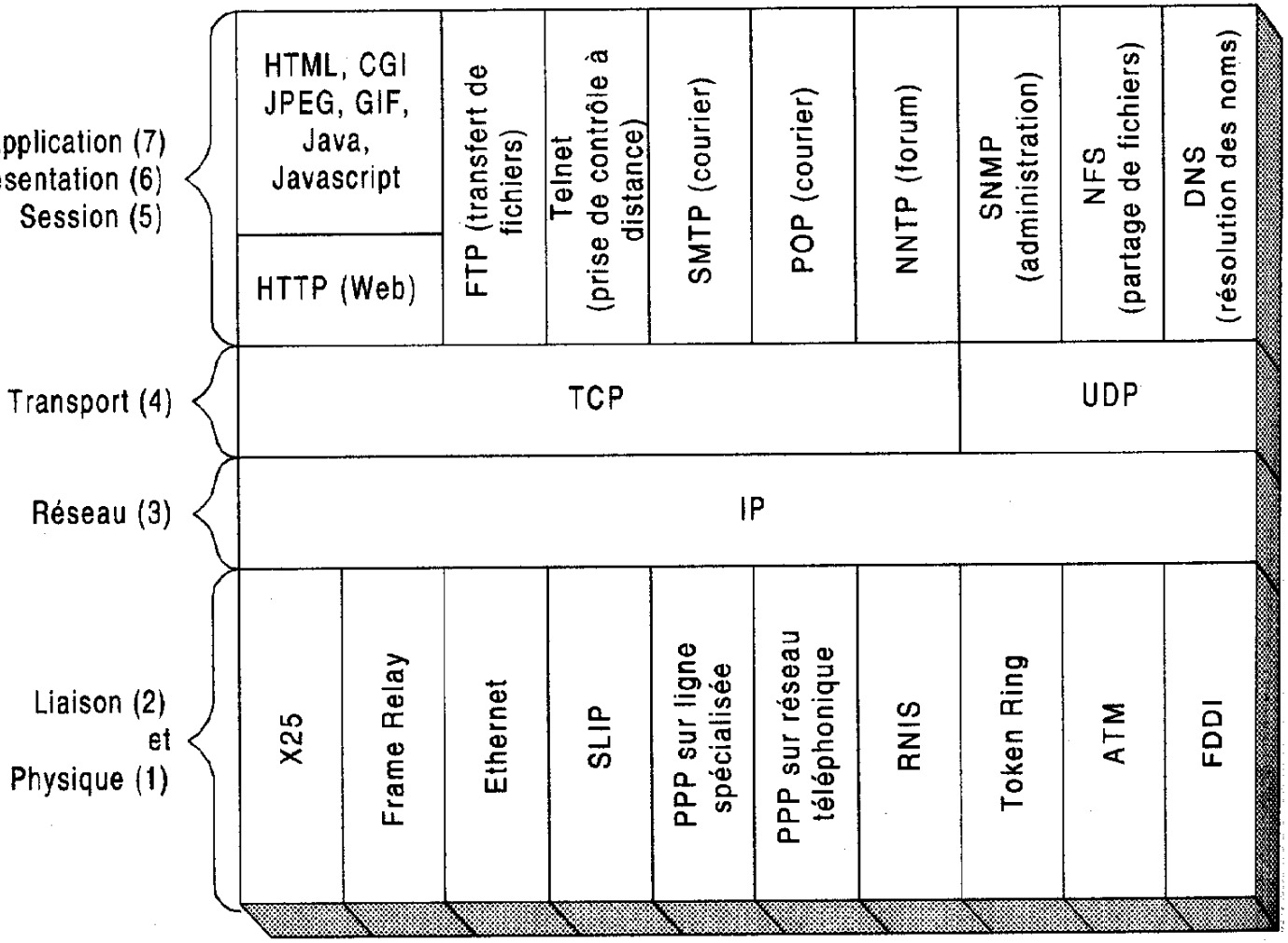
Datagramme IP, routage IP

La philosophie initiale TCP/IP

- TCP/IP : but = interconnexion de réseaux sur une base planétaire
- La technologie est constituée par des protocoles de base (suite TCP/IP) qui offrent les services de base du transfert des données
  - Transport de datagrammes : service élémentaire du routage de paquets
  - Transport de messages sécurisés (TCP) : service orienté connexion permettant d'acheminer des données en garantissant leur intégrité
- Ces services de base sont indépendants du support de transmission; adaptables à toute sorte de media depuis les réseaux locaux jusqu'aux réseaux longue distance

La philosophie initiale TCP/IP

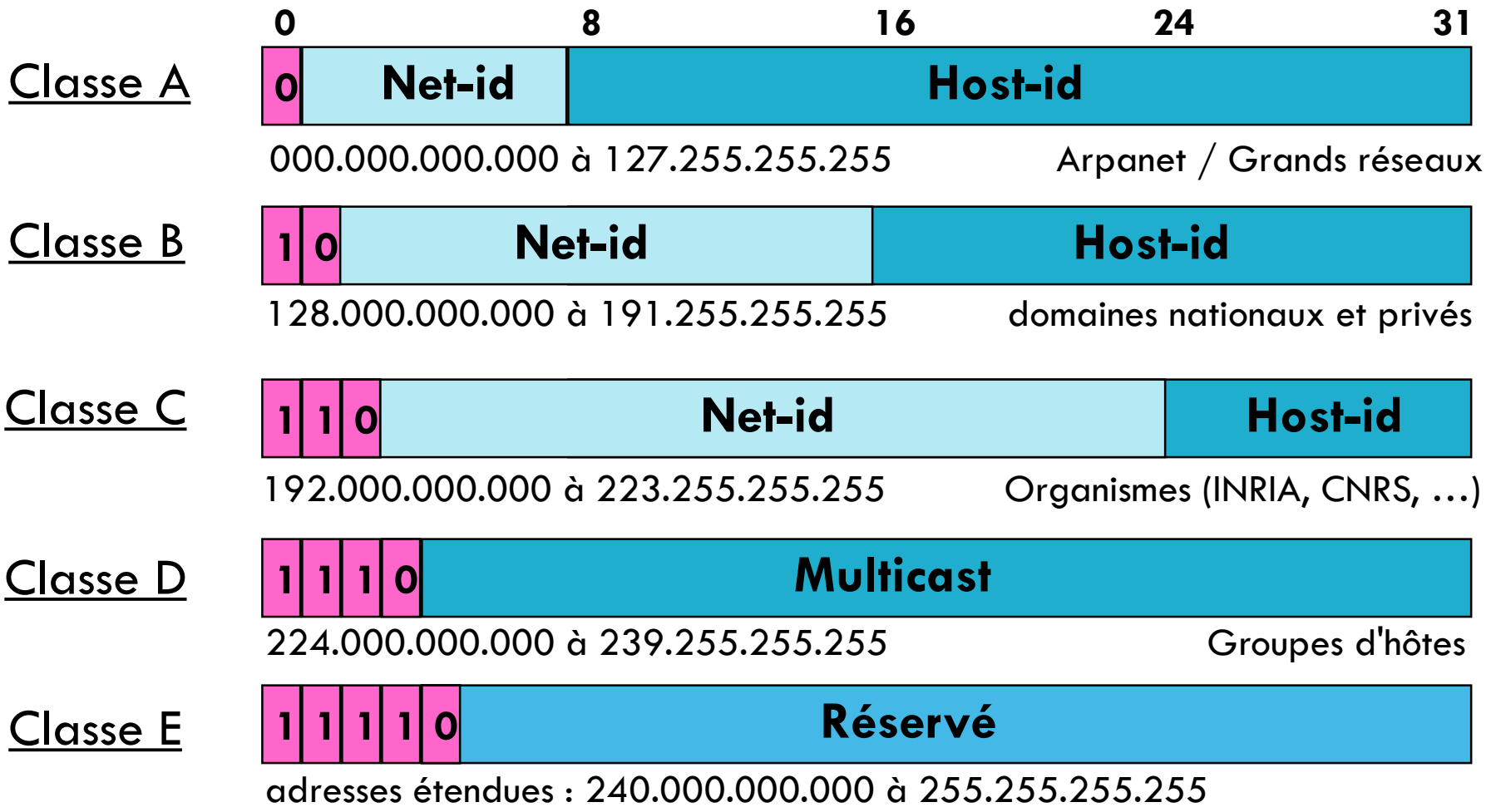




Adressage IPv4

- Adressage "à plat" par opposition à un adressage hiérarchisé
  - Adressage binaire compact assurant un routage efficace
  - Mise en oeuvre de l'interconnexion d'égal à égal
- Utilisation de noms pour identifier des machines (DNS)
- Les classes d'adresses
  - Une adresse IP = 32 bits constituée d'une paire (netid, hostid)
  - 10000000 00001010 00000010 00011110 → 128.10.2.30
  - Cette paire est structurée de manière à définir 5 classes
- Adresses Publiques ou Privées (RFC 1597/1918)
  - Classe A : 10.X.X.X
  - Classe B : 172.[16→31].X.X
  - Classe C : 192.168.X.X

Adressage IPv4

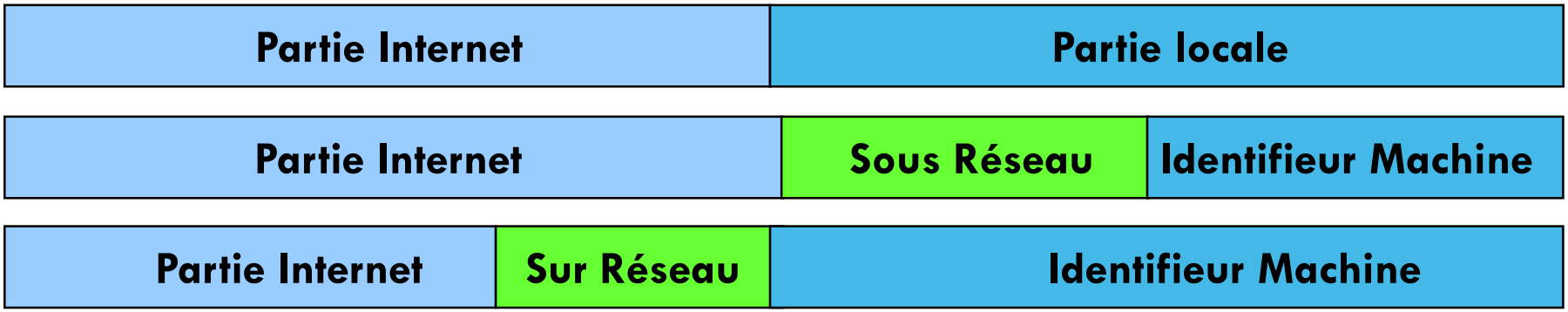


Adressage IPv4

- Adresses de diffusion : la partie host ne contient que des 1
- Adresse de diffusion dirigée : netid est une adresse réseau spécifique → la diffusion concerne toutes les machines situées sur le réseau spécifié : 191.20.255.255 pour les hosts du réseau 191.20.
- Adresse de boucle locale : Loopback = 127.0.0.1 "localhost"

Adressage IPv4

- Masque de sous-réseau (RFC 1878) ou de sur Réseau (RFC 1517-1520).

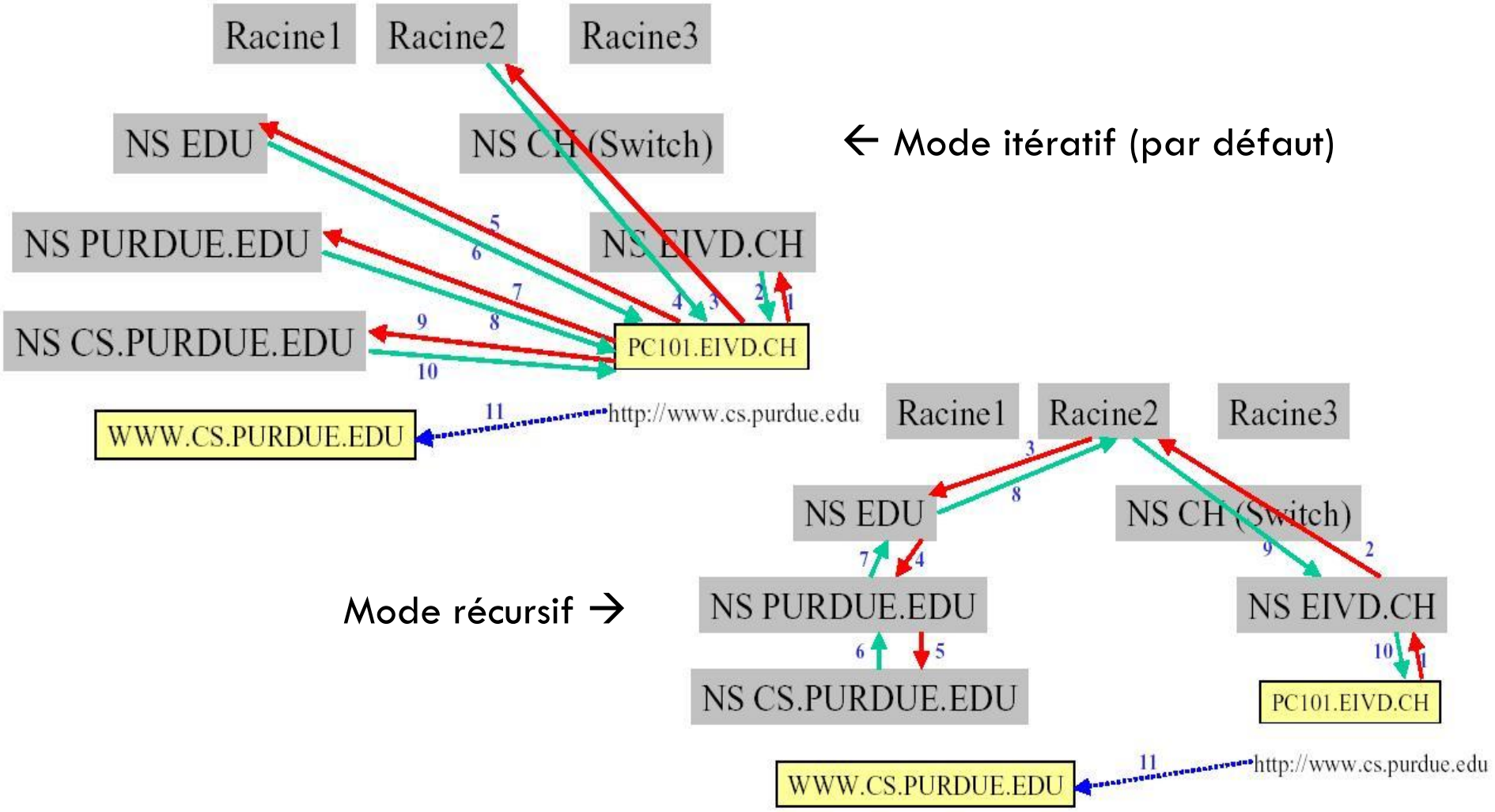


- CIDR = Classless Inter-Domain Routing

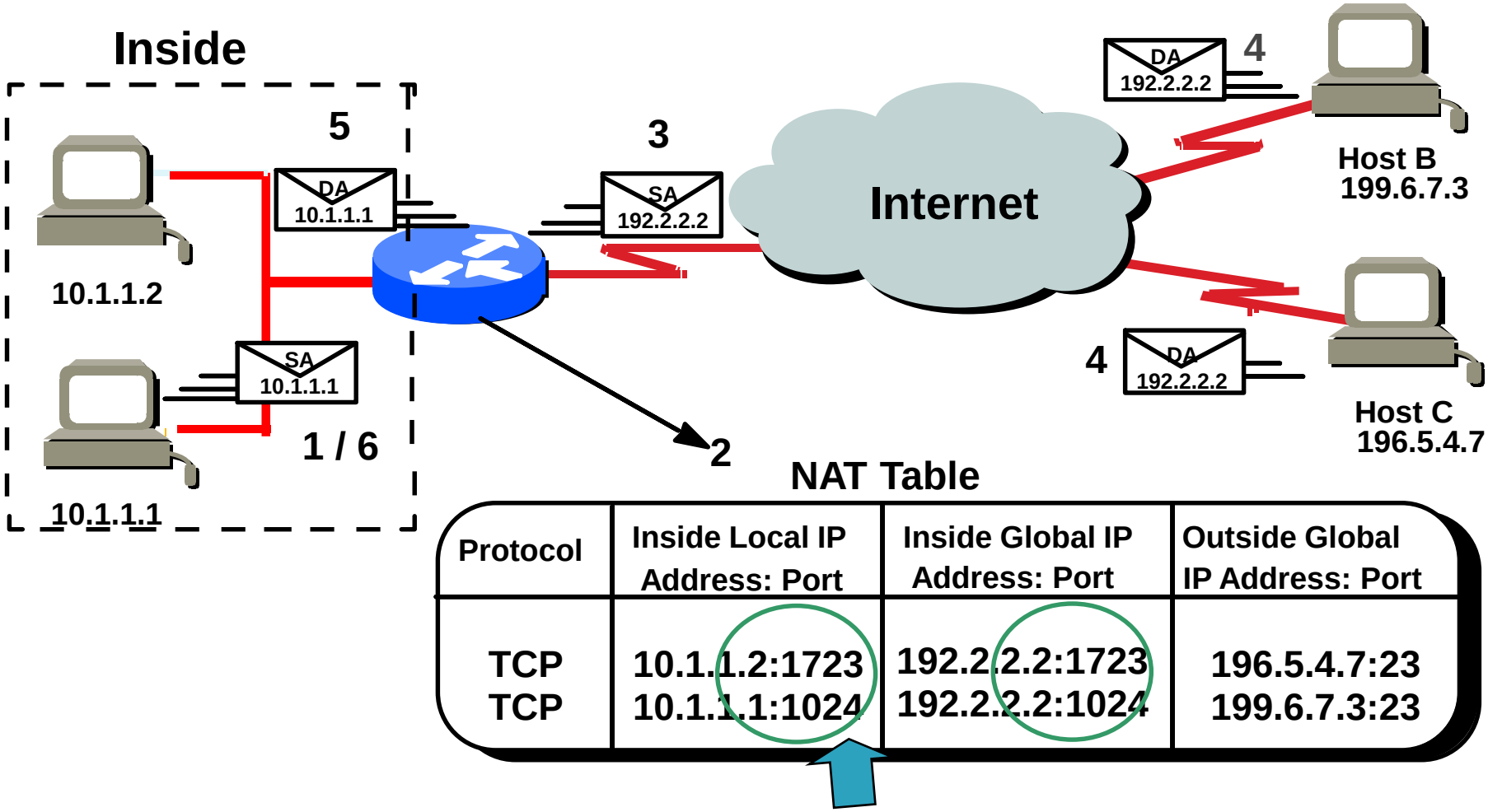
  Notation : Adresse IP / Prefix → 192.33.11.0/22

- Adresses IP dynamiques : DHCP (Dynamic Host Config. Protocol)
- NAT : Network Address Translation
- Manque d'adresses IPv4 (32 bits) → IP version 6 (128 bits)

DNS : Domain Name Server

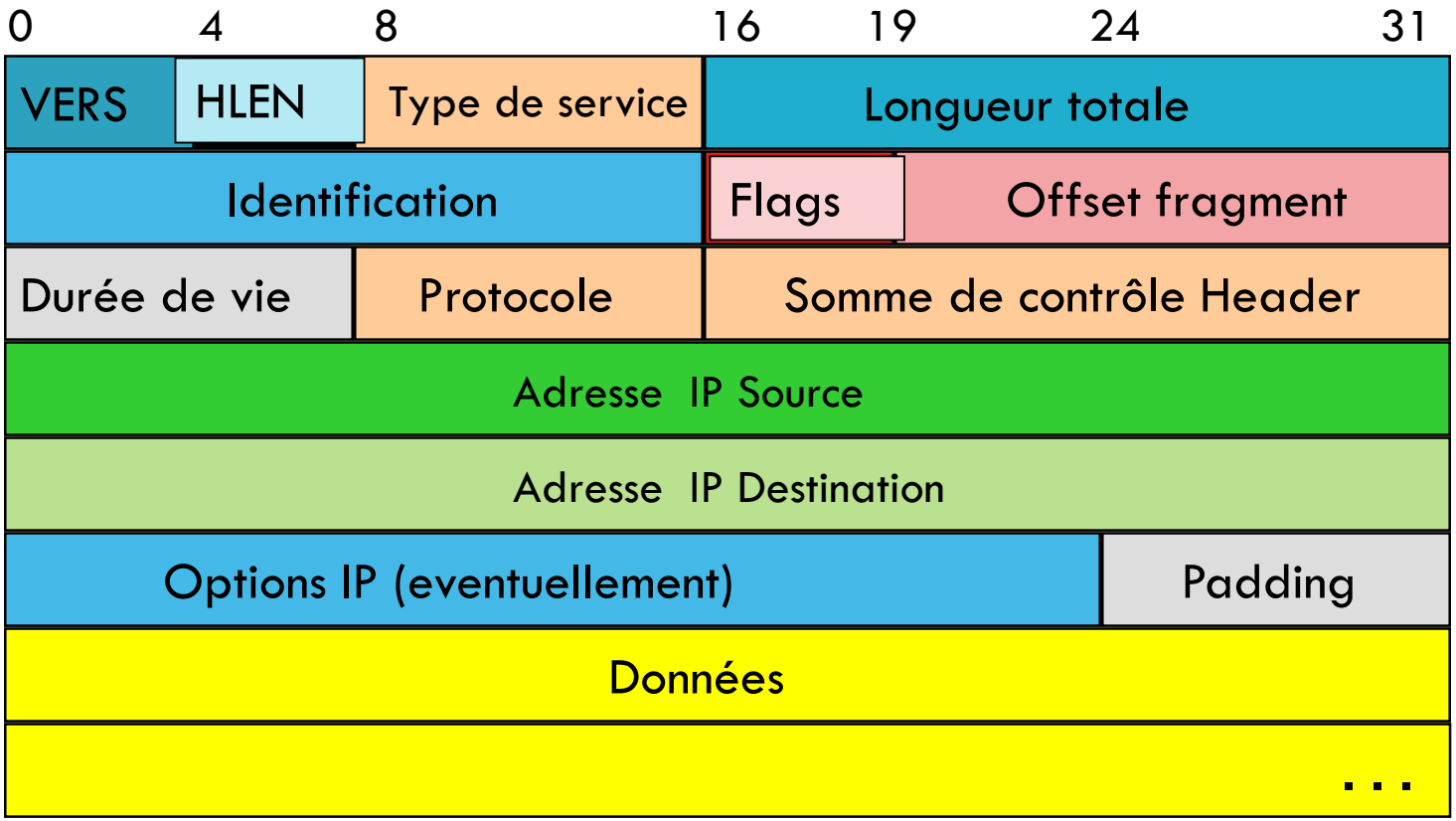


NAT : Translation d'adresses



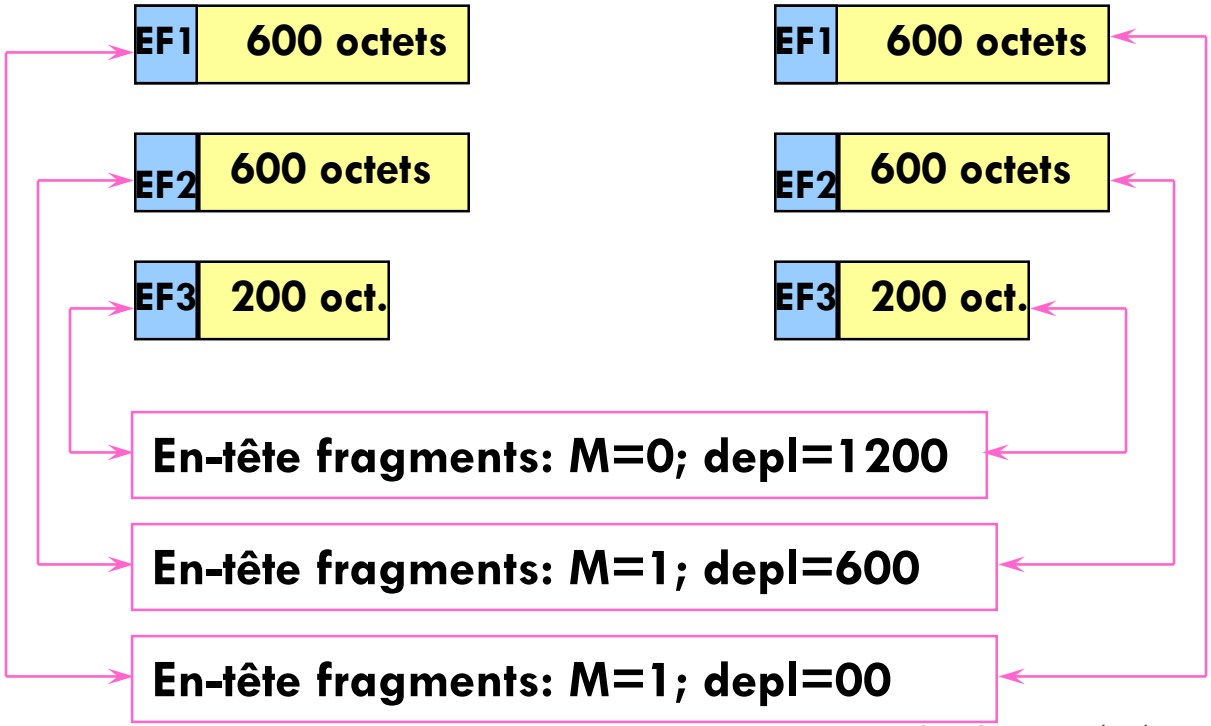
PAT : Port Address Translation → 1 IP Globale pour 2 machines internes

Datagramme IPv4



Fragmentation des datagrammes

En-tête datagramme



EF1 et EF2 ont le bit More (M) positionné.
Le déplacement (depl) est relatif au datagramme initial.

Routage des datagrammes

- Les routeurs forment une structure coopérative : Le datagramme transite de passerelle en passerelle jusqu'à ce que l'une d'entre elles le délivre à son destinataire
- Les routeurs possèdent deux ou plusieurs connexions réseaux
- Les machines participent au routage :
  - les machines déterminent si le datagramme doit être délivré sur le réseau physique sur lequel elles sont (routage direct) ou bien si le datagramme doit être acheminé vers une passerelle (routage indirect)
  - les passerelles effectuent le choix de routage vers d'autres passerelles afin d'acheminer le datagramme vers sa destination finale en fonction de l'IP_{Dest}
- ARP : Récupération de l'adresse physique (MAC) à partir de l'adresse IP via une requête Broadcast (pour routage direct)