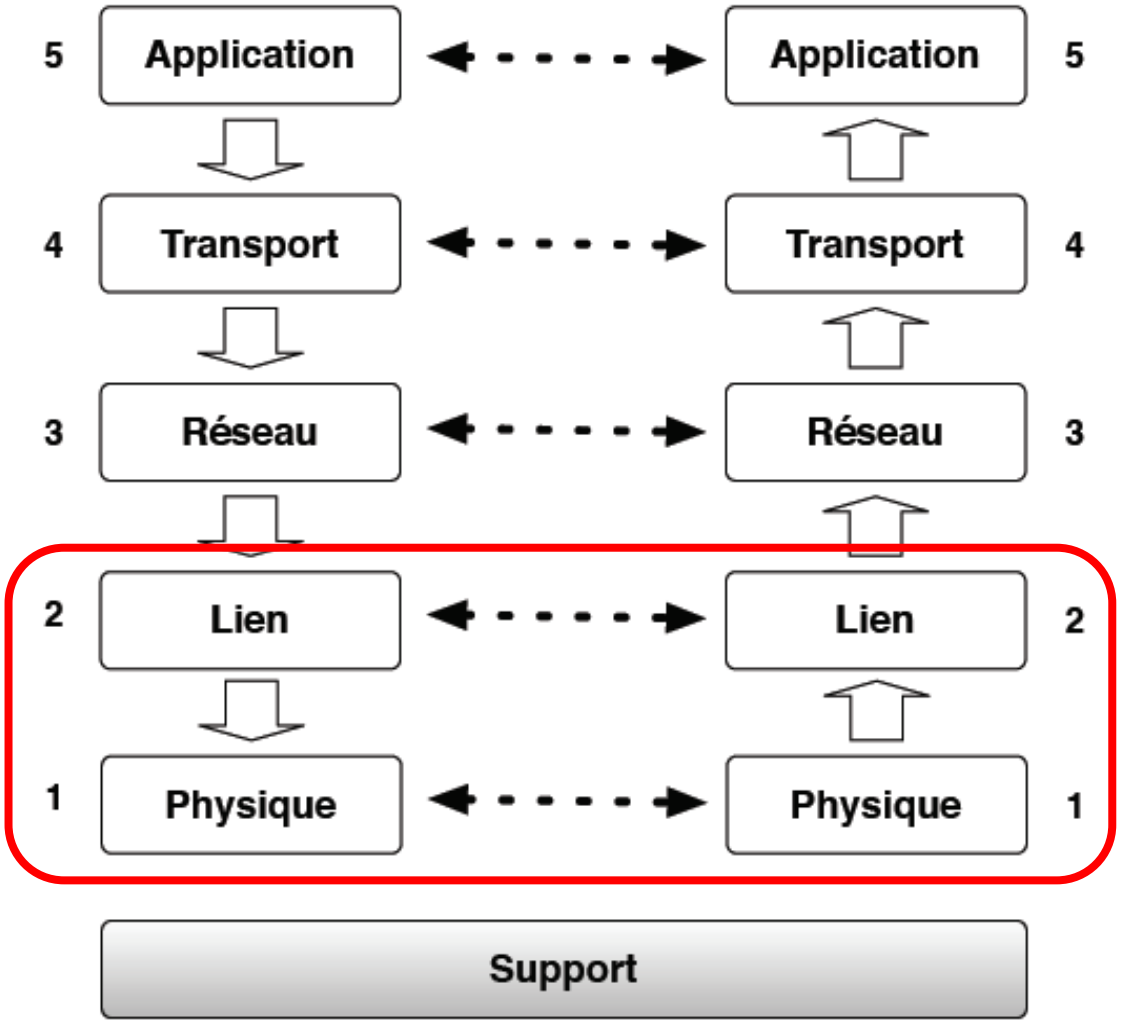
ISN : routage et transport

Couche réseau IP
Couche Transport TCP, UDP

Rappels



Transmet et reconstitue le fichier

Transmet et reconstitue les paquets

Transmet et reconstitue les datagrammes dans le réseau global

Transmet et reconstitue les trames sur le réseau local

Transmet et reconstitue les données sur le support physique

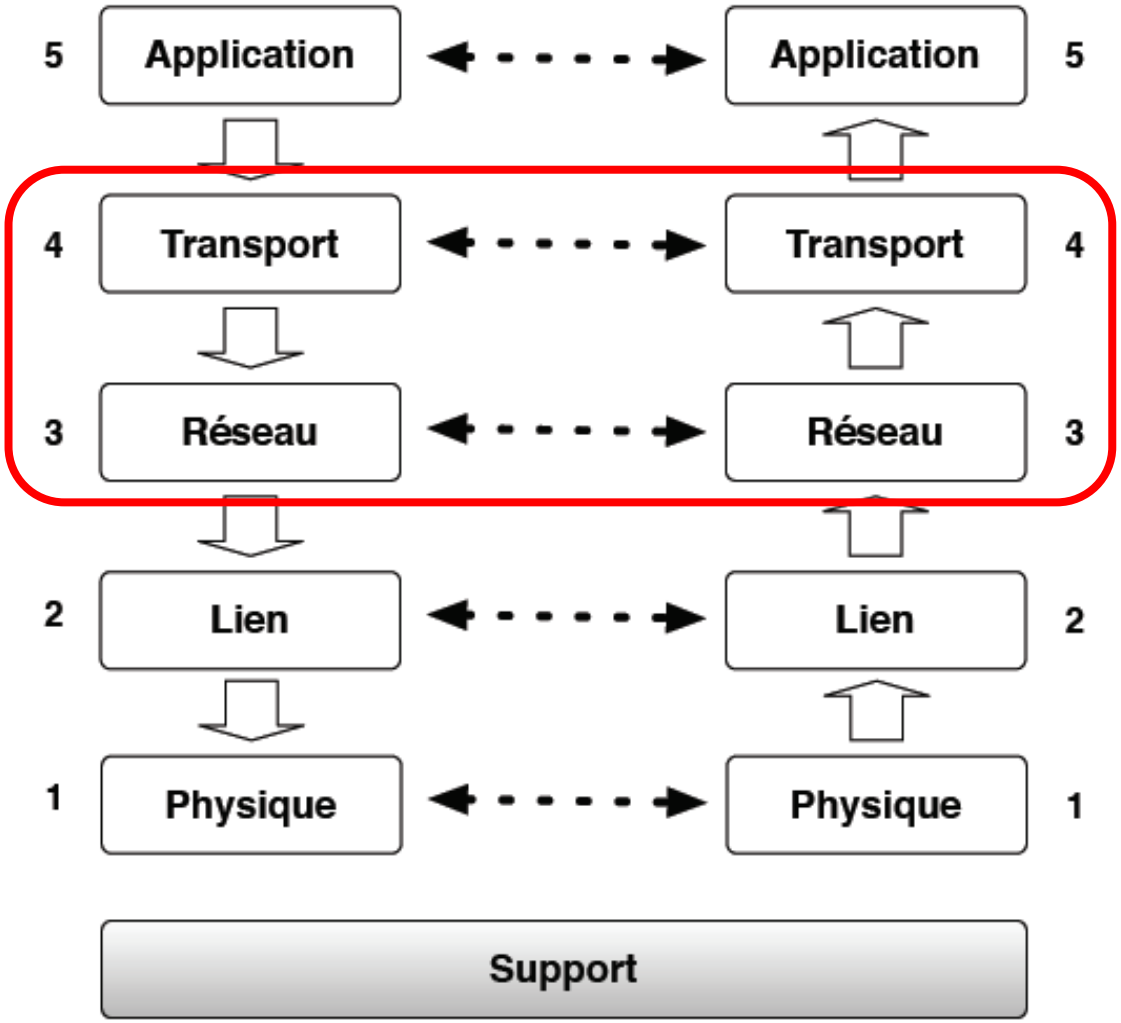
Rappels : la couche physique (niv 1)

- La couche physique s'occupe de transmettre des données binaires (0 ou 1)
- Différents supports physiques existent (filaire, aérien, optique)
- Il faut un équipement physique spécial pour envoi la donnée et la décoder à la réception (ex: modem)
- Il existe des techniques de codage spécifiques pour pallier aux défauts de transmission

Rappels : la couche liaison de données (niv 2)

- La couche liaison de données transfère des données entre des nœuds sur le même d'un réseau local.
- La couche de liaison de données peut dans certains cas détecter et potentiellement corriger les erreurs qui peuvent survenir au niveau de la couche physique (niv 1)
- Chaque nœud est doté d'un identifiant unique matériel (adresse MAC)
- Un nœud peut envoyer un message à un destinataire en particulier ou diffuser à tous les nœuds

Plan du cours



Transmet et reconstitue le fichier

Transmet et reconstitue les paquets

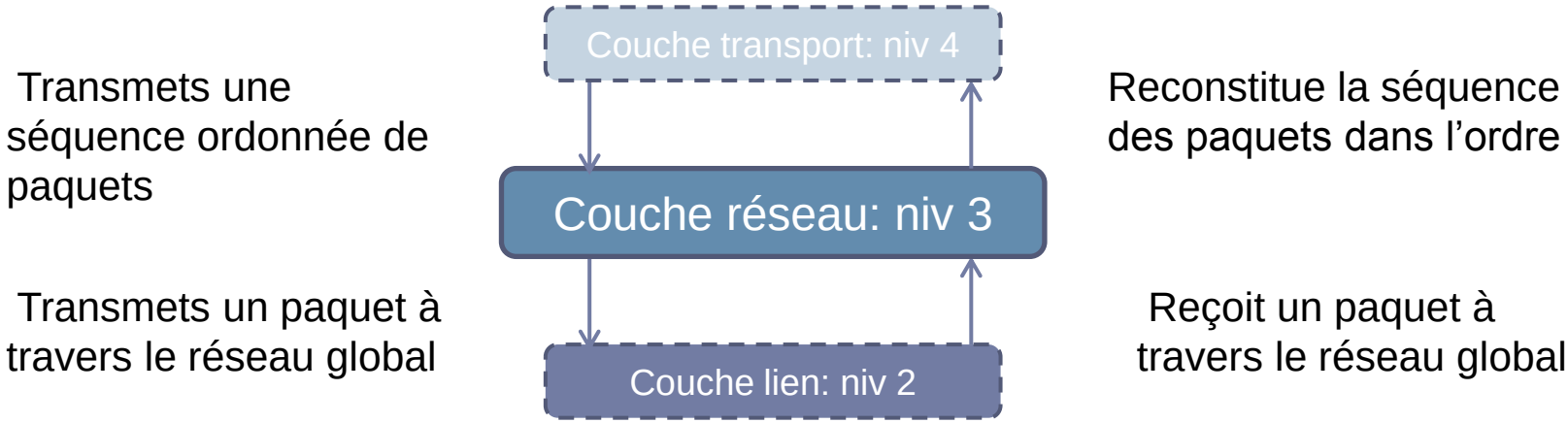
Transmet et reconstitue les datagrammes dans le réseau global

Transmet et reconstitue les trames sur le réseau local

Transmet et reconstitue les données sur le support physique

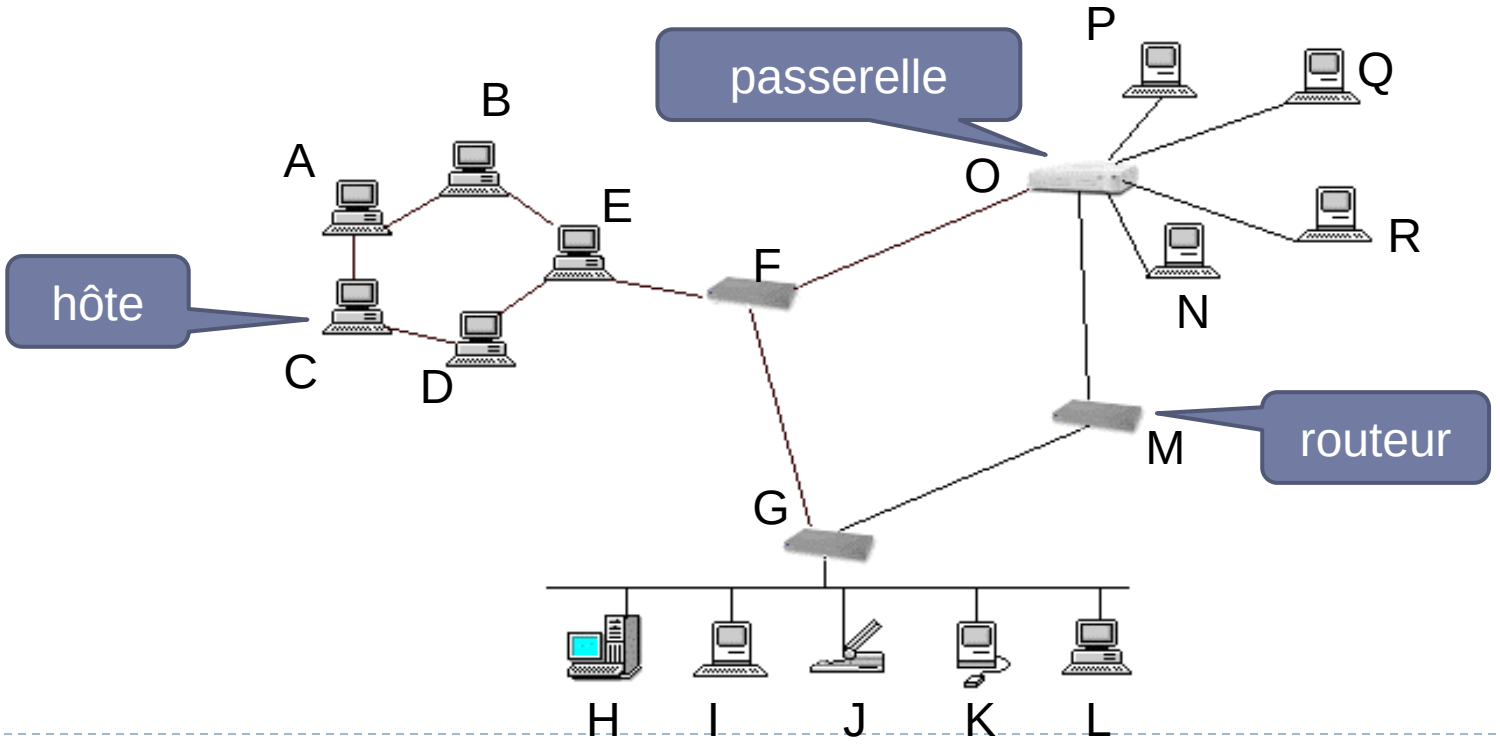
La couche réseau (niv 3)

- Les tâches accomplies par les protocoles de cette couche sont :
  - **Routage** : aiguiller chaque paquet vers sa destination, à chaque embranchement entre différents nœuds et liens rencontrés au cours du périple de ce paquet à travers le réseau.
  - **Adressage** : identifier les ordinateurs connectés sur le réseau.



La couche réseau

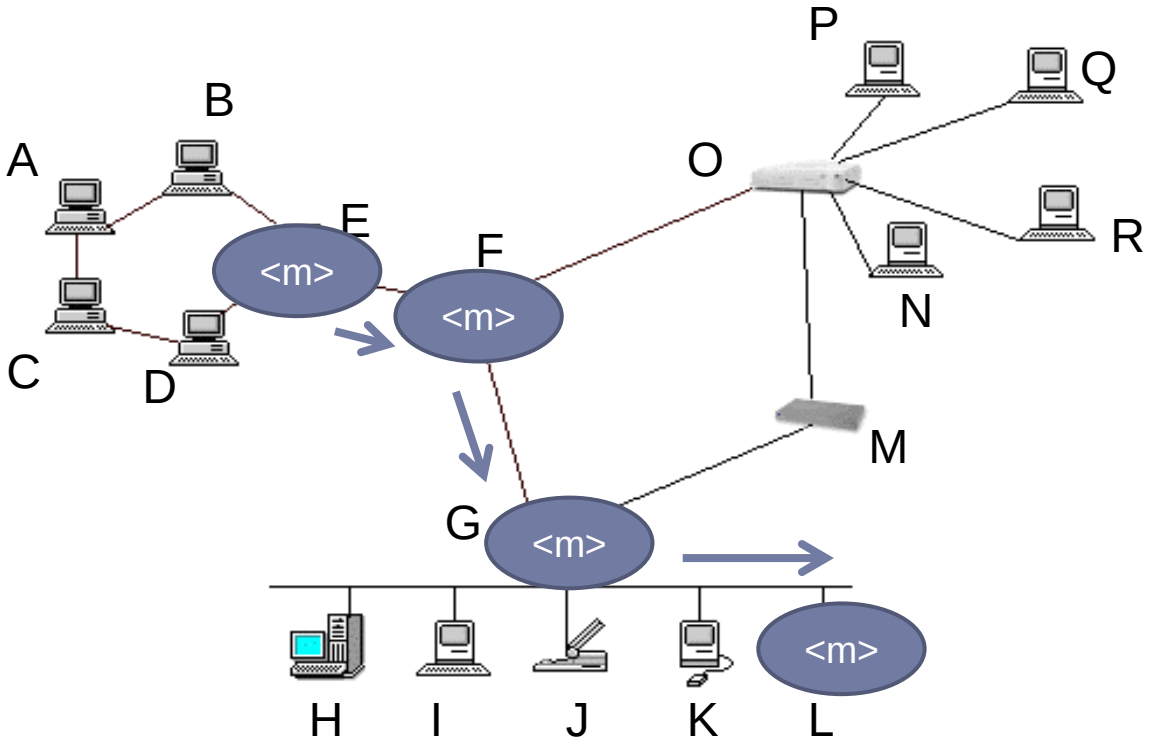
- Chaque machine étant identifiée de façon unique, le routage consiste à acheminer les paquets d'un nœud émetteur A vers un nœud destinataire B en passant par les autres nœuds du réseau.



Le routage : concepts

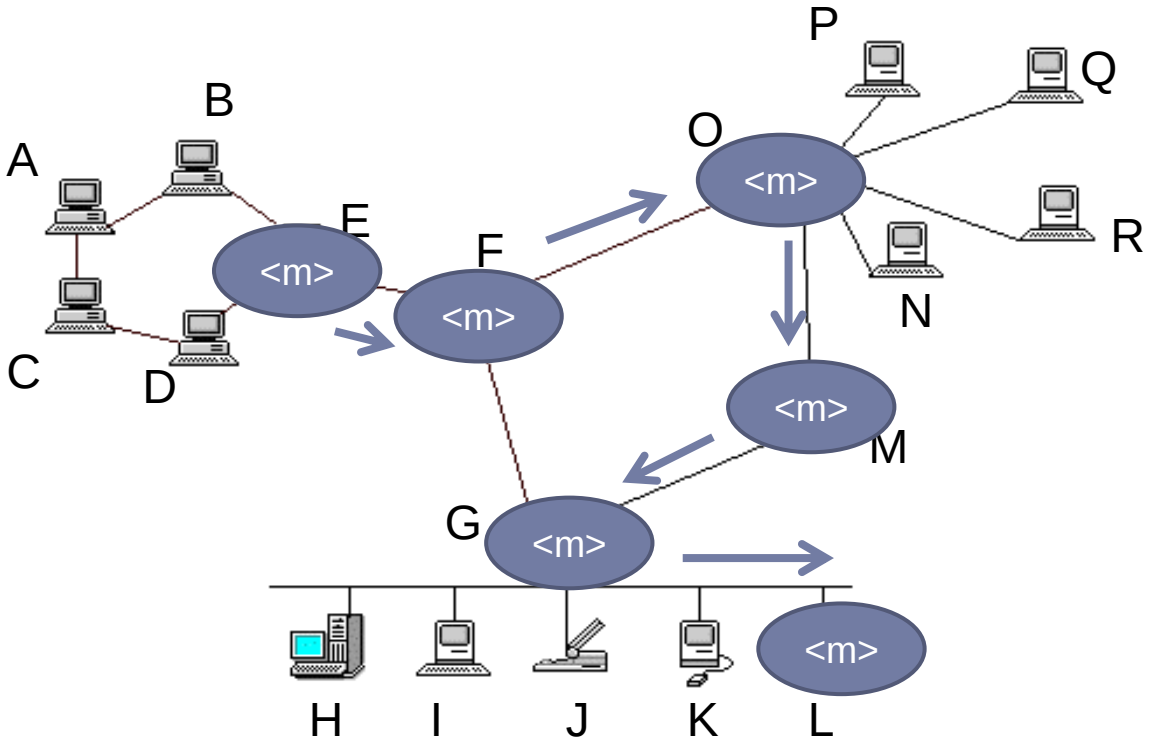
- Hôte
  - émet ou reçoit les messages
- Routeur
  - sert d'intermédiaire dans la transmission d'un message
- Passerelle
  - routeur qui se trouve entre deux réseaux dépendant d'autorités différentes, comme entre le réseau local d'une entreprise et l'Internet

Le routage : exemple



- E veut envoyer un paquet <m> à L
- <m> passe successivement par E, F, G et L
- <m> a fait 3 bonds

La couche réseau



- E veut envoyer un paquet <m> à L
- <m> passe successivement par E, F, O, M, G et L.
- <m> a fait 5 bonds

Le routage

- Protocole de routage
  - mécanisme par lequel des chemins sont sélectionnés dans un réseau pour acheminer les données d'un expéditeur jusqu'à un ou plusieurs destinataires.
  - Il y a plusieurs routes possibles d'un hôte à un autre
  - Le routage doit tenir compte des pannes éventuelles de routeur
- Types de routage
  - Unicast : une seule destination
  - Multicast : à un groupe de machines
  - Anycast : à un seul membre d'un groupe, généralement le plus proche
  - Broadcast : à toutes les machines

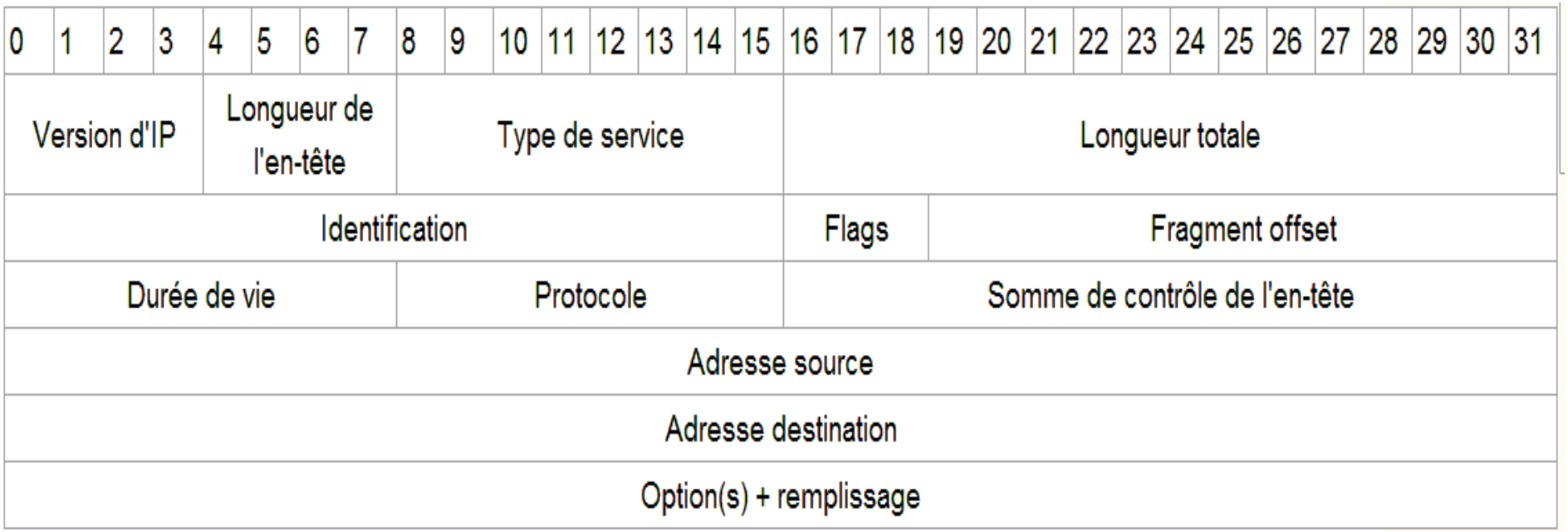
La couche réseau

- Adressage
  - Comment identifier de façon unique une multitude d'équipements (plusieurs milliards) ?
  - Il faut un encodage suffisamment grand mais pas trop (surchage chaque paquet)
  - Avec n bits, on identifie 2^n machines
    - n=32 2^{32}=4 294 967 296 ≈ 4,3Ma identifiants
- Routage
  - Pour acheminer un paquet de A vers B, quelles informations doit connaître chaque nœud ?
  - Comment faire si un nœud tombe en panne ?
- Protocole IP (Internet Protocol)

Adressage

- Couche 2 (lien) : les adresses MAC
  - Ex : 10:93:e9:0a:42:ac
  - ne sont valables que localement
  - peuvent avoir des formats différents selon le support physique
- Couche 3 (réseau) : les adresses IP
  - format d'adresse indépendant des protocoles utilisés à la couche lien
  - valables à travers tout le réseau global
  - 32 bits = 4,3Ma identifiants
  - souvent notée sous forme de 4 mots de 8 bits X.X.X.X
  - X compris entre 0 et 28=255
  - exemples:
    - 216.239.59.104
    - 127.0.0.1

Format d'un datagramme/paquet IP v4



Format d'un datagramme/paquet IP v4

- **Version**
  - 4 bits
  - v4 ou v6 (autre format pour IPv6)
- **Internet Header Length (IHL)**
  - 4 bits
  - Longueur de l'en-tête Internet, exprimée en mots de 4 octets (*bourrage* ou *padding* avec des 0 pour alignement).
  - Taille maximum de l'entête IP =15*4 = 60 octets.
- **Type Of Service (TOS)**
  - 1 octet
  - Indications sur son niveau de priorité et sa classe de service.
- **Total Length**
  - 2 octets
  - Longueur totale du datagramme, exprimée en octets, en-tête et données comprises.
  - Ce champ étant codé sur 2 octets, la longueur maximale d'un paquet IP est donc de 65 536 octets (0 à 65535)

Format d'un datagramme/paquet IP v4

- **Identification, Flags et Fragment Offset**
  - 4 octets est réservé à la gestion de la fragmentation
  - Un datagramme IP peut être fragmenté au niveau 2 en raison de sa taille.
  - Ethernet : MTU=1500 octets de données
- **Time To Live**
  - Durée de vie du datagramme. Cette valeur est décrémentée toutes les secondes ou à chaque passage dans une passerelle. Si cette valeur est à 0, le datagramme est mis au rebut.
- **Protocol**
  - 1 octet
  - identifie le protocole de niveau supérieur transporté dans le champ de données du paquet IP (TCP, UDP, ICMP,…)

Format d'un datagramme/paquet IP v4

- **Checksum**
  - Champ de contrôle d'erreur, calculé uniquement sur l'en-tête.
  - Le principe consiste à faire la somme des valeurs des octets de l'entête et à inscrire le résultats dans l'octet de checksum. Le récepteur effectue la même opération, si la valeur trouvée est identique, il n'y a pas d'erreur. Dans le cas contraire, le paquet est rejeté.
  - Le routeur n'informe personne et ne cherche pas à générer une répétition du paquet (réémission par l'émetteur). Les couches supérieures devront gérer elles-mêmes cette perte de paquet est s'occuper des demandes de réémissions éventuelles.
  - Recalculé à chaque passage dans un équipement traversé par le datagramme
- **Adresses destination et source**
  - Adresse ipv4 sur 4 octets
- **Options**

A retenir

- La couche réseau (IP) est responsable du
  - Routage : acheminer les paquets IP d'un émetteur vers un destinataire en se propageant d'un nœud à l'autre.
  - Adressage : les nœuds sont identifiées par une adresse IP sur 4 octets notée X.X.X.X
  - Les paquets (datagrammes) IP contiennent les adresses émetteur et destinataire.
- La couche réseau n'est pas responsable
  - D'assurer la fiabilité de la transmission: certains paquets peuvent être perdus ou dupliqués.
  - De reconnaître les différentes applications qui s'exécutent sur une machine

Routage

ISN : Réseau et Transport

Principe du routage

- Du point de vue de la couche réseau, la destination d'un paquet est un ordinateur connecté à Internet, identifié donc par l'**adresse IP du destinataire**, qui figure donc dans l'en-tête des paquets envoyés par la couche réseau.
- L'en-tête de ces paquets contient de plus **l'adresse IP de l'ordinateur émetteur** du paquet.
- Pour trouver son chemin à travers le réseau de câbles et de liens radio connectant les ordinateurs entre eux, jusqu'à une destination identifiée par son adresse IP, il faut faire appel à un type de protocole supplémentaire, faisant également partie de la couche réseau : **un protocole de routage**.

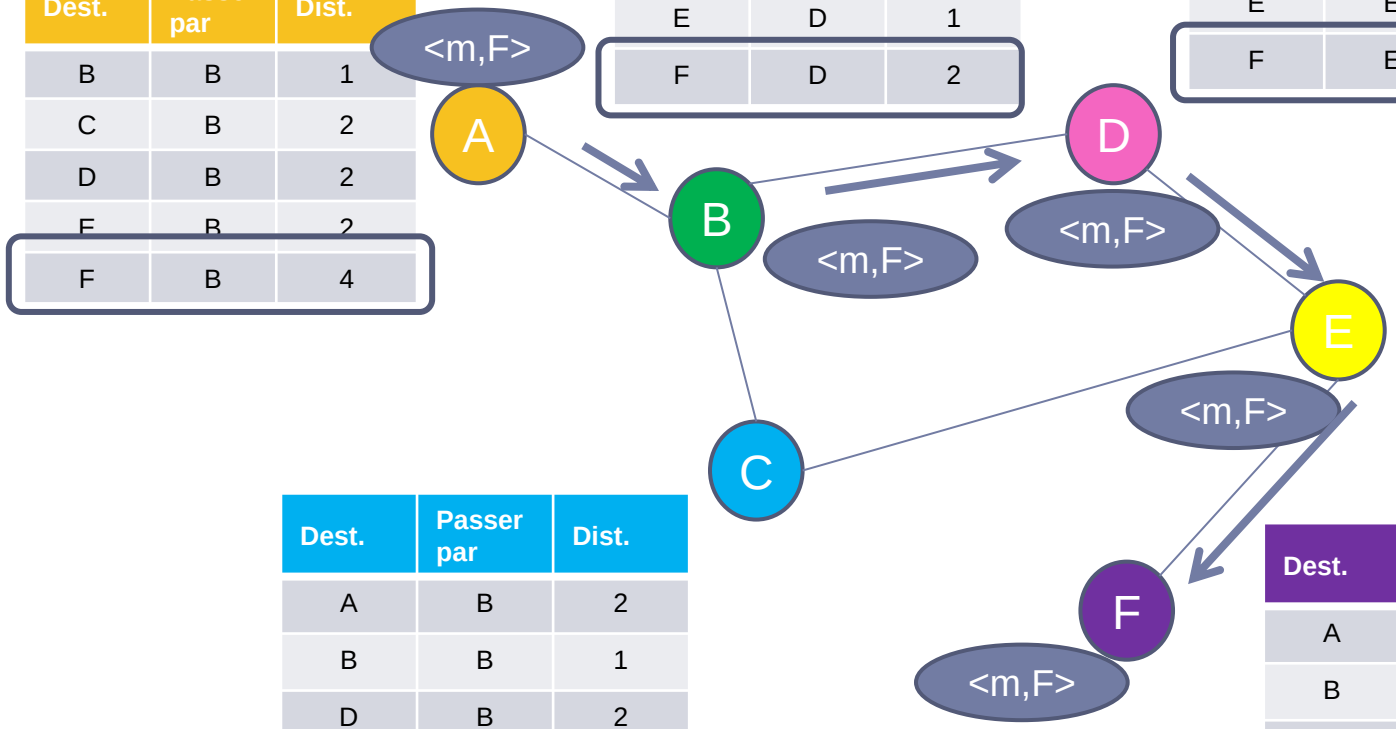
Principe du routage

- Une table de routage est une sorte de "panneau indicateur" qui donne les routes (les réseaux) joignables à partir du "carrefour" que constitue un routeur.
- Les paquets arrivent sur une interface de la machine.
- Pour "router" le paquet, le routeur fondera sa décision en deux temps :
  - d'abord il regarde dans l'en-tête IP le réseau de destination et compare toutes les entrées dont il dispose dans sa table de routage
  - ensuite, si le réseau de destination est trouvé, il commute le paquet sur le bon port de sortie
  - si ce réseau n'est pas trouvé, le paquet est jeté

Protocole de routage statique

- Réseau statique
  - Les nœud et les liens sont fixes.
  - Une route statique est une entrée manuelle dans une table de routage.
- Calcul des chemins optimaux
  - On observe les différents chemins possibles
  - On en déduit les chemins les plus courts
  - On les stocke dans une table de routage
- Table de routage
  - Sur chaque routeur, une règle par nœud du réseau
  - Règle : *<pour aller à X, passer par Y, distance d>*

Protocole de routage statique



Dest.	Passer par	Dist.
B	B	1
C	B	2
D	B	2
E	B	2
F	B	4

Dest.	Passer par	Dist.
A	B	2
B	B	1
D	B	2
E	D	1
F	D	2

Dest.	Passer par	Dist.
A	B	2
B	B	1
C	E	2
E	E	1
F	E	2

Dest.	Passer par	Dist.
A	D	3
B	D	2
C	C	1
D	D	1
F	F	1

Dest.	Passer par	Dist.
A	B	2
B	B	1
D	B	2
E	E	1
F	E	2

Dest.	Passer par	Dist.
A	E	4
B	E	3
C	E	2
D	E	2
E	E	1

Protocole de routage statique

- Avantages
  - Simple
- Inconvénients
  - Si l'on ajoute ou enlève des ordinateurs ou si on modifie les liens entre eux, il faut reprogrammer tous les ordinateurs.
  - Ca fait beaucoup d'informations à conserver

Protocole de routage : vecteur de distance

- IGP : Internet Global Protocol
- Basé sur l'algorithme de Bellman-Ford
- Principe
  - Diffusion périodique à tous ses voisins d'un paquet spécial appelé HELLO contenant sa table de routage.
  - La table de routage se remplit au fur et à mesure qu'il reçoit les tables de ses voisins (distance=1)
  - Un routeur entend parler progressivement d'autres routeurs qui ne sont pas ses voisins, mais des voisins de ses voisins — des routeurs notés à distance 2, puis 3, 4, etc. dans les tables de routage des messages HELLO.
  - Il peut ainsi répercuter ces nouvelles informations dans sa propre table de routage

Protocole de routage : vecteur de distance

- Nœud i
- Receive(voisin j, table_routage(j))

  Pour toute entrée (k->route(k,v,n)) Faire
  - // nouvelle destination

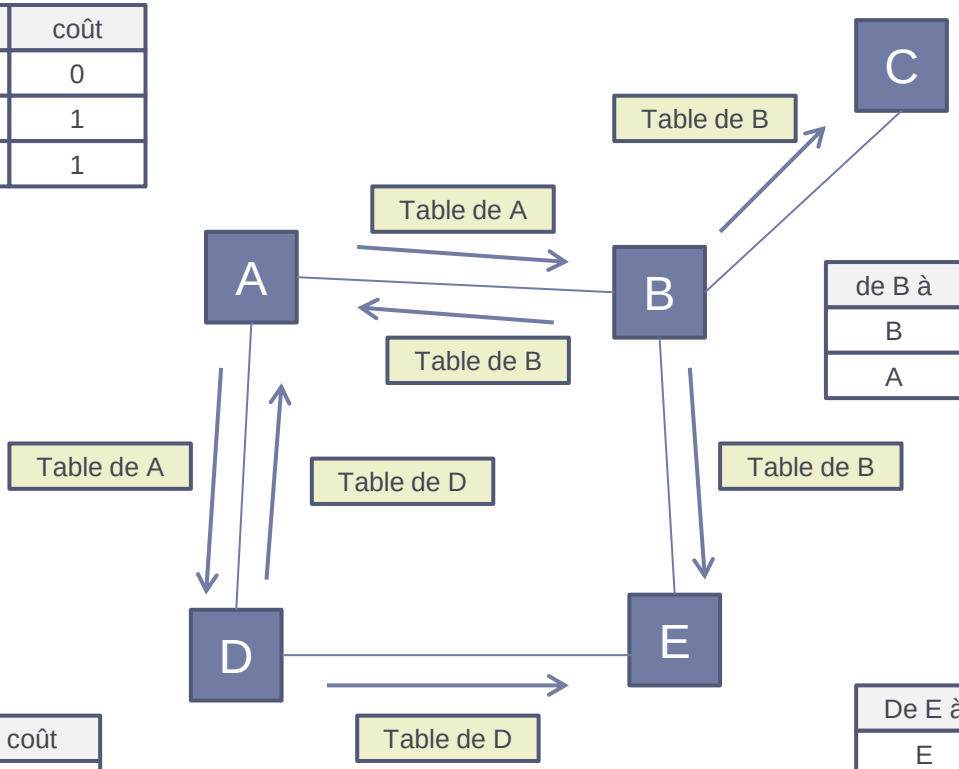
    Si k est une nouvelle destination
    Alors ajouter la nouvelle destination (k->route(k,j,n+1) dans table_routage(i)
  - // route meilleure pour une entrée k existante

    Si (k->route(k,j,n+1)) est une route plus courte que la route actuelle
    Alors modifier la route de k dans table_routage(i)

Exemple : initialisation

de A à	route	coût
A	locale	0
B	B	1
D	D	1

De C à	route	coût
C	locale	0
B	B	1
A	B	2



de B à	route	coût
B	locale	0
A	A	1

de D à	route	coût
D	locale	0
A	A	1

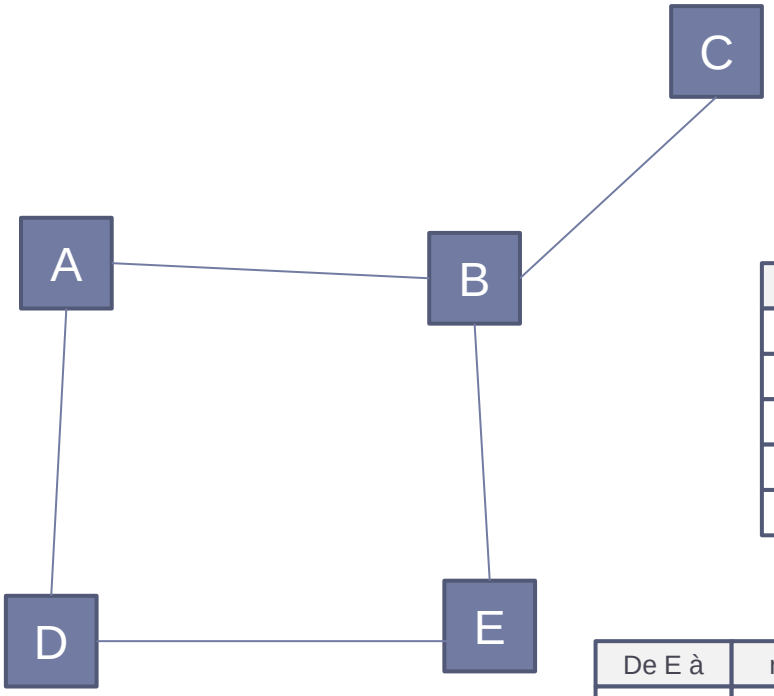
De E à	route	coût
E	locale	0
B	B	1
A	B	2
D	D	1

Ordre de diffusion : A,B,D
E et A reçoivent la table de B avant celle de D

Exemple : convergence

de A à	route	coût
A	locale	0
B	B	1
C	B	2
D	D	1
E	D	2

de C à	route	coût
A	B	2
B	B	1
C	locale	0
D	B	3
E	B	2



de B à	route	coût
A	A	1
B	locale	1
C	C	1
D	E	2
E	E	1

De D à	route	coût
A	A	1
B	E	2
C	E	3
D	locale	1
E	E	1

De E à	route	coût
A	B	2
B	B	1
C	B	2
D	D	1
E	locale	2

Au bout d'un certain temps, les tables convergent.

Exemple : cas de panne d'une liaison

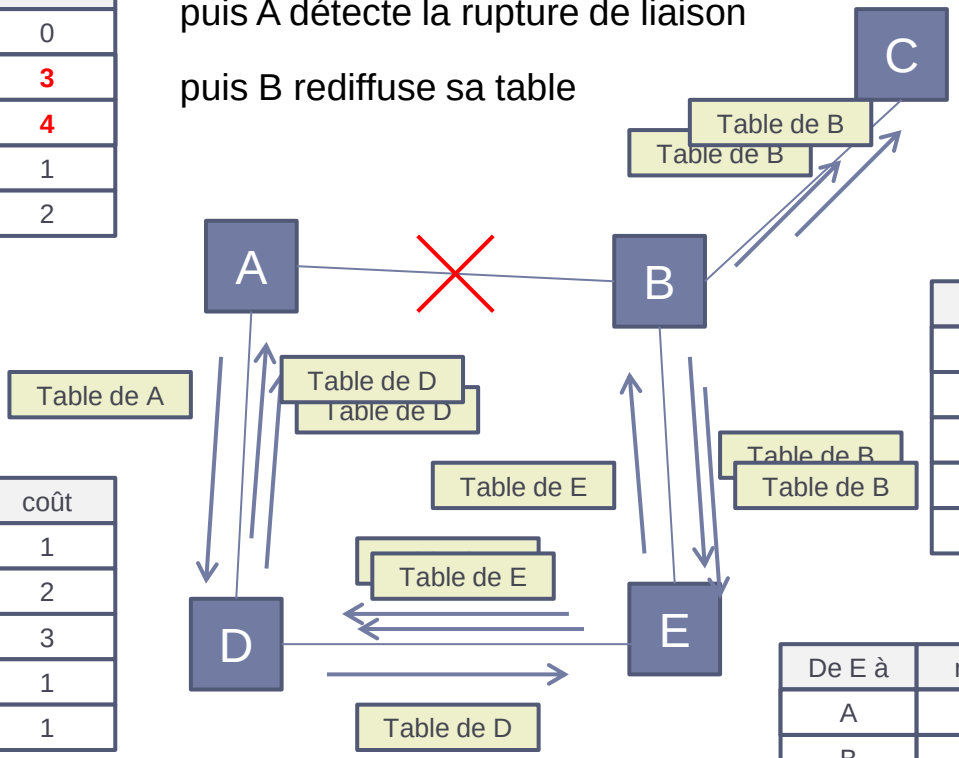
B détecte la rupture de liaison en premier

puis A détecte la rupture de liaison

puis B rediffuse sa table

de A à	route	coût
A	locale	0
B	B	**3**
C	B	**4**
D	D	1
E	D	2

de C à	route	coût
A	**B**	**4**
B	B	1
C	locale	0
D	B	3
E	B	2



de B à	route	coût
A	**E**	**3**
B	locale	1
C	C	1
D	E	2
E	E	1

De D à	route	coût
A	A	1
B	E	2
C	E	3
D	locale	1
E	E	1

D inchangé

De E à	route	coût
A	**D**	**2**
B	B	1
C	B	2
D	D	1
E	locale	2

Si la liaison A-B tombe en panne, A et B modifient leurs tables avec une distance infinie et propagent l'information.

Exemple : cas de boucle

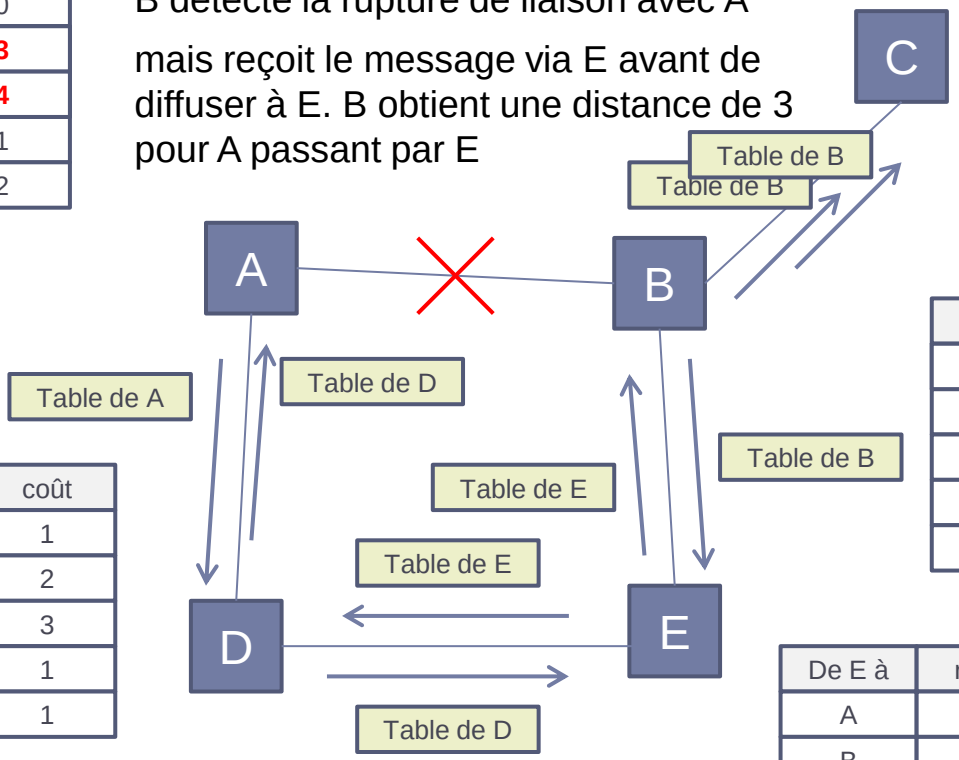
de A à	route	coût
A	locale	0
B	**D**	**3**
C	**D**	**4**
D	D	1
E	D	2

A détecte la rupture de liaison avec B

B détecte la rupture de liaison avec A

mais reçoit le message via E avant de diffuser à E. B obtient une distance de 3 pour A passant par E

de C à	route	coût
A	**B**	**4**
B	B	1
C	locale	0
D	B	3
E	B	2



de B à	route	coût
A	**E**	**3**
B	locale	1
C	C	1
D	E	2
E	E	1

De D à	route	coût
A	A	1
B	E	2
C	E	3
D	locale	1
E	E	1

D inchangé

De E à	route	coût
A	**B**	2
B	B	1
C	B	2
D	D	1
E	locale	2

E inchangé

Tous les messages de C pour A rebondiront entre B et E jusqu'à expiration du TTL

Protocole de routage : vecteur de distance

- Avantages
  - Simple
  - Routeurs autonomes
- Limitations
  - Convergent lentement.
  - Dans certains cas le protocole dérègle durablement les tables de routage en ne détectant pas correctement qu'un ou plusieurs routeurs sont soudainement devenu hors-service
  - Sensible aux boucles
  - La TTL ou durée de vie du paquet est décrémentée à chaque saut. Le paquet est supprimé quand son TTL arrive à 0.
  - De nombreux algorithmes de routage existent.

A retenir

- Les protocoles de routage ont pour fonction de décider sur quel lien transmettre un paquet IP.
- Chaque nœud maintient une table de routage contenant des règles de la forme
  - *<pour aller à X, passer par Y, distance d>*
- La route qu'emprunte un paquet peut être déterminée
  - Statiquement : chaque route est entrée manuellement par l'administrateur réseau.
  - Dynamiquement : les nœuds découvrent incrémentalement leurs voisins puis le reste du réseau en échangeant leur table de routage. Détecte les pannes.

Adressage IP

L’adressage IP

- Fonction routage dans un réseau = transmettre de routeur en routeur les paquets en fonction de leur *adresse IP*.
- **Adresse IP** = identifiant unique valable dans tout le réseau global.
- Un paquet IP véhicule simultanément deux adresses:
  - **L'adresse source** qui est l'adresse IP de la machine qui a formaté et qui émet le paquet IP
  - **L'adresse destination**, qui est inscrite par la machine émettrice, et qui correspond à l'adresse de la machine pour qui est destiné le paquet IP.

L’adressage IP : adresse d'interface

- Une petite subtilité doit être précisée. En fait, **IP n'affecte pas une adresse à une machine, mais à l'interface d'une machine**.
- Une machine qui dispose de plusieurs interfaces raccordées à un réseau (un routeur par exemple) est dotée de plusieurs adresses IP, une par interface.
  - Ex: 127.0.0.1 adresse de rebouclage
- Ce n'est pas le cas d'autres protocoles comme DRP de Decnet Phase IV, par exemple, qui affecte une seule adresse à une machine quel que soit son nombre d'interfaces réseaux.
- Retenons donc **qu'IP fixe une adresse à une interface réseau et pas à une machine**.

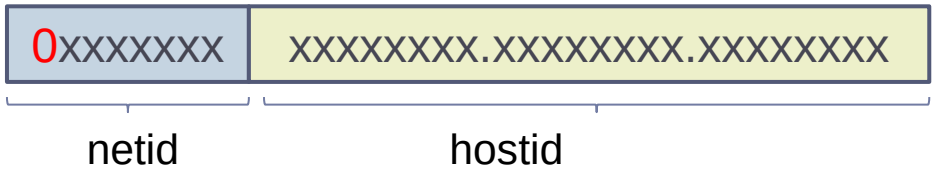
Adresse IP

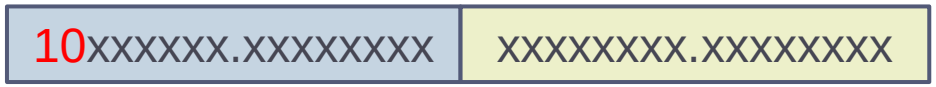
- Adresse IPv4
  - Adresse de 32 bits
  - Découpé en 4 mots de 8 bits avec notation pointée X.X.X.X
  - X = de 0 à 255
- Exemples:
  - 216:239:59:10
  - 127.0.0.1
- 2^32 = 4 milliards d'adresses possibles
- En février 2011, la réserve d'adresse IPv4 est épuisée
- Adresse IPv6 sur 128 bits
  - 667 millions de milliards d'adresses par millimètre carré de surface terrestre

Adresse réseau et adresse machine

- H**iérarchie** définie dans le format d'adresse IP.
- Une adresse IP est scindée en deux parties :
  - **Le préfix réseau (netid)** : il identifie un groupe de machine, généralement regroupées sur un même sous réseau physique (Ethernet, Token Ring, X25, etc.)
  - **l'adresse machine (hostid)** : elle identifie la machine dans le sous-réseau considéré.
- Le préfixe réseau
  - Statique : au début, les adresses IP étaient classées en plusieurs catégories appelées « classes »
  - Dynamique : aujourd'hui, il faut préciser le préfix réseau associé à chaque adresse IP

Préfixe réseau statique : les classes d'adresse

- Les adresses IP sont regroupées en classe
  - Classe A
    - 126 réseaux, 16777214 hôtes max par réseau
    - 

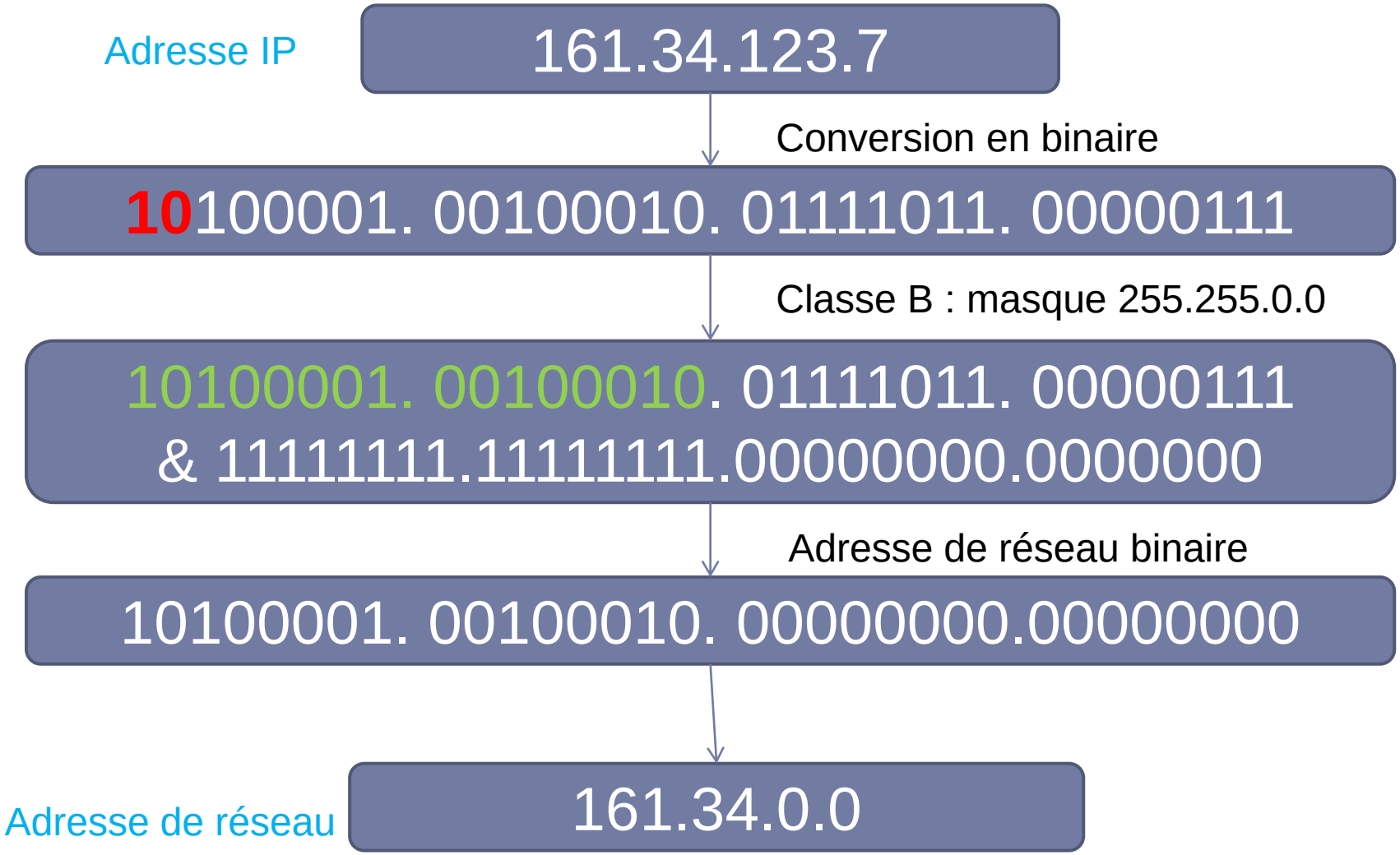
  - Classe B
    - 16384 réseaux, 65534 hôtes max par réseau
    - 

  - Classe C
    - 2097152, 254 hôtes max par réseau
    - 

Masque de réseau

- Masque
  - Un masque contient 1 aux emplacements des bits que l'on désire conserver, et des 0 pour ceux que l'on veut annuler.
  - Une fois ce masque créé, il suffit de faire un ET logique entre la valeur que l'on désire masquer et le masque afin de garder intacte la partie que l'on désire et annuler le reste.
- Le masque de réseau standard
  - tous les bits de réseau d'une adresse placés à '1', et tous les bits d'interface placés à '0'.
  - masque de réseau de classe A: 255.0.0.0
  - masque de réseau de classe B: 255.255.0.0
  - masque de réseau de classe C: 255.255.255.0
  - Le masque de réseau (statique) peut être déduit de la classe d'adresse.

Adresse réseau



Adresse de réseau, diffusion et hôte

- Adresse d'un réseau IP.
  - Tous les bits *hostid* positionnés à 0.
  - Ex: 125.0.0.0
- Adresse de diffusion d'un réseau IP
  - Tous les bits *hostid* positionnés à 1.
  - Ex: 125.255.255.255
- Adresse d'un hôte
  - N'importe quelle valeur pour les bits *hostid*, sauf tous à 0 ou tous à 1.
  - Ex : 125.98.43.1

Adresses spéciales

- Route par défaut
  - 0.0.0.0
- Le réseau de boucle de retour - loopback
  - 127.0.0.0
- Les réseaux privés 'non-connectés'
  - Réseaux qui utilisent IP mais ne sont pas connectés à l'Internet.
  - Un réseau de classe A: 10.0.0.0
  - 16 réseaux de classe B: 172.16.0.0 - 172.31.0.0
  - 256 réseaux de classe C: 192.168.0.0 - 192.168.255.0
    - 192.168.0.0 sont des réseaux privés (non routables) utilisés pour les réseaux personnels ou les réseaux internes de petites entreprises.
    - Par défaut, les routeurs des FAI (FreeBox, LiveBox, sfrBox, etc.) créent un réseau de type 192.168.0.0, avec 192.168.0.1 comme IP pour le routeur (passerelle). Il affecte ensuite aux machines du réseau les adresses 192.168.0.2 à 192.168.0.254.

Sous-réseau de taille variable

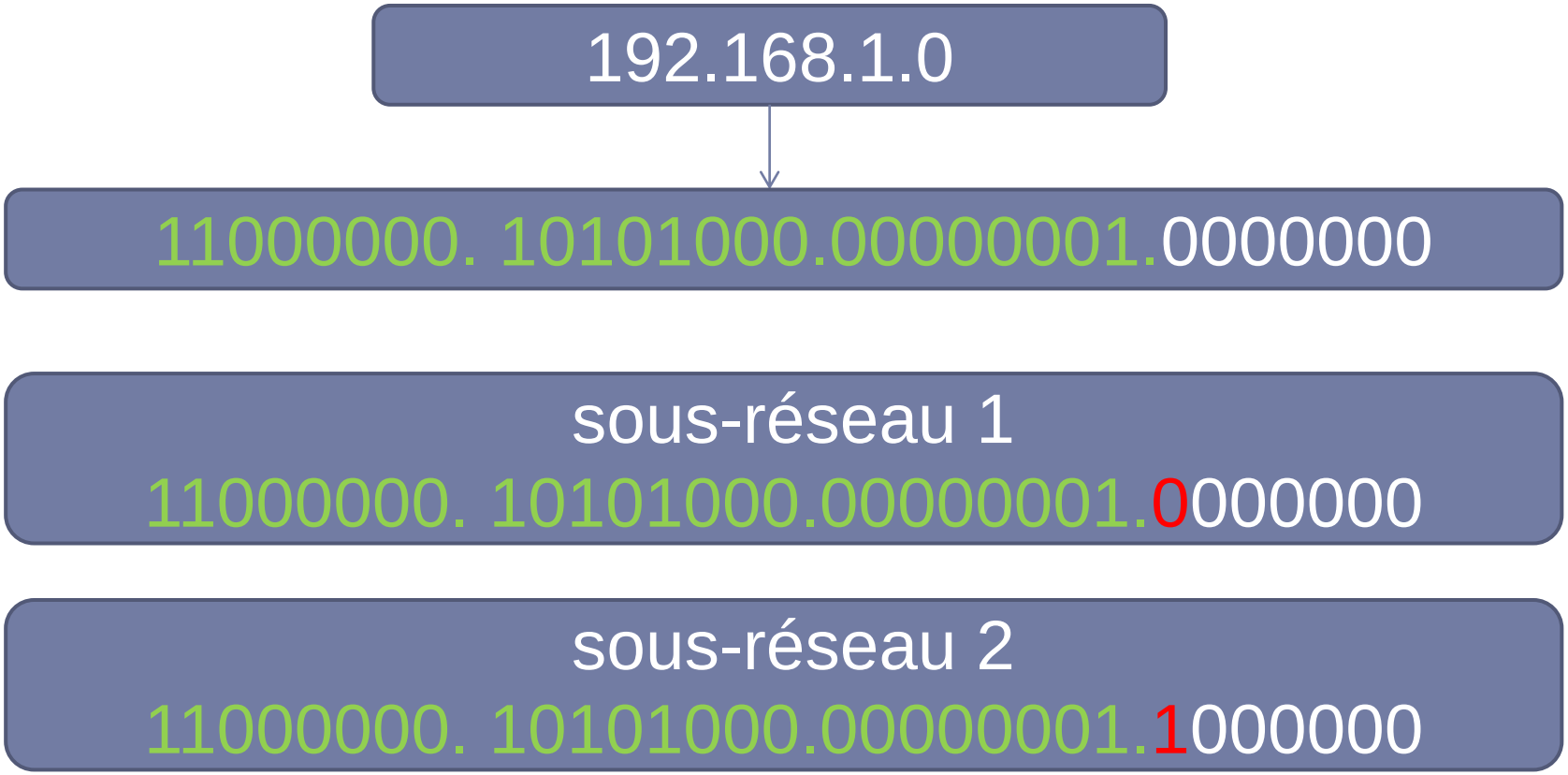
- La notion de classe est obsolète depuis les années 90.
- Elle conduisait au gaspillage des adresses IPv4
- Un réseau peut être localement découpé en sous-réseaux logiques
- L'adresse du réseau est découpé en plusieurs adresses de sous-réseaux.
  - Un sous-réseau correspond typiquement à un réseau local sous-jacent.
- Le *masque de sous-réseau* permet de distinguer la partie de l'adresse utilisée pour le routage et celle utilisable pour numéroter des interfaces.

Masque de sous-réseau

- Pour mettre en œuvre le découpage en sous-réseaux, on réserve un ou plusieurs bits parmi les bits de poids forts d'interface/d'hôte, et on les interprète localement comme faisant partie des bits de réseau
- Un **masque de sous-réseau** (*netmask*) est un masque indiquant le nombre de bits d'une adresse IP utilisés pour identifier le sous-réseau, et le nombre de bits caractérisant les hôtes
- Il indique aussi le nombre d'hôtes possibles dans ce sous-réseau.
- Le masque de sous-réseau ne peut pas être déduit : il doit être transmis en plus de l'adresse IP !

Découpage en sous-réseau

- Exemple : on veut découper en 2 le réseau 192.168.1.0 (classe C) en 2 sous-réseaux même taille (2^7-2=126)



Masque de sous-réseau

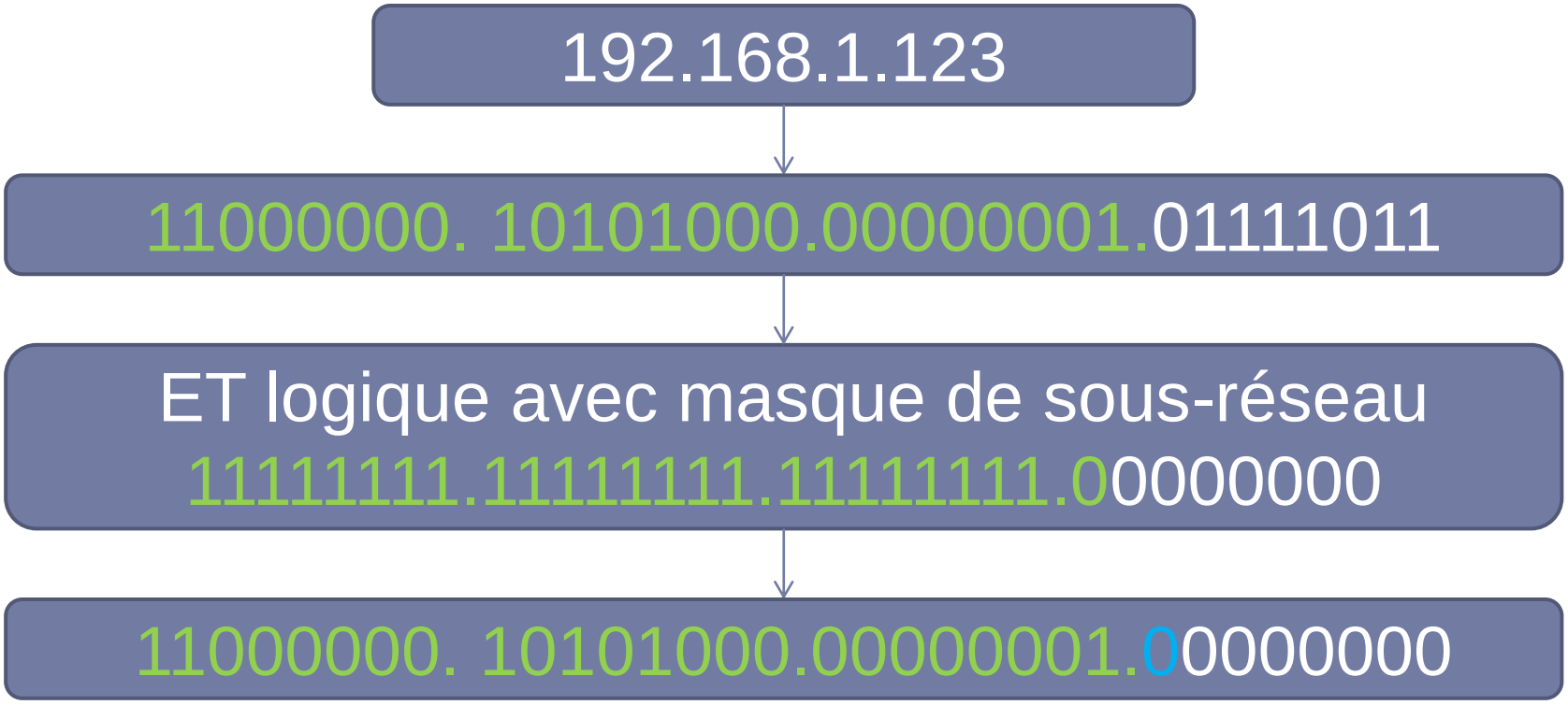
- Le masque de sous-réseau est obtenu en mettant tous les bits du sous-réseau à 1 et le reste à 0.
- Exemple du découpage de 192.168.1.0 en 2 sous-réseaux (un seuil bit du *hostid* réservé pour chaque sous-réseau)

masque des 2 sous-réseaux
11111111.11111111.11111111.10000000
255.255.255.128

- Pour savoir à quel sous-réseau appartient une adresse IP, on effectue un ET logique entre le masque et l'adresse. Le résultat est l'adresse du sous-réseau recherché.

Masque de sous-réseau

- Exemple : on veut tester à quel sous-réseau appartient 192.168.1.123



192.168.1.123 appartient au sous-réseau 1

Sous-réseaux

- Le découpage en sous-réseaux est une configuration locale et invisible au reste du monde.
  - Du point de vue du monde extérieur aux machines et réseaux physiques couverts par le réseau découpé en sous-réseaux, absolument rien n'a changé - cela reste un unique réseau IP.
- Notation CIDR (*Classless Inter-Domain Routing*)
  - Cette notation donne le numéro du réseau suivi par une barre oblique (ou *slash*, « / ») et le nombre de bits à 1 dans la notation binaire du masque de sous-réseau.
  - Le masque 255.255.255.128, équivalent en binaire à 11111111.11111111.11111111.10000000, sera donc représenté par ***/25*** (25 bits à la valeur 1).
- Prefix IP
  - On appelle prefix IP le masque du sous-réseau en notation CIDR.

AS, BGP et prefix IP

- **Domaine de routage autonome AS**
  - AS = Autonomous System
  - Sorte de grand réseau identifié sous une même autorité administrative
  - Environ 40000 identifiés par un numéro ASxxx
  - Les AS sont tous interconnectés
- **Border Gateway Protocol**
  - Protocol de routage inter-AS
  - Un message doit contenir l'adresse IP et son prefix IP

BGP

BGP fonctionne différemment d'un protocole à vecteur de distance

- Pas de transmission périodique des meilleures routes
  - uniquement des modifications
- Mémorise toutes les routes vers toutes les destinations
  - Récupération rapide lorsque une destination devient inaccessible par la route initialement choisie
- Construit des routes sans boucle
  - Le chemin suivi est décrit explicitement à l'aide de la liste des AS traversés
  - Les boucles sont facilement détectées

Résolution d'adresse

- Problème 1 : translation d'adresse
  - Couche 3, identifiant = adresseIP (logique)
  - Couche 2 , identifiant = adresse MAC (matérielle)
- Problème 2 : adresse IP variable
  - l'adresse IP d'un ordinateur peut changer au cours du temps (déménagement, DHCP)
  - Il n'y a pas de lien fixe entre l'adresse MAC correspondant à une adresse IP donnée.

Résolution d'adresse : protocole ARP

- Adresse Resolution Protocol
  - ARP gère une table de correspondance entre les adresses logiques et les adresses physiques dans une mémoire cache.

1. Lorsqu'une machine doit communiquer avec une autre, elle consulte la table de correspondance.
2. Si jamais l'adresse demandée ne se trouve pas dans la table, le protocole ARP émet une requête sur le réseau en broadcast.
3. L'ensemble des machines du réseau vont comparer cette adresse logique à la leur. Si l'une d'entre-elles s'identifie à cette adresse, la machine va répondre à l'émetteur en lui retournant son adresse MAC
4. La machine reçoit la réponse et stocke le couple d'adresses dans la table de correspondance
5. La communication va alors pouvoir avoir lieu...

Internet = réseau virtuel

- Le réseau de câbles et de liens radio connectant les ordinateurs entre eux peut dès à présent être considéré par l'émetteur d'un paquet comme une sorte de câble virtuel le connectant au destinataire du paquet, qui lui permet de communiquer avec la destination de la même manière que si un même câble les connectait directement, même si en réalité, la connexion est indirecte, à travers des ordinateurs et des liens intermédiaires.
- C'est ce concept de câble virtuel que l'on appelle Internet.
- Dans la suite, on appellera simplement réseau l'abstraction fournie par les protocoles de la couche réseau, permettant donc la transmission de paquets de données à travers une suite de liens.

Internet, mais sans garantie

- les transmissions de paquets sur le réseau ne sont pas garanties d'arriver à destination ou même dans l'ordre
  - Les paquets peuvent emprunter des routes différentes
- Un paquet peut être perdu par la couche lien ou par la couche réseau
- C'est le rôle de la couche transport d'assurer la fiabilité de la communication

A retenir

- Les adresses IP sont des adresse logiques qui identifient des machines interconnectées via internet
- Les routeurs examinent l'adresse du destinataire pour déterminer sur quel lien transmettre le message
- Un routeur ne conserve que les adresses des réseaux dans ses tables et non de tous les hôtes.
- Il utilise le masque de réseau fourni dans le message pour déterminer sur l'adresse réseau associée au destinataire
- Localement, la passerelle peut router les paquets sur des sous-réseaux locaux via une table de routage locale qui stocke les adresses des sous-réseaux.
- Les sous-réseaux sont invisibles à l'extérieur de la passerelle.

Couche Transport

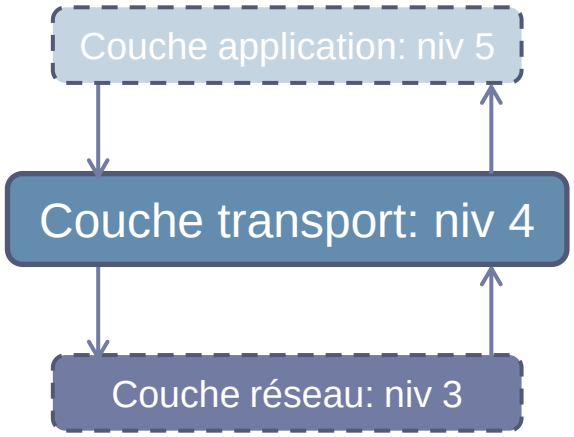
La couche transport (niv 4)

- Les tâches accomplies par les protocoles de cette couche sont :
  - **Adressage** : identifier les *applications* en cours d'exécution qui utilisent le réseau.
  - **Fiabilité** : assurer le transport des paquets de bout en bout en corrigeant les erreurs éventuelles de la couche 3.

Transmets des données applicatives de taille variable

Transmets une séquence ordonnée de paquets

Transmets un paquet à travers le réseau global



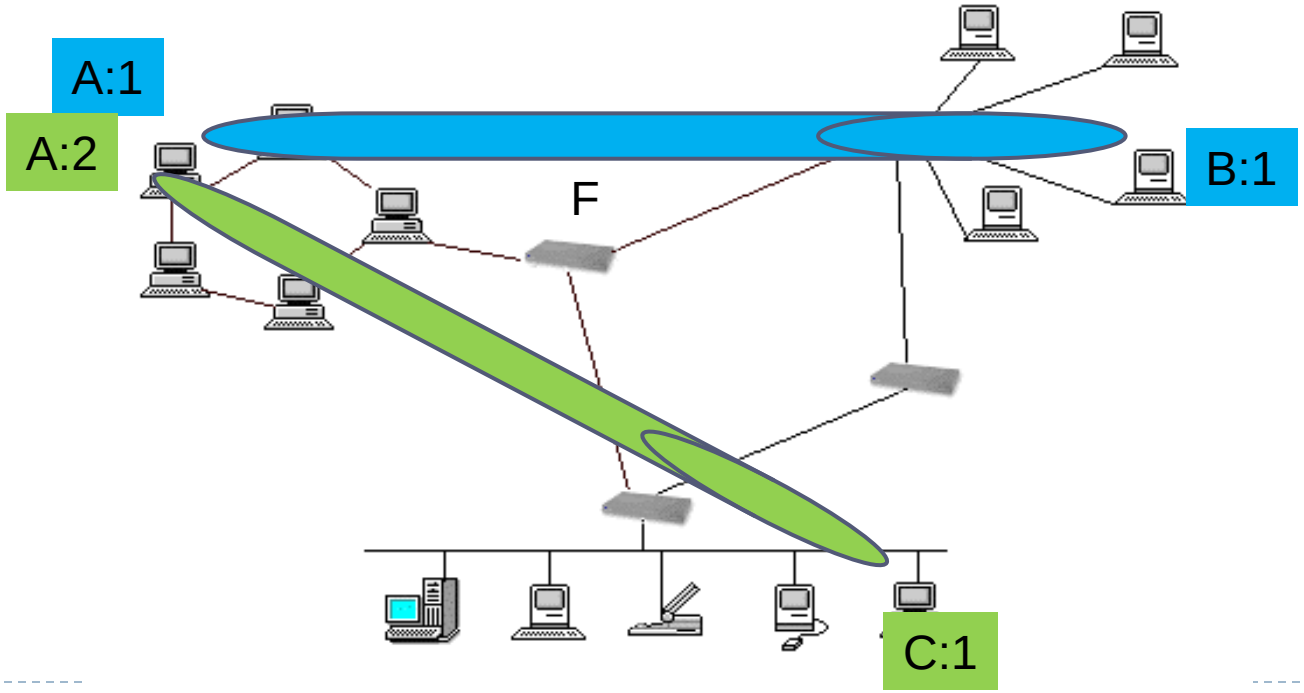
Reconstitue les données applicatives

Reconstitue la séquence des paquets dans l'ordre

Reçoit un paquet à travers le réseau global

La couche transport

- Chaque application est identifiée de façon unique A:i (A machine, i numéro).
- Le transport consiste à acheminer tous les paquets d'une application émettrice A:i vers une application destinataire B:j
- De plus, TCP corrige les erreurs dues au routage non fiable.



UDP

- Au niveau de la couche *Internet* les datagrammes sont routés d'une machine à une autre en fonction des bits de l'adresse IP qui identifient le numéro de réseau. Lors de cette opération aucune distinction n'est faite entre les services ou les utilisateurs qui émettent ou recoivent des datagrammes, tous les datagrammes sont mélangés.
- La couche UDP ajoute un mécanisme qui permet l'identification du service (niveau *Application*). En effet, il est indispensable de faire un tri entre les divers applications : plusieurs programmes de plusieurs utilisateurs peuvent utiliser simultanément la même couche de transport et il ne doit pas y avoir de confusion entre eux.

Identifier une application par son port

- Identifiant de l'application = <@IP + port>
  - Le port est simplement un **entier positif** (2 octets).
  - Pour communiquer avec une application sur le réseau, il faut avoir connaissance de son numéro de port, en plus de l'adresse IP de la machine elle-même.
- Le système d'exploitation local a à sa charge de définir le mécanisme qui permet à une application d'accéder à un port.
- Envoi non bloquant
  - Les paquets sont envoyés sur le réseau au destinataire. Le système met les paquets qui arrivent dans une file d'attente jusqu'à ce qu'un processus (*Application*) les lise. L'émetteur n'est pas bloqué.
- Lecture bloquante
  - A l'inverse, le système bloque un processus qui tente de lire une donnée non encore disponible.

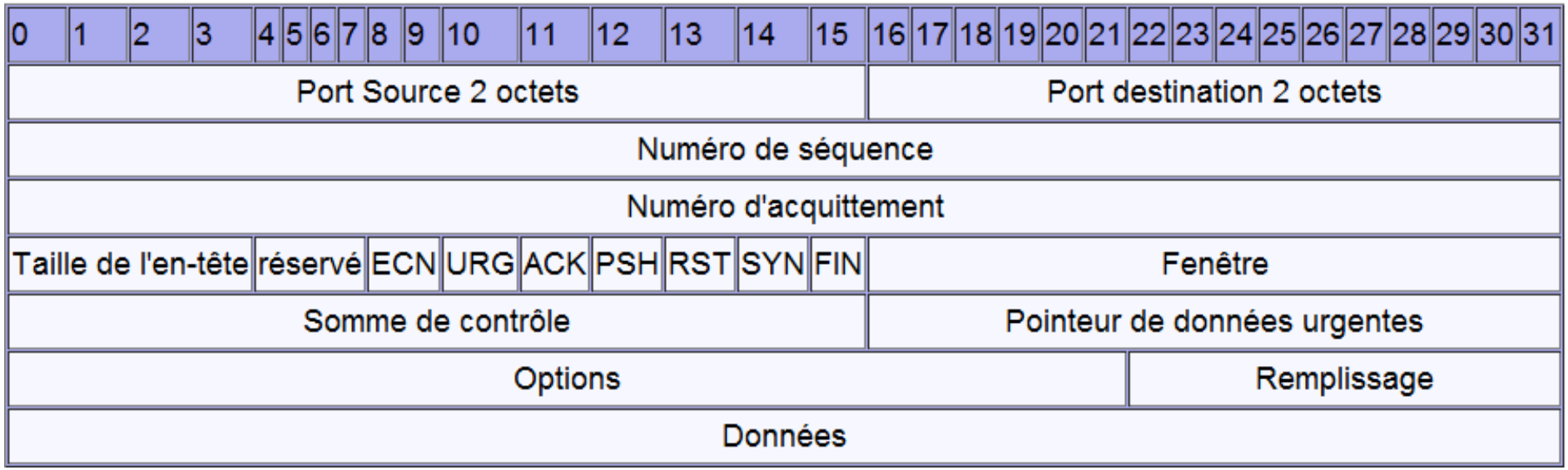
UDP

- UDP apporte un mécanisme de gestion des ports, au dessus de la couche Internet.
- UDP est sans garantie
  - Tous les défauts d'IP sont applicables à UDP.
- C’est à l’application de tenir compte des risques d’acheminement
- UDP est aussi désigné comme un mode de transport « non connecté », ou encore mode datagramme, par opposition à TCP.
- Applications au-dessus d’UDP
  - DNS
  - TFTP ou NFS.

TCP

- Transmission de données :
- Par paquets de tailles variables
- En mode connecté (3 phases) :
  1. Etablissement de la connexion
  2. Transfert de données
  3. Libération de la connexion
- Bidirectionnelle
- Flux non structuré de données
  - ⇒ suite d'octets ("Stream")
- Fiable

Format d'un segment TCP



- En-tête TCP de 20 octets avec options sur 4 octets
- La taille du segment TCP dépend de la MTU de la couche sous-jacente

Identifier une application TCP

- Idem qu'UDP
- La connexion est identifiée par :
  - <@IP source + n° port source,
  - <@IP destination +n° port destination>
- Plusieurs applications peuvent s'exécuter sur la même machine, avec des numéro de port différents
  - Port <= 1024 : numéros réservés
- Multiplexage/Démultiplexage
  - Un seul support physique permet de transmettre des données de plusieurs applications différentes en examinant leur port.

Reséquencement

- Les paquets de la couche transport ont une taille limitée
  - Les données de l'application sont découpées en segment
- Problèmes
  - 2 segments contenant l'information constituant un seul et même fichier à l'origine peuvent arriver dans n'importe quel ordre.
  - Certaines fois, un paquet de données peut également se perdre en chemin et ne pas arriver du tout.
- Services transport
  - service d'accusés de réception des paquets
  - Service de remise en ordre des paquets reçus conformément à l'ordre dans lequel ils ont été émis

Reséquencement

- Emetteur
  - Segmente les données de l'application en plus petits morceaux
  - **Chaque segment est numéroté**. Les numéros de séquence vont croissant d'une unité pour chaque nouveau paquet envoyé vers la destination.
- Destinataire
  - A la réception d'un segment, **envoie un accusé de réception** à l'émetteur mentionnant le numéro de séquence du paquet
  - Le destinataire peut réordonner les paquets qui ne sont pas arrivés dans l'ordre.
- Emetteur
  - L'émetteur **vérifie qu'un acquittement** a bien été reçu pour chaque paquet envoyé, mentionnant le numéro de séquence correspondant à ce paquet.
  - Si un acquittement n'a pas été reçu pour un paquet envoyé, l'émetteur le considère comme perdu et l'envoie de nouveau, en espérant recevoir cette fois un acquittement.

Contrôle de congestion

- Internet peut souffrir de congestion au niveau transport.
  - Trop d'applications envoient trop de paquets à la fois. Certains sont perdus.
- le protocole TCP fournit dans ce but un service supplémentaire : l'ajustement du rythme auquel une application envoie un flux de paquets vers une destination donnée
- Objectif contradictoires:
  - L'application veut envoyer que ses paquets arrivent le plus vite possible
  - Dans l'intérêt général, une seule application ne doit pas monopoliser le réseau

Contrôle de congestion

Les capacités de transmission dépendent de plusieurs paramètres.

- L'état **d'engorgement** du réseau peut varier de manière extrême d'une seconde à l'autre.
- Le **chemin** utilisé entre l'émetteur et la destination a un débit bridé par le lien de plus faible débit
- La **mémoire tampon**
  - Elle stocke les messages en attente dans les routeurs
  - Elle peut être saturée par trop de trafic sur l'un des ordinateurs intermédiaire ou sur l'ordinateur destinataire. Les paquets suivants sont perdus.

Contrôle de congestion

- Principe
  - le protocole TCP ajuste le rythme auquel une application envoie un flux de paquets vers une destination donnée.
- Paramètres
  - Fenêtre de congestion ***W***
    - le nombre de paquets que l'émetteur tolère avoir envoyé vers la destination sans encore avoir reçu l'acquittement
    - W varie dans le temps. Initialement, W=1.
  - Seuil de tolérance ***S***
    - Le seuil de fluctuation maximal toléré pour W
    - S varie dans le temps. Initialement, S=64.
  - Durée d'acquittement estimée ***Tw***
    - estimation du laps de temps au bout duquel on devrait avoir reçu les acquittements correspondant à W paquets envoyés.
    - Estimée initialement avec un test (ping).

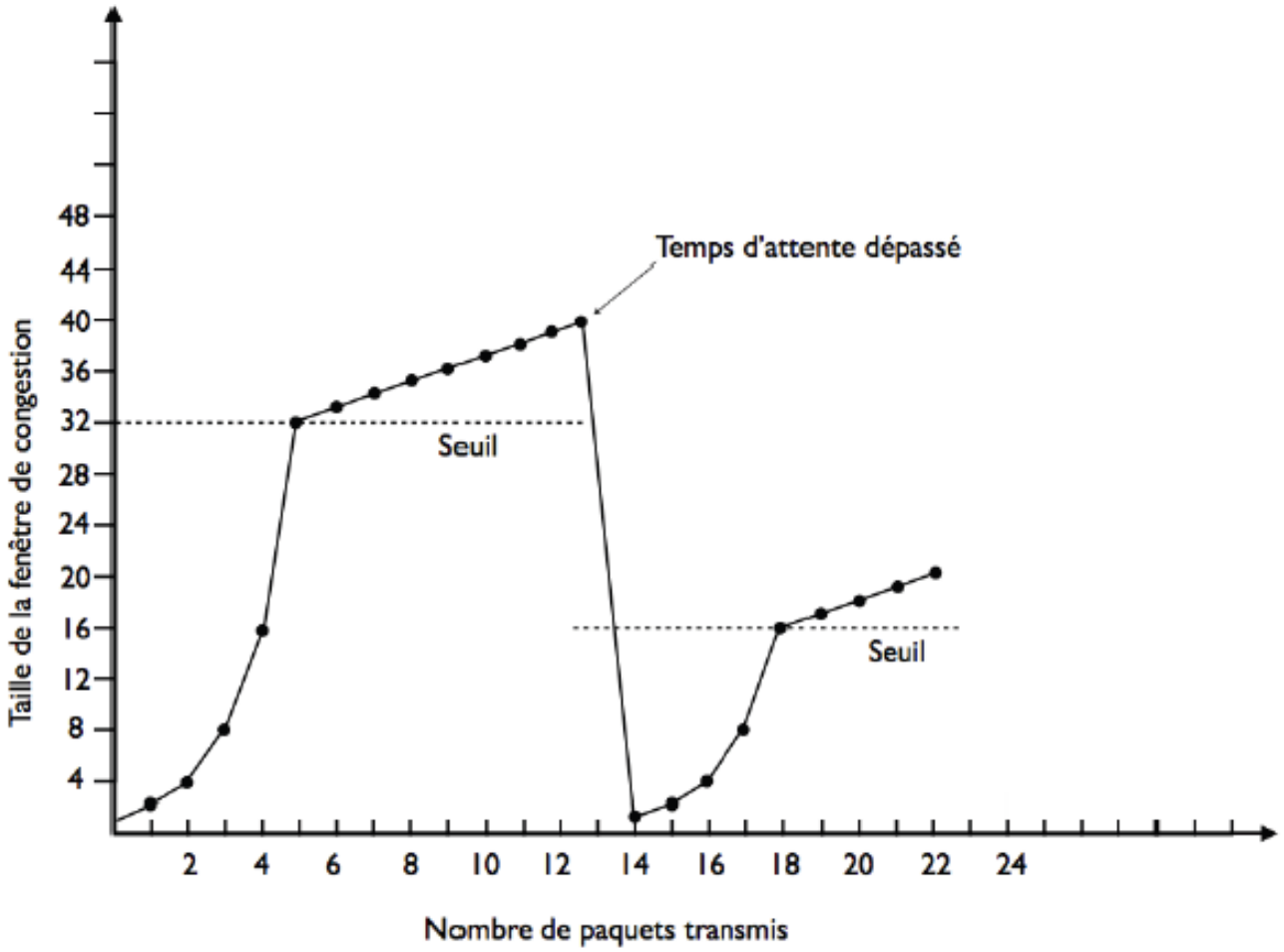
Contrôle de congestion : phase 1

- Protocole de fenêtre glissante
  - **Fenêtre en cours** : il envoie W paquets et attend qu'ils soient tous acquittés.
  - Si tous les paquets sont acquittés avec Tw unités de temps, la valeur de **W est doublée**.
  - L'émetteur envoie alors les W paquets suivants qui deviennent la nouvelle fenêtre en cours et attend leurs acquittements
  - ainsi de suite jusqu'à ce que W = S.

Contrôle de congestion : phase 2

- A partir du seuil W = S, une deuxième phase s'enclenche
  - A chaque itération, **W augmente d'une unité** — au lieu de doubler sa valeur.
- Détection de congestion
  - Si à tout moment au cours la procédure, l'émetteur doit attendre plus de Tw unités de temps avant d'avoir reçu tous les acquittements de la fenêtre en cours, il considère que des paquets ont été perdu dû à une congestion du réseau
  - il ajuste S en divisant sa valeur par deux, réinitialise W = 1 et recommence depuis le début

Contrôle de congestion



TCP. Ajustements progressifs de la cadence d'envois des paquets via les variations de la fenêtre de congestion.

A retenir

- La couche transport a pour fonctions:
  - L'adressage des applications via un identifiant de la forme <@Ip + port>, port entier.
    - UDP, TCP
  - La fiabilité de la communication en cas de perte de paquet ou de déséquencement via :
    - La numérotation des segments
    - L'acquittement des segments. Les segments non acquittés sont réémis et peuvent être à nouveau perdus ou bien redondants.
  - Pour limiter les erreurs, TCP tente de limiter la congestion du réseau en ajustant la cadence d'envoi des paquets aux capacités du réseau.