Module R102: Principes et architecture des réseaux

Laure Petrucci

Département R&T, IUT de Villetaneuse
Université Paris 13
laure.petrucci@univ-paris13.fr

8 septembre 2023

Objectifs du cours

Page web :

http://lipn.univ-paris13.fr/~petrucci/R102

Objectifs :

- Connaître l'architecture des réseaux
- Comprendre une architecture de réseau et les protocoles associés,
- Analyser le fonctionnement des protocoles principaux,
- Codes pour la détection et correction d'erreurs

R102 vs R101

R101

but : vous donner une idée de comment fonctionnent les réseaux en pratique.

- Codage de l'information,
- Notions de base sur la couche IP,
- Configuration des machines,
- Utilisation de certains protocoles et services : NIS, NFS, DNS, NAT.

R102 vs R101

R101

but : vous donner une idée de comment fonctionnent les réseaux en pratique.

- Codage de l'information,
- Notions de base sur la couche IP,
- Configuration des machines,
- Utilisation de certains protocoles et services : NIS, NFS, DNS, NAT.

R102

but : expliquer les principes de structuration du logiciel qui gère les réseaux,

- architecture des réseaux,
- structuration en couches,
- encapsulation,
- modèle TCP/IP,
- techniques de traitement des erreurs.

R102 vs R101

R101

but : vous donner une idée de comment fonctionnent les réseaux en pratique.

- Codage de l'information,
- Notions de base sur la couche IP,
- Configuration des machines,
- Utilisation de certains protocoles et services : NIS, NFS, DNS, NAT.

R102

but : expliquer les principes de structuration du logiciel qui gère les réseaux,

- architecture des réseaux,
- structuration en couches,
- encapsulation,
- modèle TCP/IP,
- techniques de traitement des erreurs.

but secondaire : vérifier que vous savez faire des calculs.

Organisation du cours

Organisation :

- 4 CM, 2 TD, 3 TP.
- Intervenants :
  L. Petrucci, M. Graba, M.-A. Ouamri, M. Rogers.

Évaluation :

- TDs notés
- un contrôle (2h)
  - documents autorisés : le polycopié du cours,
  - **La calculette est interdite.**

Organisation du cours

Organisation :

- 4 CM, 2 TD, 3 TP.
- Intervenants :
  L. Petrucci, M. Graba, M.-A. Ouamri, M. Rogers.

Évaluation :

- TDs notés
- un contrôle (2h)
  - documents autorisés : le polycopié du cours,
  - **La calculette est interdite.**

Structure du module

1. Généralités
2. Modèle de référence ISO/OSI
3. Modèle TCP/IP
4. Détection et correction d'erreurs

Qu'est-ce qu'un réseau ?

Réseau informatique

Un réseau informatique est un ensemble de dispositifs matériels et logiciels qui permettent à des équipements d'échanger des données numériques.

Données numériques : Une suite d'éléments binaires (bits).

Exemple

1110001110010111

Unités utilisées

Unités multiples des bits :

Unité	Symbole	Valeur (bits)
kilo-bit	Kb	$10^3 = 1\,000$
méga-bit	Mb	$10^6 = 1\,000\,000$
giga-bit	Gb	$10^9 = 1\,000\,000\,000$
téra-bit	Tb	$10^{12} = 1\,000\,000\,000\,000$
péta-bit	Pb	$10^{15} = 1\,000\,000\,000\,000\,000$
exa-bit	Eb	$10^{18} = 1\,000\,000\,000\,000\,000\,000$

- unités utilisées pour mesurer la vitesse de transmission des données

Unités utilisées

Rappel

1 octet $= 8$ bits $= 2^3$ bits.

Unités multiples des octets :

Unité	Symbole	Valeur (octets)
kibi-octet	Kio	$2^{10} = 1\,024$
mébi-octet	Mio	2^{20}
gibi-octet	Gio	2^{30}
tébi-octet	Tio	2^{40}
pébi-octet	Pio	2^{50}
exbi-octet	Eio	2^{60}

Les unités multiples des octets sont utilisées pour mesurer la capacité de stockage.

Attention !

Entre kilo et kibi la difference est petite :

$$\begin{aligned} 1Ko &= 1\,000\,o \\ 1Kio &= 1\,024\,o \end{aligned}$$

mais ensuite elle devient de plus en plus grande :

$$\begin{aligned} 1To &= 1\,000\,000\,000\,000\,o \\ 1Tio &= 1\,099\,511\,627\,776\,o \end{aligned}$$

Presque 100 giga-octets de difference !

Quelles sont les entités communiquant ?

Les ressources matérielles

- composants de traitement de l'information :
  - Ordinateur, serveur, smartphone, tablette, ...
  - Imprimante, scanner, disque réseau, ...
  - Automates industriels, robots, ...
  - Objets connectés : montres, voitures, frigo, ...
- composants de transmission :
  - Câbles, cartes réseau, modems,...
  - Borne Wifi, point d'accès, ...
  - Concentrateur (hub), commutateur, routeur

Les ressources logicielles

- Protocoles réseaux : Ethernet, ARP, IP, TCP, UDP, HTTPS, DNS ...
- Système d'exploitation : IOS des switch/routeur,
- Applications : services web, plateformes des réseaux sociaux, bases de données, jeux

Quelles sont les entités communiquant ?

Les ressources matérielles

- composants de traitement de l'information :
  - Ordinateur, serveur, smartphone, tablette, ...
  - Imprimante, scanner, disque réseau, ...
  - Automates industriels, robots, ...
  - Objets connectés : montres, voitures, frigo, ...
- composants de transmission :
  - Câbles, cartes réseau, modems,...
  - Borne Wifi, point d'accès, ...
  - Concentrateur (hub), commutateur, routeur

Les ressources logicielles

- Protocoles réseaux : Ethernet, ARP, IP, TCP, UDP, HTTPS, DNS ...
- Système d'exploitation : IOS des switch/routeur,
- Applications : services web, plateformes des réseaux sociaux, bases de données, jeux

Mesures de performance - Débit

Débit

quantité d'information que le réseau peut transmettre par unité de temps :

$$débit = \frac{quantité\ d'information}{temps}$$

L'unité est par conséquent le bit par seconde, noté b/s ou $b.s^{-1}$.

Pour les réseaux à haut débit on utilise :

- des méga-bits par secondes (notés Mb/s ou $Mb.s^{-1}$)
- ou giga-bits par secondes (notés Gb/s ou $Gb.s^{-1}$).

On parle de

- Bas débit : entre 0 et 512 Kb/s,
- Haut débit : entre 512 Kb/s et 30 Mb/s,
- Très haut débit : entre 30 Mb/s et 1Gb/s.

Mesures de performance - Débit

Débit

quantité d'information que le réseau peut transmettre par unité de temps :

$$débit = \frac{quantité\ d'information}{temps}$$

L'unité est par conséquent le bit par seconde, noté b/s ou $b.s^{-1}$.

Pour les réseaux à haut débit on utilise :

- des méga-bits par secondes (notés Mb/s ou $Mb.s^{-1}$)
- ou giga-bits par secondes (notés Gb/s ou $Gb.s^{-1}$).

On parle de

- Bas débit : entre 0 et 512 Kb/s,
- Haut débit : entre 512 Kb/s et 30 Mb/s,
- Très haut débit : entre 30 Mb/s et 1Gb/s.

Mesures de performance - Débit

Débit

quantité d'information que le réseau peut transmettre par unité de temps :

$$débit = \frac{quantité\ d'information}{temps}$$

L'unité est par conséquent le bit par seconde, noté b/s ou $b.s^{-1}$.

Pour les réseaux à haut débit on utilise :

- des méga-bits par secondes (notés Mb/s ou $Mb.s^{-1}$)
- ou giga-bits par secondes (notés Gb/s ou $Gb.s^{-1}$).

On parle de

- Bas débit : entre 0 et 512 Kb/s,
- Haut débit : entre 512 Kb/s et 30 Mb/s,
- Très haut débit : entre 30 Mb/s et 1Gb/s.

Débits nominal et utile

Débit nominal/théorique

quantité théorique maximale d'information pouvant être transmise par unité de temps

Débit utile/effectif

quantité d'information effectivement transmise par unité de temps

Remarque

Le débit théorique est toujours plus élevé que le débit effectif :

$$\text{débit effectif} < \text{débit théorique}$$

Débits nominal et utile

Débit nominal/théorique

quantité théorique maximale d'information pouvant être transmise par unité de temps

Débit utile/effectif

quantité d'information effectivement transmise par unité de temps

Remarque

Le débit théorique est toujours plus élevé que le débit effectif :

$$\text{débit effectif} < \text{débit théorique}$$

Causes de la différence entre sébits nominal et utile

- Les protocoles utilisés ajoutent de l'information de contrôle :
  - *Analogie : emballage des colis*
- Les protocoles utilisés imposent des règles de traitement qui augment le temps de traitement.
  - *Exemple : Traitement de pertes de message*
- Il faut ajouter :
  - le temps de propagation de signaux,
  - les temps de traitement par les équipements intermédiaires
  - les retards en raison de la charge du réseau

Taux d'utilisation

Taux d'utilisation du réseau

rapport du débit utile au débit nominal :

$$taux\ d'utilisation = \frac{débit\ utile}{débit\ nominal}$$

Le taux d'utilisation est toujours inférieur à 100%

Temps d'acheminement des messages

Le temps total d'acheminement d'un message se compose de deux parties :

- le temps de transmission est le temps mis pour transmettre la quantité d'information du message :

$$\text{temps}_{\text{transmission}} = \frac{\text{quantité information}}{\text{débit}}$$

- le temps de propagation (latence) est le temps mis pour que le signal se propage sur le matériel

$$\text{temps}_{\text{propagation}} = \frac{\text{distance parcourue}}{\text{vitesse}}$$

Temps d'acheminement des messages

Le temps total d'acheminement d'un message se compose de deux parties :

- le temps de transmission est le temps mis pour transmettre la quantité d'information du message :

$$\text{temps}_{\text{transmission}} = \frac{\text{quantité information}}{\text{débit}}$$

- le temps de propagation (latence) est le temps mis pour que le signal se propage sur le matériel

$$\text{temps}_{\text{propagation}} = \frac{\text{distance parcourue}}{\text{vitesse}}$$

Temps d'acheminement des messages

Le temps total d'acheminement d'un message se compose de deux parties :

- le temps de transmission est le temps mis pour transmettre la quantité d'information du message :

$$\text{temps}_{\text{transmission}} = \frac{\text{quantité information}}{\text{débit}}$$

- le temps de propagation (latence) est le temps mis pour que le signal se propage sur le matériel

$$\text{temps}_{\text{propagation}} = \frac{\text{distance parcourue}}{\text{vitesse}}$$

Temps d'acheminement des messages

Le temps total d'acheminement d'un message se compose de deux parties :

- le temps de transmission est le temps mis pour transmettre la quantité d'information du message :

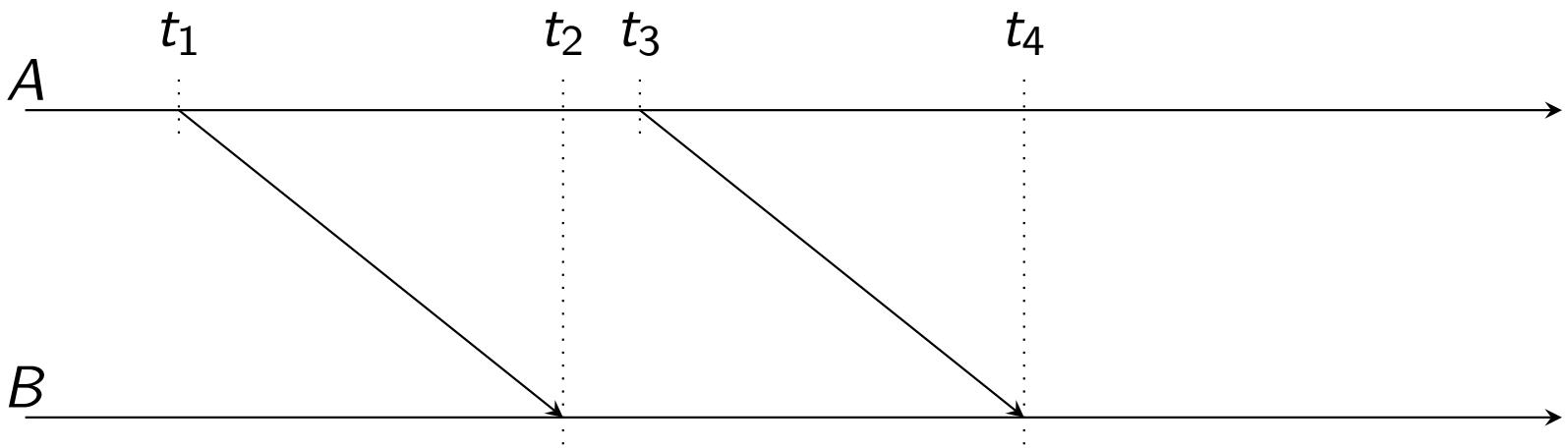
$$\text{temps}_{\text{transmission}} = \frac{\text{quantité information}}{\text{débit}}$$

- le temps de propagation (latence) est le temps mis pour que le signal se propage sur le matériel ; les équipements traversés peuvent introduire des *retards*.

$$\text{temps}_{\text{propagation}} = \frac{\text{distance parcourue}}{\text{vitesse}} + \text{retards}$$

Temps d'acheminement des messages

La machine A envoie un message à la machine B :



- $t_3 - t_1 = t_4 - t_2 =$ temps de transmission,
- $t_2 - t_1 = t_4 - t_3 =$ temps de propagation,
- $t_4 - t_1 =$ temps total d'acheminement.

$$\text{temps}_{\text{total}} = \text{temps}_{\text{transmission}} + \text{temps}_{\text{propagation}}$$

La Gigue (jitter)

Gigue : variation de la latence avec le temps

- Causes :
  - Congestion du réseau
  - Changement des routes
  - ...
- La gigue peut être faible avec une latence forte ou inversement
- Une gigue forte affecte la qualité des applications temps réel : jeux interactifs, visioconférence, ...

Classification des réseaux

Plusieurs critères :

1. la performance du réseau (débit)
2. la technologie de transmission utilisée
3. le rayon de couverture géographique
4. le mode de communication

Sigle	Nom	Distance	Débit
PAN	Personal Area Network	quelques mètres	$1 Mb/s$
LAN	Local Area Network	jusqu'à $2 km$	de $10 Mb/s$ à $1 Gb/s$
MAN	Metropolitan Area Network	jusqu'à $100 km$	environ $100 Mb/s$
WAN	Wide Area Network	milliers de km	quelques Mb/s

Classification des réseaux

Plusieurs critères :

1. la performance du réseau (débit)
2. la technologie de transmission utilisée
3. le rayon de couverture géographique
4. le mode de communication

Sigle	Nom	Distance	Débit
PAN	Personal Area Network	quelques mètres	$1 Mb/s$
LAN	Local Area Network	jusqu'à $2 km$	de $10 Mb/s$ à $1 Gb/s$
MAN	Metropolitan Area Network	jusqu'à $100 km$	environ $100 Mb/s$
WAN	Wide Area Network	milliers de km	quelques Mb/s

Technologies de transmission

Deux types de technologies de transmission :

1. Diffusion : un seul canal de transmission partagé par tous les équipements connectés
2. Point-à-point : plusieurs connexions, chacune entre deux machines

Réseau à diffusion

Réseau à diffusion

- un seul canal de transmission partagé par tous les équipements
- chaque message envoyé est reçu par toutes les machines du réseau
- dans le message un champ d'adresse permet d'identifier le destinataire réel,
- à la réception d'un message :
  - la machine lit ce champ et traite le message si elle reconnait son adresse
  - sinon la machine ignore le message
- utilisé sur des petits réseaux géographiquement limités

Valeur spéciale dans le champ d'adresse

- **diffusion générale** (broadcast) : message reçu et traité par toutes les machines
- **diffusion restreinte** (multicast) : message traité par un sous-ensemble des machines

Réseau à diffusion

Réseau à diffusion

- un seul canal de transmission partagé par tous les équipements
- chaque message envoyé est reçu par toutes les machines du réseau
- dans le message un champ d'adresse permet d'identifier le destinataire réel,
- à la réception d'un message :
  - la machine lit ce champ et traite le message si elle reconnait son adresse
  - sinon la machine ignore le message
- utilisé sur des petits réseaux géographiquement limités

Valeur spéciale dans le champ d'adresse

- **diffusion générale** (broadcast) : message reçu et traité par toutes les machines
- **diffusion restreinte** (multicast) : message traité par un sous-ensemble des machines

Réseau point-à-point

- plusieurs connexions, chacune entre deux machines
- pour aller de la source à la destination un paquet peut transiter par plusieurs équipements intermédiaires.
- ce type de réseau offre plusieurs routes possibles, de longueurs différentes, pour rejoindre la même destination
- il est important de pouvoir sélectionner les meilleures routes
- utilisé sur de grands réseaux

Architectures de réseaux

- Architecture physique : topologie d'interconnexion physique des composants du réseau
- Architecture logique : topologie de circulation de l'information du réseau
- Architectures classiques : bus, étoile, anneau, arbre, graphe et topologie complète

Analogie :

- Architecture physique : cartographie des routes
- Architecture logique : sens de circulation

Architectures de réseaux

- Architecture physique : topologie d'interconnexion physique des composants du réseau
- Architecture logique : topologie de circulation de l'information du réseau
- Architectures classiques : bus, étoile, anneau, arbre, graphe et topologie complète

Analogie :

- Architecture physique : cartographie des routes
- Architecture logique : sens de circulation

Architectures de réseaux

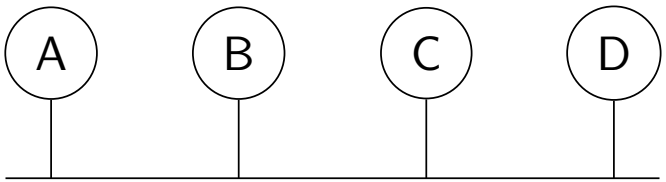
- Architecture physique : topologie d'interconnexion physique des composants du réseau
- Architecture logique : topologie de circulation de l'information du réseau
- Architectures classiques : bus, étoile, anneau, arbre, graphe et topologie complète

Analogie :

- Architecture physique : cartographie des routes
- Architecture logique : sens de circulation

La topologie bus

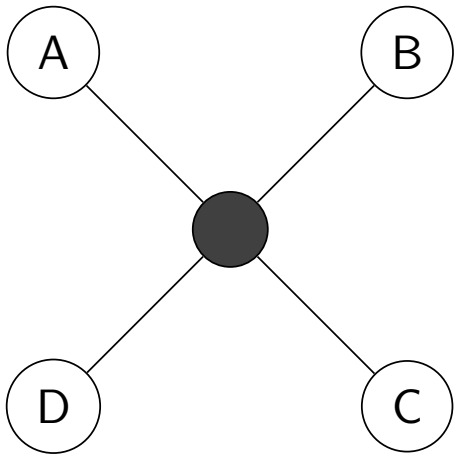
Câblage unique auquel les différents nœuds sont connectés



- Le câble est l'unique élément matériel constituant le réseau et seuls les nœuds génèrent des signaux
- Lorsqu'une station est en panne et ne transmet plus sur le réseau, elle ne perturbe pas le réseau.
- Une seule coupure du câble empêche toute station d'échanger des messages sur le réseau.

La topologie en étoile

Tout nœud est connecté à un point central.

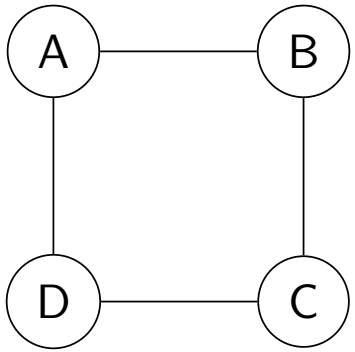


Le point central doit être un matériel actif, comme un concentrateur ou un commutateur, c'est-à-dire un matériel qui remet en forme les signaux et les régénère avant de les retransmettre.

- La panne d'un nœud ne perturbe pas le fonctionnement global du réseau.
- Une panne de l'équipement central qui relie tous les stations rend le réseau totalement inutilisable.

La topologie en anneau

La topologie en anneau repose sur une boucle fermée.

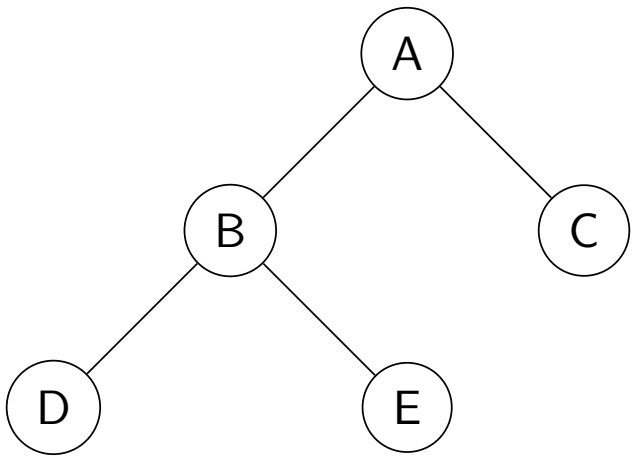


Toutes les stations sont connectées en chaîne les unes aux autres par une liaison point-à-point.

- Chaque station joue le rôle de nœud intermédiaire.
- Les informations transitent par chaque nœud, qui se comporte comme un répéteur et retransmet les informations au nœud suivant.
- La défaillance d'une station rend le réseau inutilisable.

La topologie en arbre

Dans la topologie en arbre, les nœuds sont reliés entre eux de manière hiérarchique.

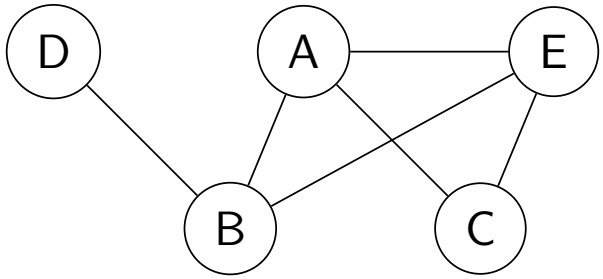


Le sommet de haut niveau est appelé *racine*.
Chaque nœud peut être connecté à plusieurs nœuds de niveau inférieur (ses *fils*).

Si un nœud tombe en panne, il paralyse tout le sous-arbre dont il est racine.

Le graphe

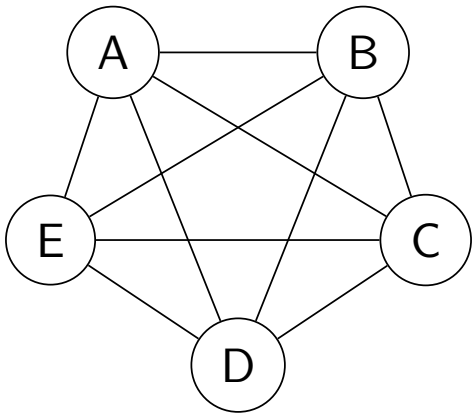
Dans la topologie en graphe les nœuds sont connectés entre eux par des liaisons point-à-point, de manière à constituer un graphe connexe.



L'information peut circuler sur le réseau en suivant des itinéraires différents.

La topologie complète

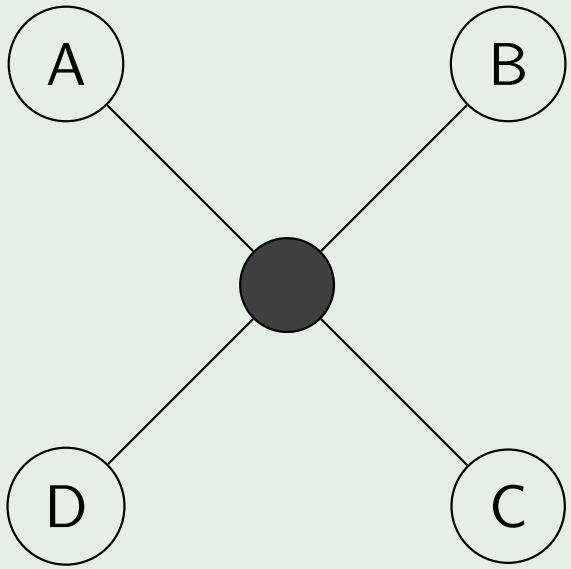
Dans la topologie complète il y a une liaison point-à-point entre chaque paire de nœuds, donc toute station est reliée à toutes les autres.



L'inconvénient de cette topologie est le nombre de liaisons nécessaires qui devient très élevé lorsque le nombre de stations augmente.

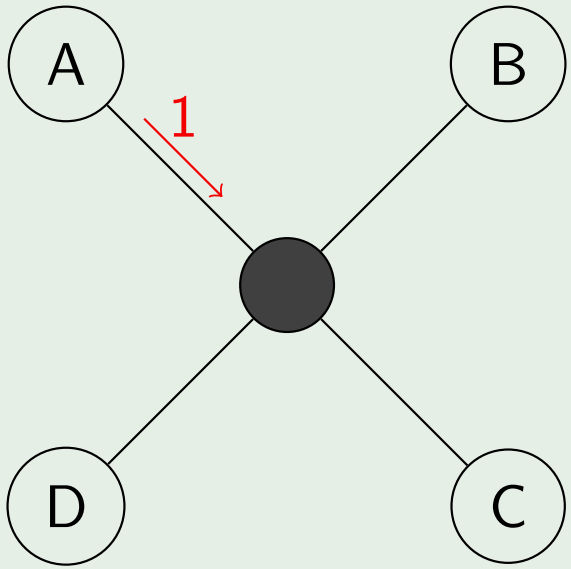
Architecture logique / physique

Bus sur une étoile



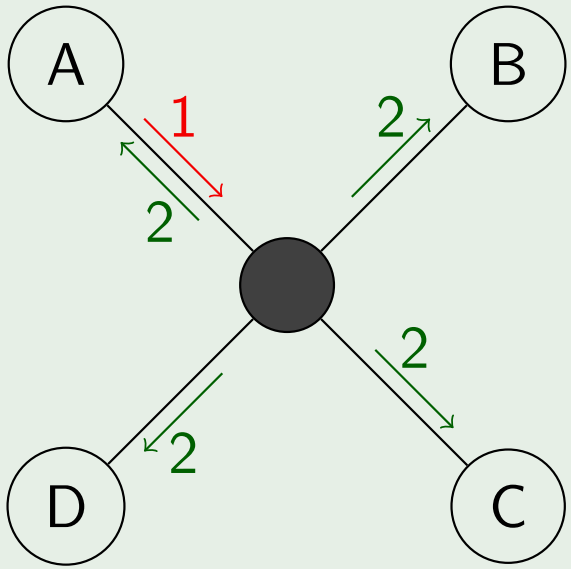
Architecture logique / physique

Bus sur une étoile



Architecture logique / physique

Bus sur une étoile



Le modèle OSI

Logiciels de réseau

But d'un réseau : Interconnecter des ordinateurs différents.

Problème

Les ordinateurs à connecter peuvent être vraiment différents : architectures hétérogènes, différents systèmes d'exploitations, etc.

Logiciels de réseau

But d'un réseau : Interconnecter des ordinateurs différents.

Problème

Les ordinateurs à connecter peuvent être vraiment différents : architectures hétérogènes, différents systèmes d'exploitations, etc.

Normalisation

- Une norme offre un cadre qui permet à des entités hétérogène de communiquer.
- Elle permet également d'assurer un niveau minimum de qualité (QoS).

- Acteurs de la normalisation en télécoms :

Union Internationale des Télécommunications
https://www.itu.int/fr/

Commission Internationale de l'Electronique
https://www.iec.ch/

Organisation Internationale de Standardisation
https://www.iso.org/

Modèle de référence OSI

Le modèle de référence défini par l'ISO est

OSI : *Open Systems Interconnection*

- Il permet à des systèmes hétérogènes de s'interconnecter et d'échanger des informations.
- Il est indépendant de la structure et de la technologie des matériels employés.
- Cadre permettant d'assurer le maximum de compatibilité entre les entités communicantes, tout en minimisant les contraintes permettant de les réaliser.

Problème : complexité des logiciels de réseaux

Les logiciels de réseaux sont très complexes puisqu'ils doivent gérer beaucoup de problèmes :

- transmission des données entre les machines,
- vérifier que les données sont transmises sans erreur,
- trouver le bon chemin pour acheminer les messages d'un réseau à un autre (routage),
- offrir des services divers (streaming $\neq$ transfert de fichier),
- respecter des paramètres de qualité de service,
- etc.

Comment réduire la complexité de conception ?

Problème : complexité des logiciels de réseaux

Maintenance du logiciel : Après la mise en œuvre des logiciels des réseaux, il faut pouvoir apporter des modifications pour

- en corriger les fautes,
- en améliorer l'efficacité ou autres caractéristiques,
- ajouter des nouvelles fonctionnalités.

Cela demande un gros effort si le logiciel est très complexe et s'il n'est pas bien structuré.

Comment permettre la maintenance logicielle ?

Principes de la structuration en couches

La complexité de conception, de réalisation et de maintenance des logiciels et de l'architecture des réseaux, est maîtrisée grâce à une organisation en couches ou niveaux :

Niveau	Couche
$N+1$	Couche $N+1$
N	Couche N
$N-1$	Couche $N-1$
$\vdots$	$\vdots$
3	Couche 3
2	Couche 2
1	Couche 1

Chaque couche est bâtie au-dessus de la précédente.

Idées de la structuration en couches

Chaque couche s'occupe de gérer seulement certains problèmes :

- fournit des services à la couche immédiatement supérieure, tout en lui dissimulant les détails d'implémentation
- est implémentée de façon indépendante des autres,

Concept répandu en informatique : masquage d'informations, types abstraits de données, encapsulation de données, programmation par objets.

Idées de la structuration en couches

Chaque couche s'occupe de gérer seulement certains problèmes :

- fournit des services à la couche immédiatement supérieure, tout en lui dissimulant les détails d'implémentation

  $\Rightarrow$ Cela permet de réduire la complexité de l'implémentation.
- est implémentée de façon indépendante des autres,

Concept répandu en informatique : masquage d'informations, types abstraits de données, encapsulation de données, programmation par objets.

Idées de la structuration en couches

Chaque couche s'occupe de gérer seulement certains problèmes :

- fournit des services à la couche immédiatement supérieure, tout en lui dissimulant les détails d'implémentation
  $\Rightarrow$ Cela permet de réduire la complexité de l'implémentation.
- est implémentée de façon indépendante des autres,
  $\Rightarrow$ Cela rend possible la maintenance logicielle.

Concept répandu en informatique : masquage d'informations, types abstraits de données, encapsulation de données, programmation par objets.

Idées de la structuration en couches

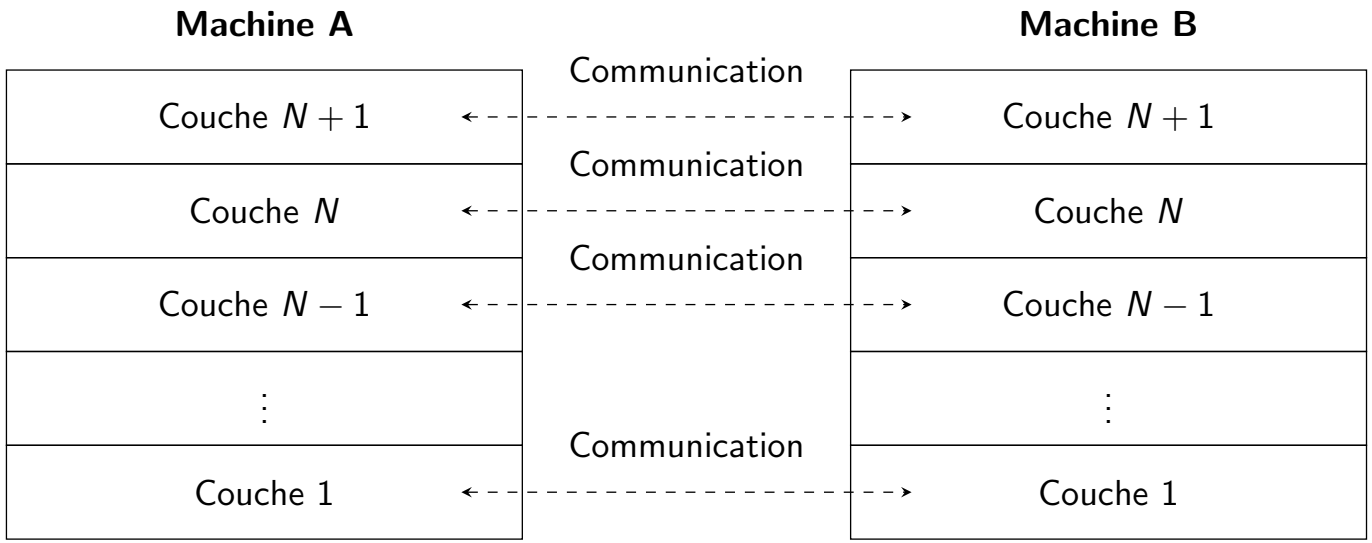
Chaque couche s'occupe de gérer seulement certains problèmes :

- fournit des services à la couche immédiatement supérieure, tout en lui dissimulant les détails d'implémentation
  $\Rightarrow$ Cela permet de réduire la complexité de l'implémentation.
- est implémentée de façon indépendante des autres,
  $\Rightarrow$ Cela rend possible la maintenance logicielle.

Concept répandu en informatique : masquage d'informations, types abstraits de données, encapsulation de données, programmation par objets.

Idées de la structuration en couches

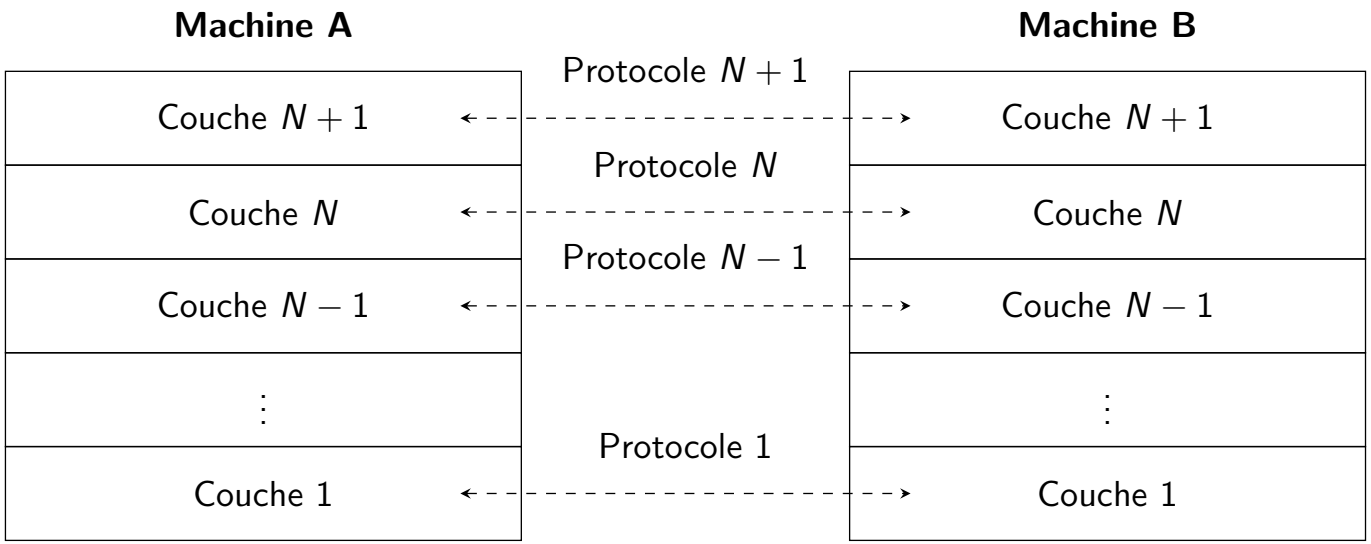
Chaque couche ne peut dialoguer qu'avec une couche de même niveau sur une autre machine.



Les communications se font donc entre entités homologues.

Protocole

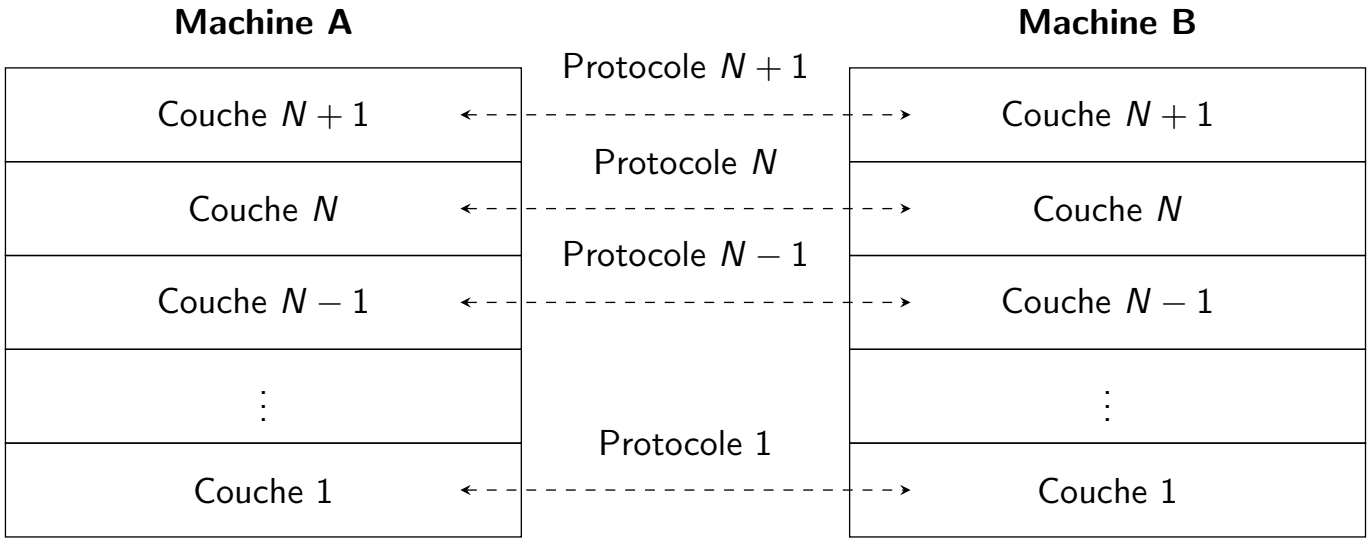
La communication entre deux entités homologues de niveau N obéit à un ensemble de règles.



Un protocole est une convention acceptée par les parties communicantes sur la façon dont leur dialogue doit prendre place. La violation du protocole rendrait la communication difficile, voire impossible.

Protocole

La communication entre deux entités homologues de niveau N obéit à un ensemble de règles.

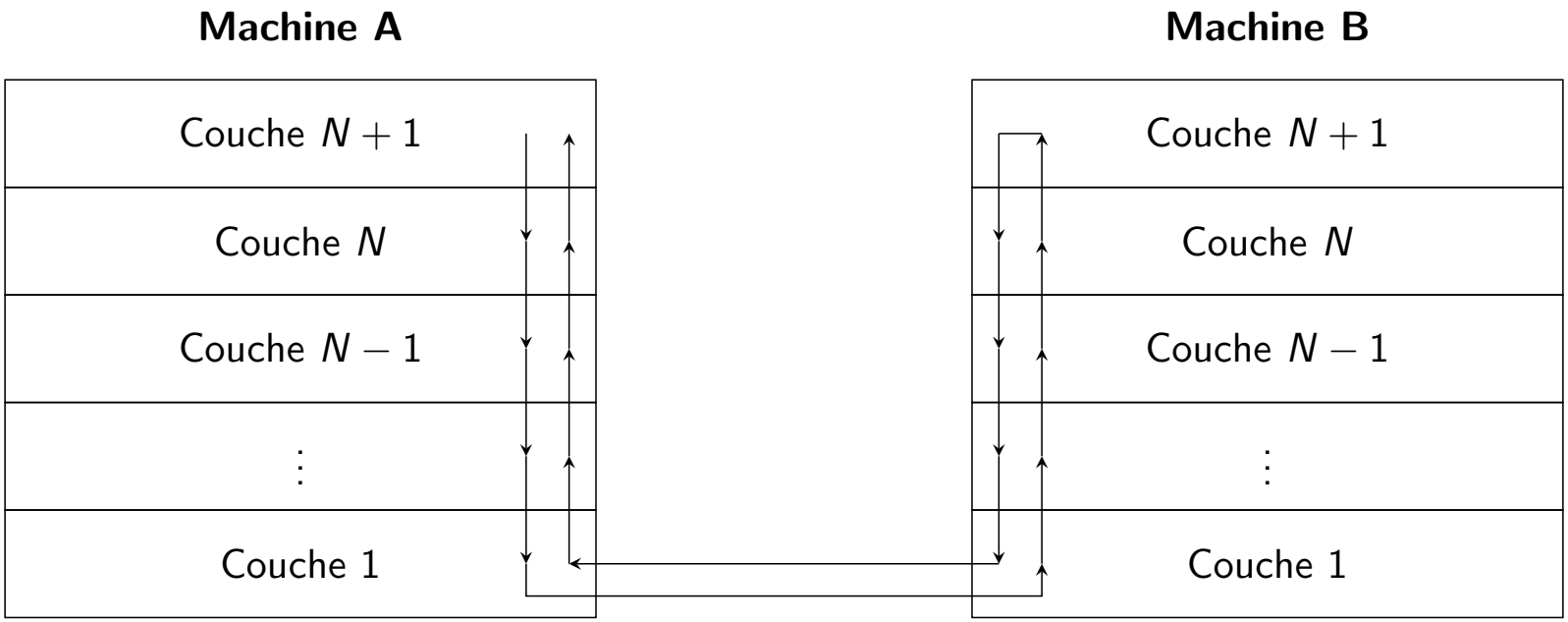


Protocole

Un protocole de niveau N est un ensemble de règles et formats, syntaxiques et sémantiques utilisés pour assurer la communication entre deux entités homologues de niveau N.

Implémentation

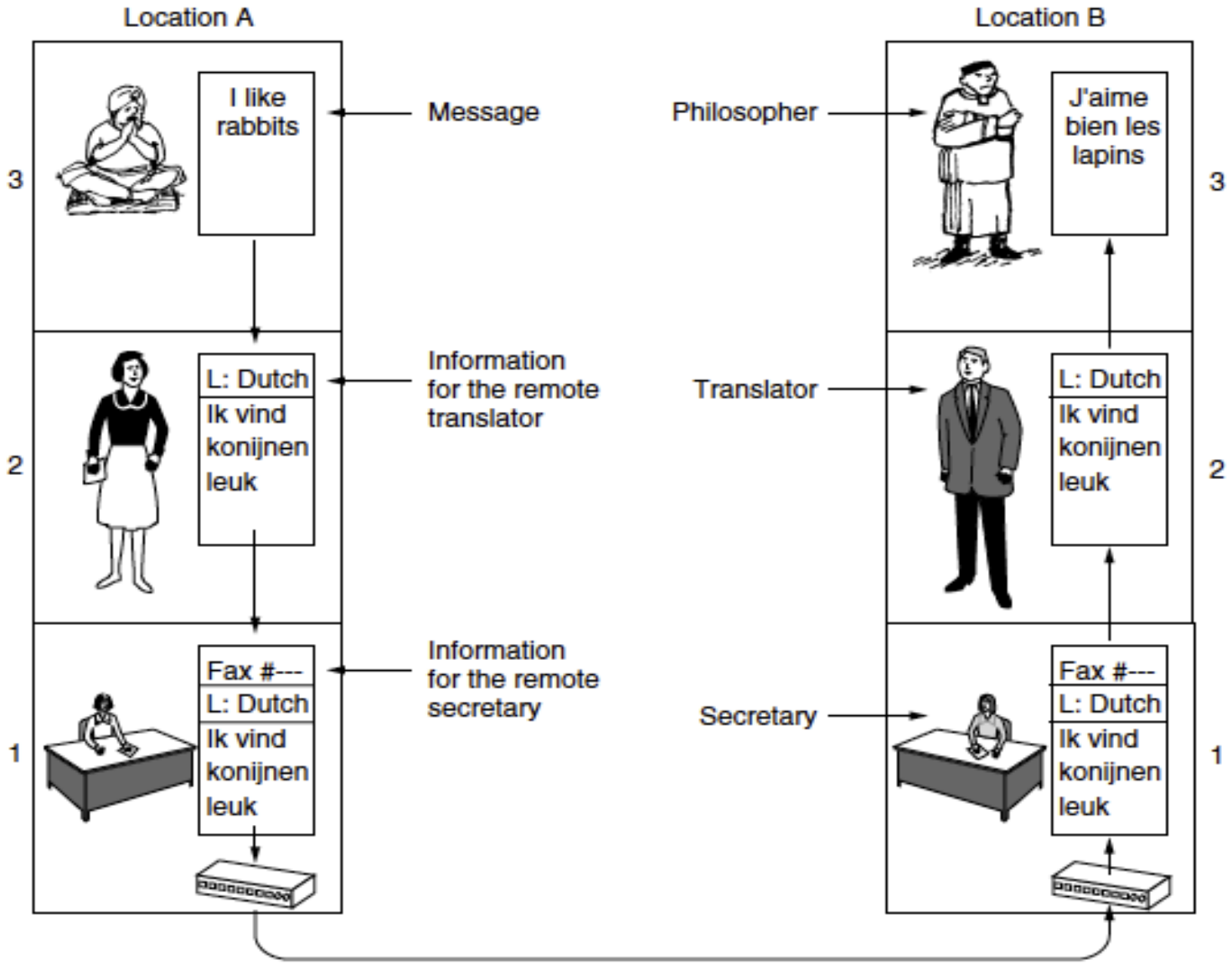
En réalité aucune donnée ne passe directement de la couche N de la machine A à la couche N de la machine B.



Chaque couche :

- utilise les services de la couche immédiatement au-dessous
- offre des services à la couche au-dessus.

Exemple : les philosophes



La couche N

- La couche de niveau N doit fournir des services à la couche de niveau $N+1$ tout en lui dissimulant les détails d'implémentation.
- La couche de niveau $N+1$ communique à la couche N les caractéristiques du service attendu.
- Pour offrir les services à la couche $N+1$, la couche N utilise les services offerts par la couche $N-1$.
- Chaque couche peut interagir uniquement avec les deux couches adjacentes.
- Une couche N est constituée d'un ensemble d'entités formant un sous-système de niveau N.

La couche N

- La couche de niveau N doit fournir des services à la couche de niveau $N+1$ tout en lui dissimulant les détails d'implémentation.
- La couche de niveau $N+1$ communique à la couche N les caractéristiques du service attendu.
- Pour offrir les services à la couche $N+1$, la couche N utilise les services offerts par la couche $N-1$.
- Chaque couche peut interagir uniquement avec les deux couches adjacentes.
- Une couche N est constituée d'un ensemble d'entités formant un sous-système de niveau N.

La couche N

- La couche de niveau N doit fournir des services à la couche de niveau $N+1$ tout en lui dissimulant les détails d'implémentation.
- La couche de niveau $N+1$ communique à la couche N les caractéristiques du service attendu.
- Pour offrir les services à la couche $N+1$, la couche N utilise les services offerts par la couche $N-1$.
- Chaque couche peut interagir uniquement avec les deux couches adjacentes.
- Une couche N est constituée d'un ensemble d'entités formant un sous-système de niveau N.

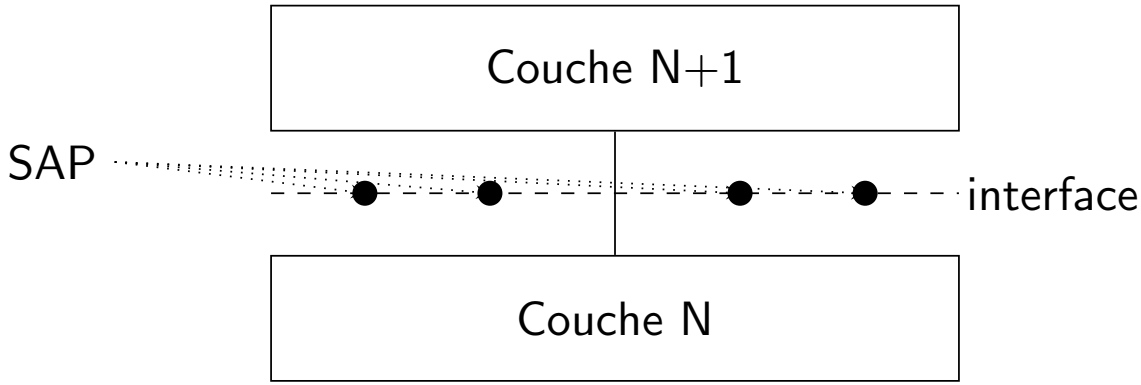
La couche N

- La couche de niveau N doit fournir des services à la couche de niveau $N+1$ tout en lui dissimulant les détails d'implémentation.
- La couche de niveau $N+1$ communique à la couche N les caractéristiques du service attendu.
- Pour offrir les services à la couche $N+1$, la couche N utilise les services offerts par la couche $N-1$.
- Chaque couche peut interagir uniquement avec les deux couches adjacentes.
- Une couche N est constituée d'un ensemble d'entités formant un sous-système de niveau N.

La couche N

- La couche de niveau N doit fournir des services à la couche de niveau $N+1$ tout en lui dissimulant les détails d'implémentation.
- La couche de niveau $N+1$ communique à la couche N les caractéristiques du service attendu.
- Pour offrir les services à la couche $N+1$, la couche N utilise les services offerts par la couche $N-1$.
- Chaque couche peut interagir uniquement avec les deux couches adjacentes.
- Une couche N est constituée d'un ensemble d'entités formant un sous-système de niveau N.

SAP et interfaces

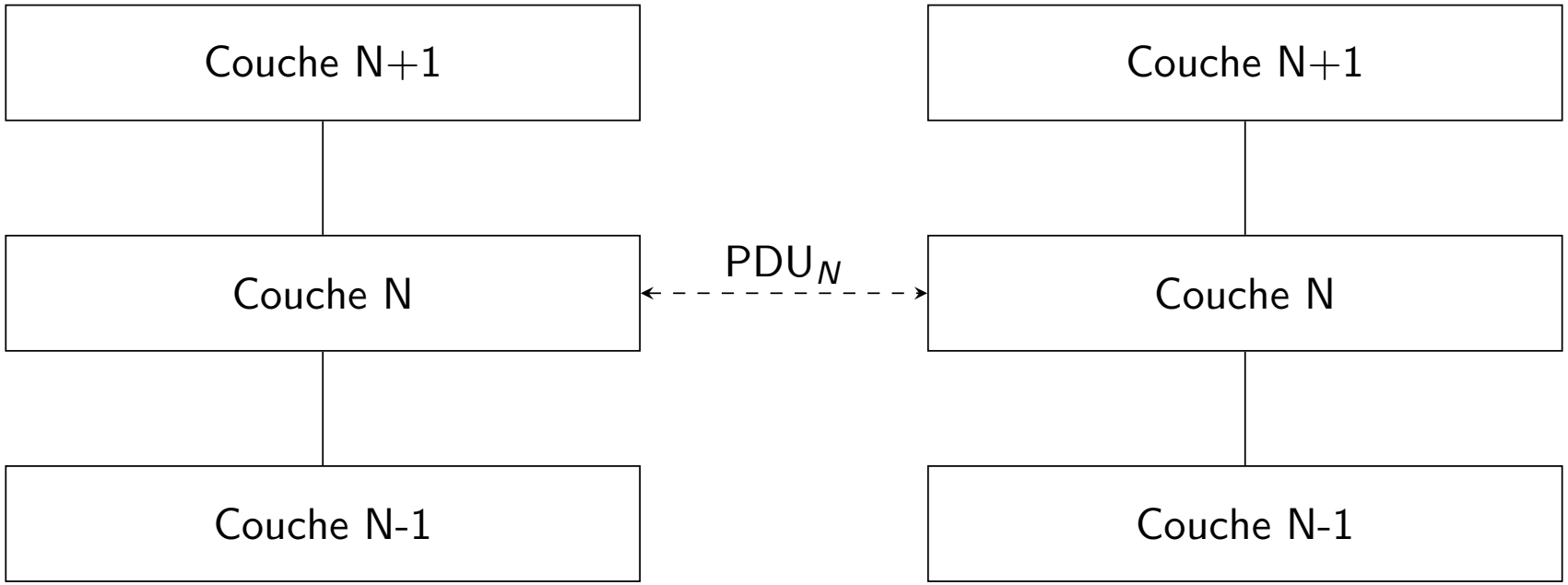


- Les services fournis par une couche *N* sont identifiés par des SAP (Service Access Point).
- L'interface définit les opérations fondamentales et les services que la couche inférieure offre à la couche supérieure.

PDU : Protocol Data Unit

Protocol Data Unit de niveau N

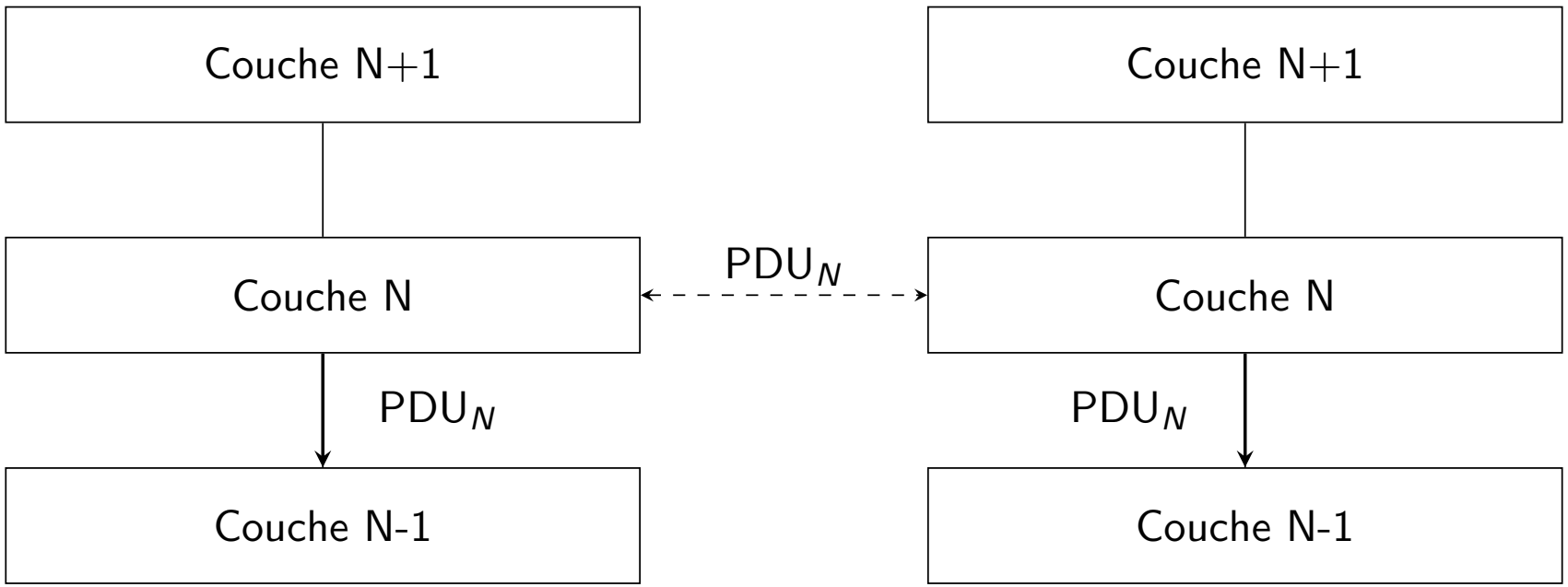
Les messages échangés par un protocole de niveau N (donc entre la couche N et $N-1$) sont appelés des PDU_N.



PDU : Protocol Data Unit

Protocol Data Unit de niveau N

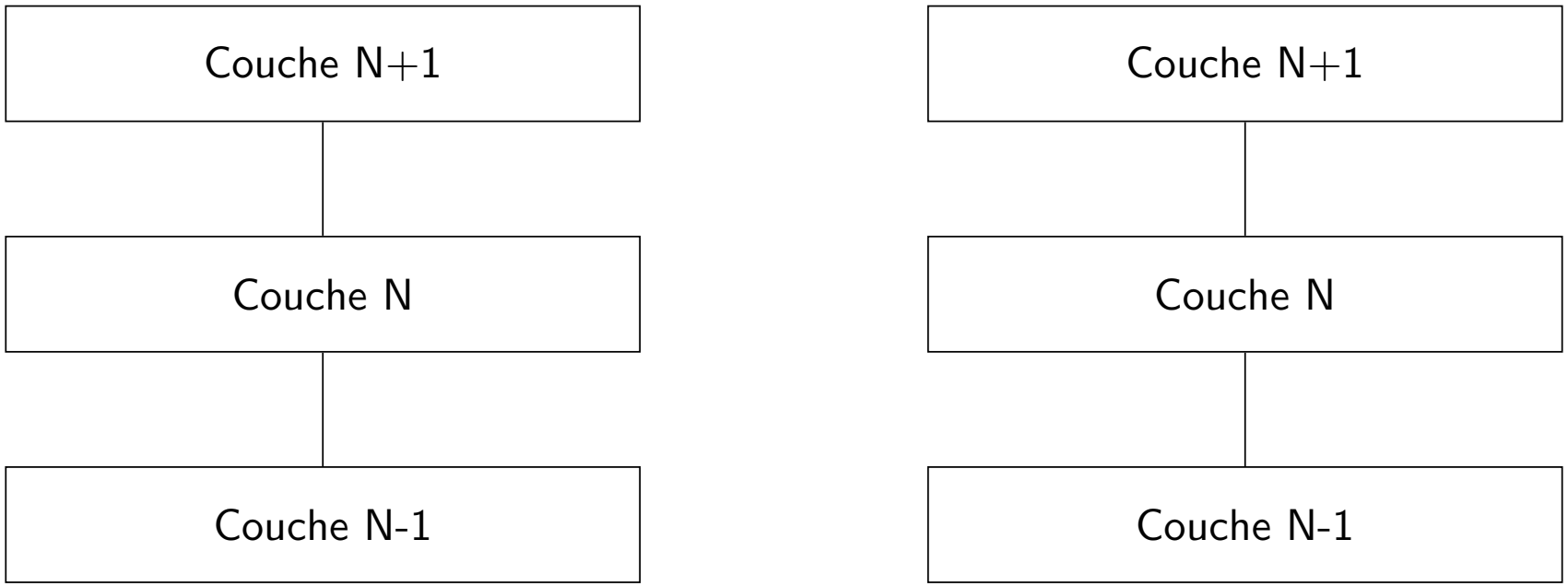
Les messages échangés par un protocole de niveau N (donc entre la couche N et $N-1$) sont appelés des PDU_N.



SDU : Service Data Unit

Service Data Unit de niveau N

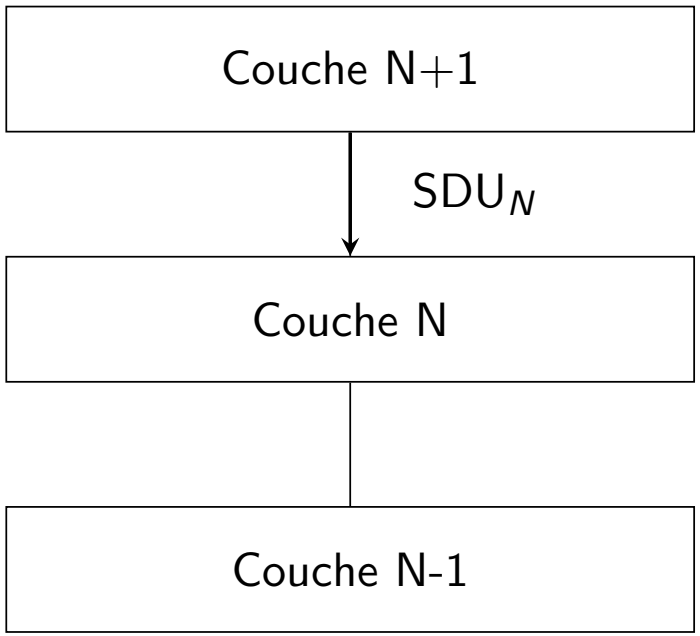
Les messages échangés entre la couche $N+1$ et la couche inférieure N sont appelés des SDU_N.

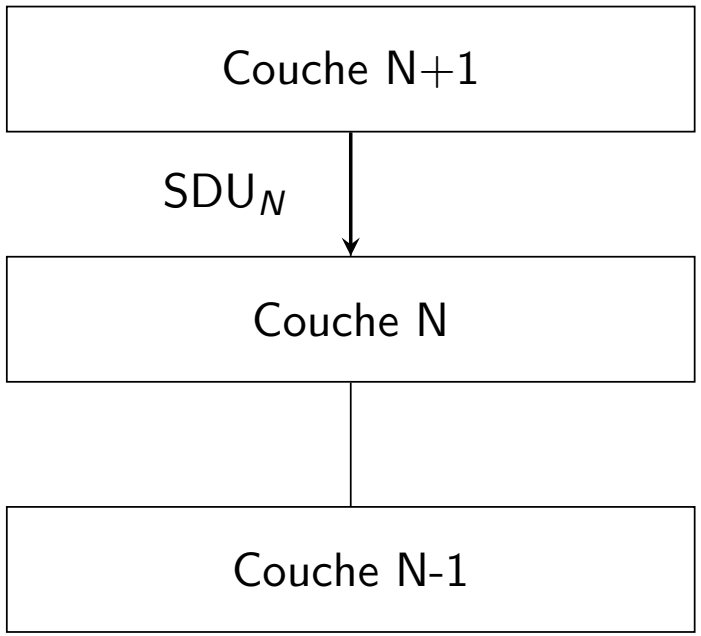


SDU : Service Data Unit

Service Data Unit de niveau N

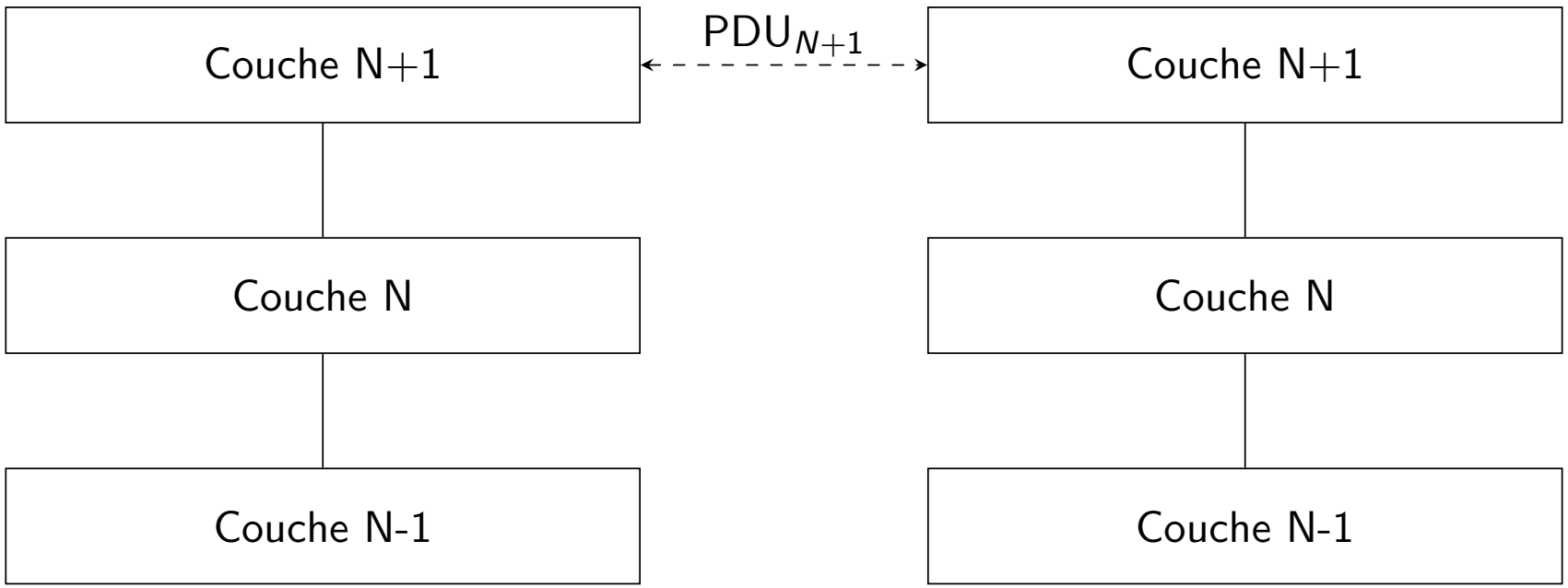
Les messages échangés entre la couche $N+1$ et la couche inférieure N sont appelés des SDU_N.





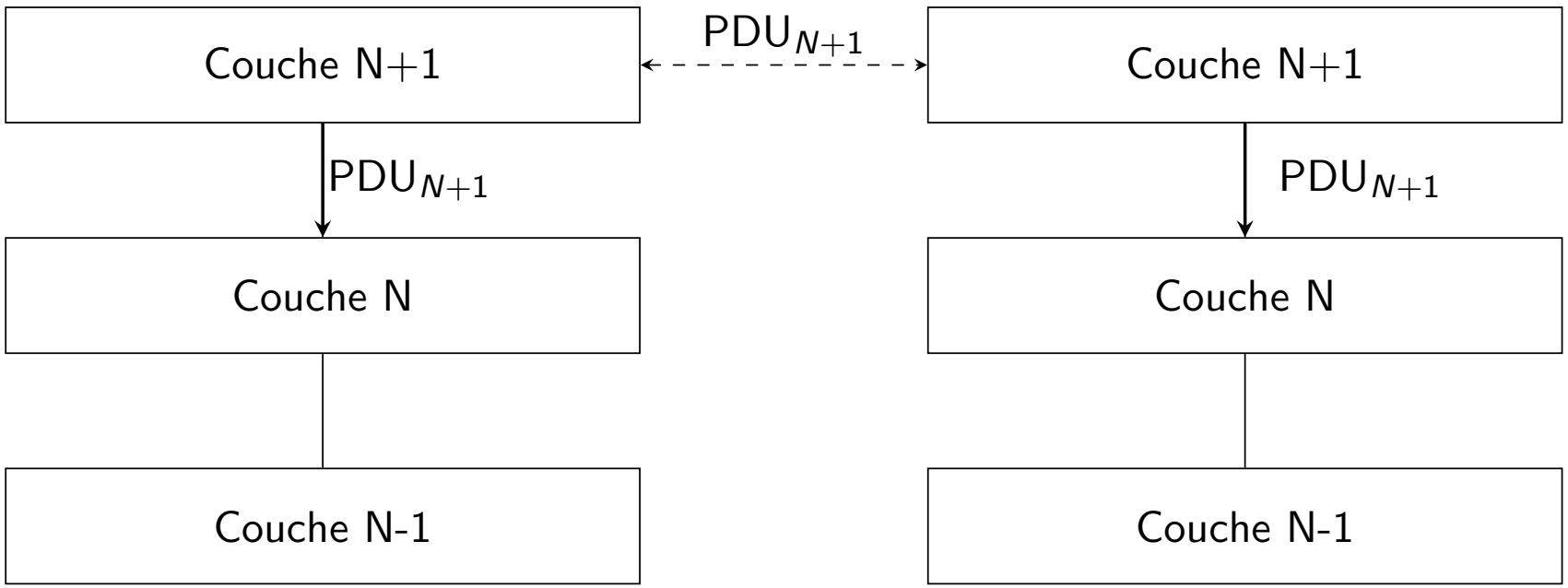
PDU et SDU

PDU vs SDU



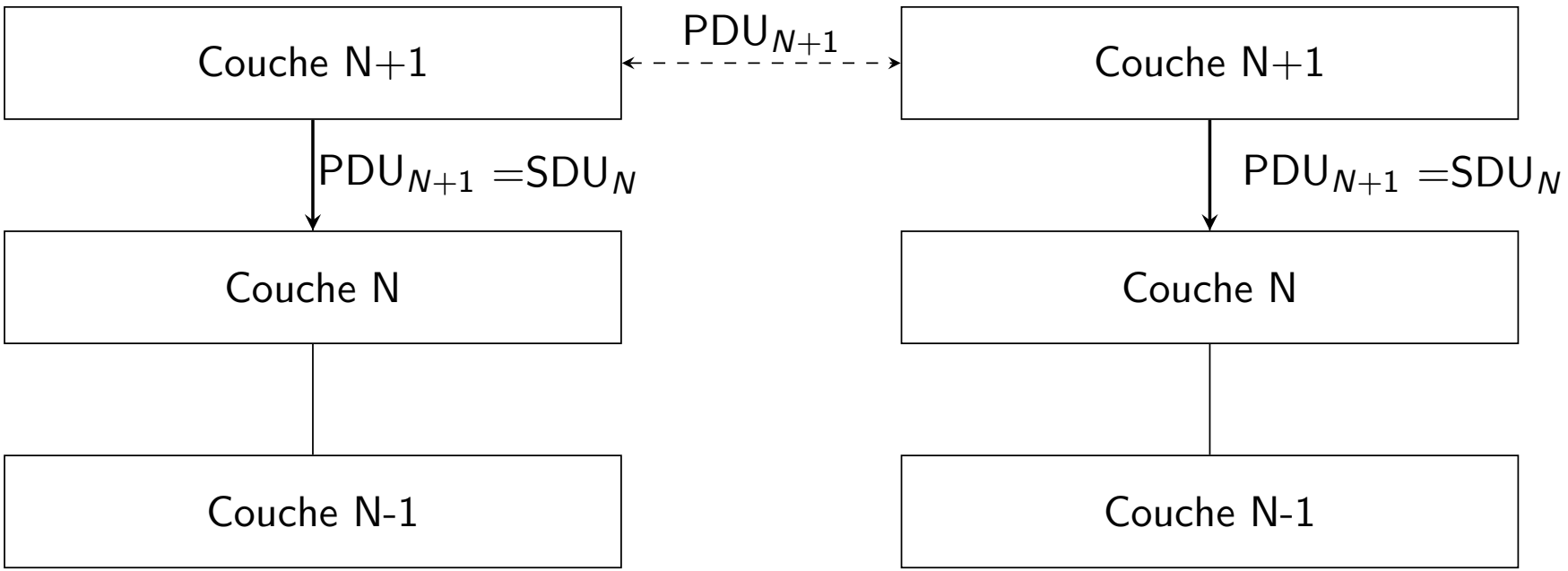
PDU et SDU

PDU vs SDU



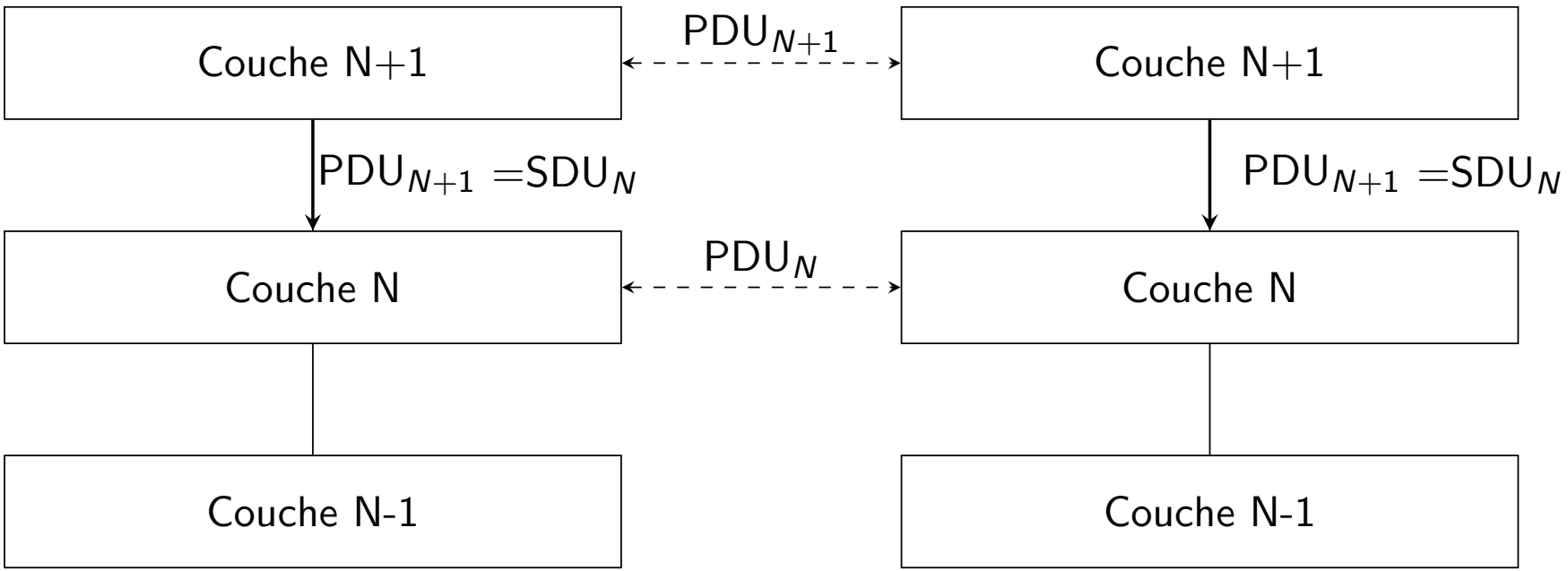
PDU et SDU

PDU vs SDU



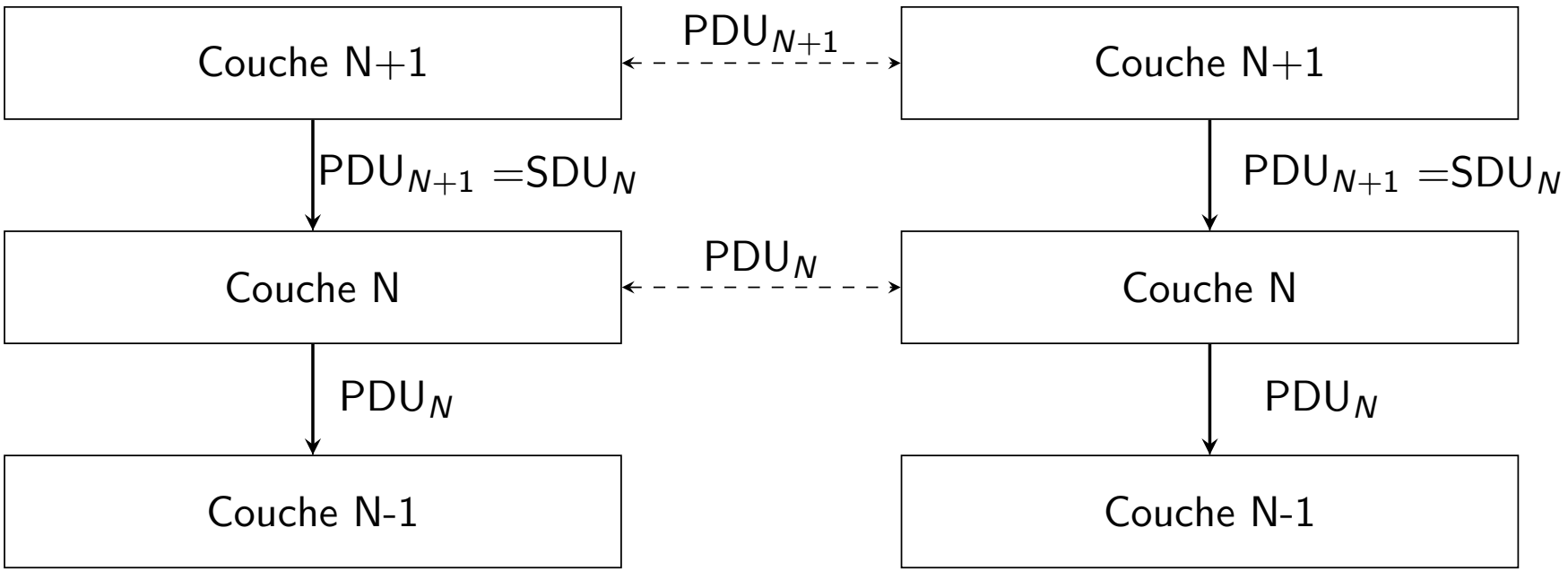
PDU et SDU

PDU vs SDU



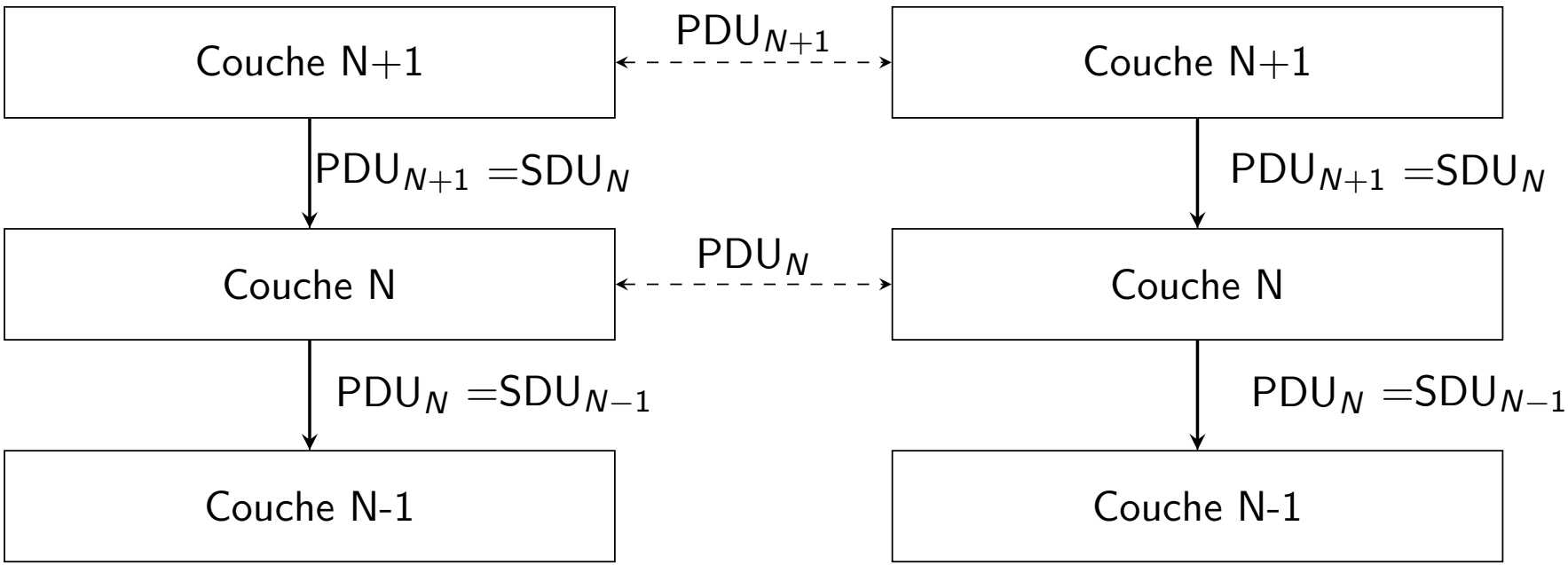
PDU et SDU

PDU vs SDU



PDU et SDU

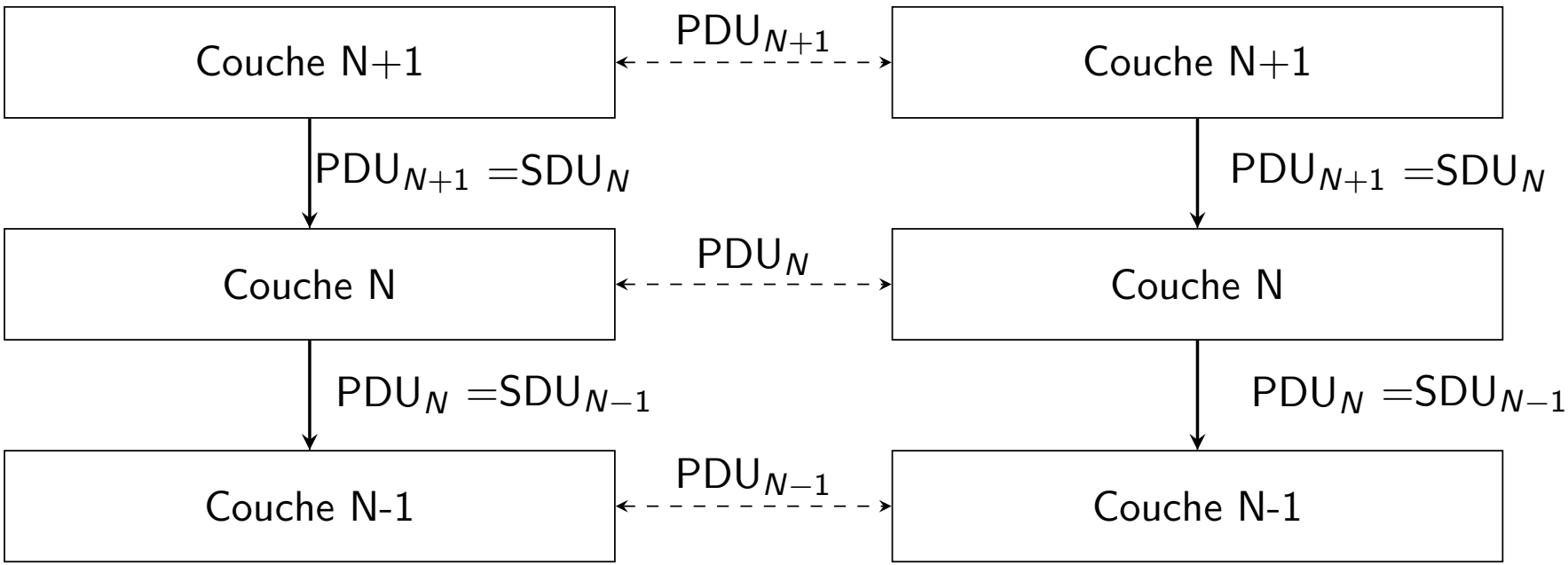
PDU vs SDU



PDU et SDU

PDU vs SDU

$$SDU_N = PDU_{N+1}$$



Encapsulation et PCI

Un protocole de niveau N ajoute au SDU_N qu'il a reçu des informations de contrôle visant à contrôler la bonne exécution du protocole.

Protocol Control Information de niveau N

Les informations de contrôle ajoutées par le protocole de niveau N au SDU_N sont appelées PCI_N.

- en-tête : PCI ajouté au début du message.
- en-queue : PCI ajouté à la fin du message.

Encapsulation et PCI

Un protocole de niveau N ajoute au SDU_N qu'il a reçu des informations de contrôle visant à contrôler la bonne exécution du protocole.

Protocol Control Information de niveau N

Les informations de contrôle ajoutées par le protocole de niveau N au SDU_N sont appelées PCI_N.

- en-tête : PCI ajouté au début du message.
- en-queue : PCI ajouté à la fin du message.

Encapsulation et PCI

Un protocole de niveau N ajoute au SDU_N qu'il a reçu des informations de contrôle visant à contrôler la bonne exécution du protocole.

Protocol Control Information de niveau N

Les informations de contrôle ajoutées par le protocole de niveau N au SDU_N sont appelées PCI_N.

- en-tête : PCI ajouté au début du message.
- en-queue : PCI ajouté à la fin du message.

Encapsulation

Donc nous avons :

$$PDU_N = SDU_N + PCI_N$$

On dit alors que le PDU_N encapsule le SDU_N.

Au lieu d'indexer le *PDU* ou le *SDU* par le numéro de la couche, on le fait souvent précéder de la première lettre du nom de la couche (en Anglais).

$\mathrm{NPDU} = \mathrm{PDU}_3$, où le N indique la couche réseau (*network*).

Encapsulation

Donc nous avons :

$$PDU_N = SDU_N + PCI_N$$

On dit alors que le PDU_N encapsule le SDU_N.

Au lieu d'indexer le *PDU* ou le *SDU* par le numéro de la couche, on le fait souvent précéder de la première lettre du nom de la couche (en Anglais).

$\mathrm{NPDU} = \mathrm{PDU}_3$, où le N indique la couche réseau (*network*).

Encapsulation

Donc nous avons :

$$PDU_N = SDU_N + PCI_N$$

On dit alors que le PDU_N encapsule le SDU_N.

Au lieu d'indexer le *PDU* ou le *SDU* par le numéro de la couche, on le fait souvent précéder de la première lettre du nom de la couche (en Anglais).

$\mathrm{NPDU} = \mathrm{PDU}_3$, où le N indique la couche réseau (*network*).

Exemple sur 5 couches

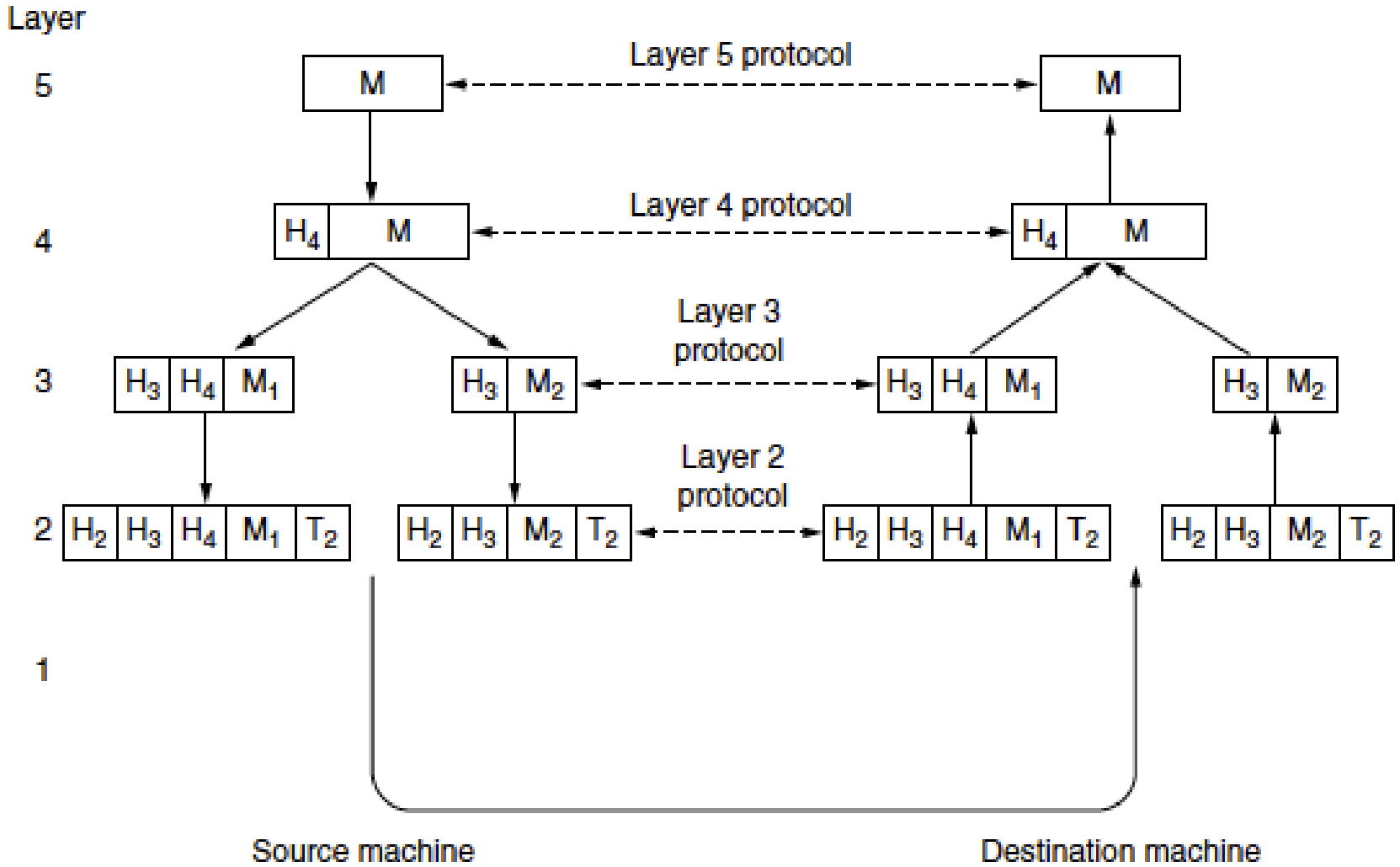


Figure 1-15. Example information flow supporting virtual communication in layer 5.

Services fiables/non-fiables

Les services peuvent être :

Fiables : il n'y a pas de perte de données,

Non-fiables : il peut y avoir de la perte de données,

Services confirmés/non-confirmés

Les services peuvent être :

Confirmés : le destinataire envoie un message d'acquittement à l'expéditeur pour confirmer la réception des données.

Non-confirmés : le destinataire reçoit les données sans donner de confirmation.

Mode connecté/deconnecté

Les protocoles peuvent opérer :

En mode connecté

La communication entre entités homologues de même niveau passe par l'établissement d'une connexion.

Le transfert des données comporte donc trois phases :

1. l'établissement de la connexion,
2. le transfert des données,
3. la déconnexion.

Le contexte de la communication est préservé.

En mode déconnecté

Seule la phase de transfert a lieu et la communication s'effectue sans mémoire.

Exemples

Mode	Service	Exemple
Connecté	séquence de messages fiable	séquence de pages
	séquence d'octets fiable	téléchargement d'un film
	connection non fiable	Voix sur IP
Déconnecté	datagramme non fiable	mails indésirables
	datagramme accusé	messages texte
	requête/réponse	base de données

Quelles sont les couches du modèle OSI ?

Le modèle OSI est composé de *sept couches*, qui sont organisées de la façon suivante :

Niveau	Couche	messages
7	Application	APDU
6	Présentation	PPDU
5	Session	SPDU
4	Transport	TPDU
3	Réseau	Paquet ou NPDU
2	Liaison de données	Trame
1	Physique	bits

Chaque couche a un rôle spécifique.

Les couches 4 à 7 sont dites « de bout en bout »

1. La couche physique

couche physique

La couche physique s'occupe de la transmission des bits sur un canal de communication.

Elle définit les moyens mécaniques, électriques et fonctionnels nécessaires :

- à l'activation,
- au maintien,
- à la désactivation,

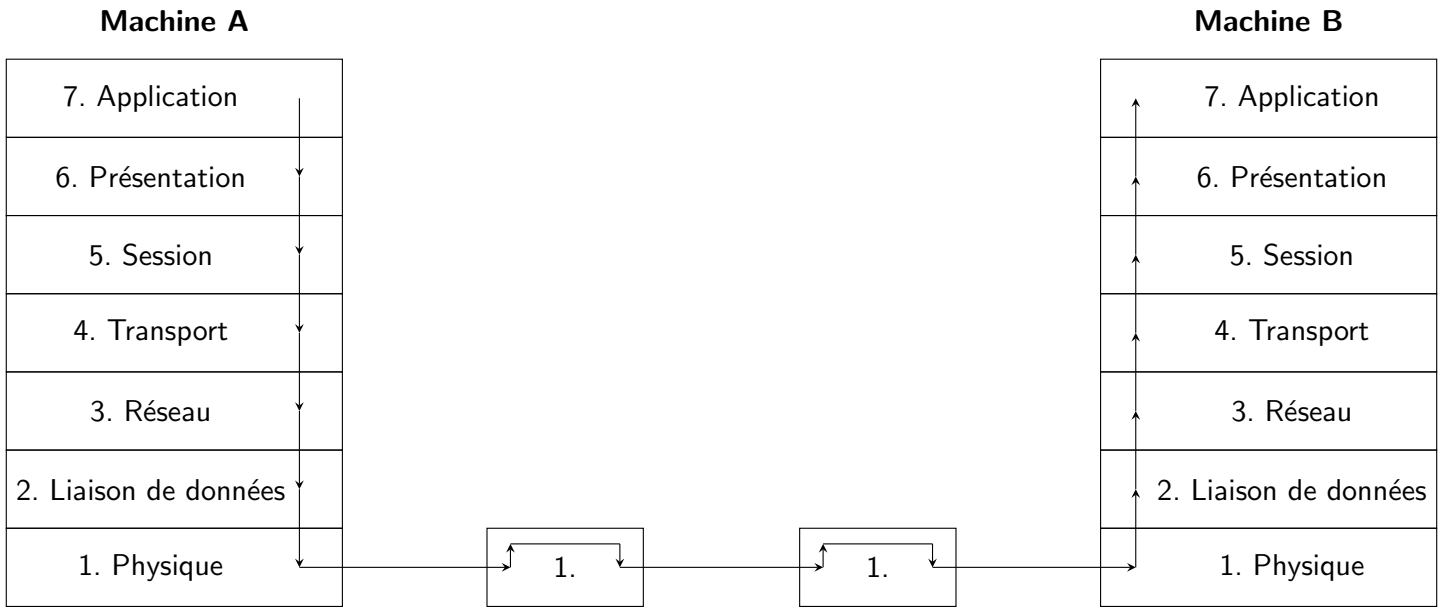
des connexions physiques destinées à la transmission des données binaires.

1. La couche physique

La couche physique fournit tous les éléments matériels et logiciels nécessaires au transport correct des données binaires, comme :

- les interfaces de connexion des équipements informatiques au support de transmission (*jonctions*),
- les supports de transmission,
- les cartes réseaux,
- les modems,
- les multiplexeurs (qui concentrent plusieurs communications sur une ligne de transmission unique).

Hub : équipement de niveau 1



Le hub opère au niveau 1

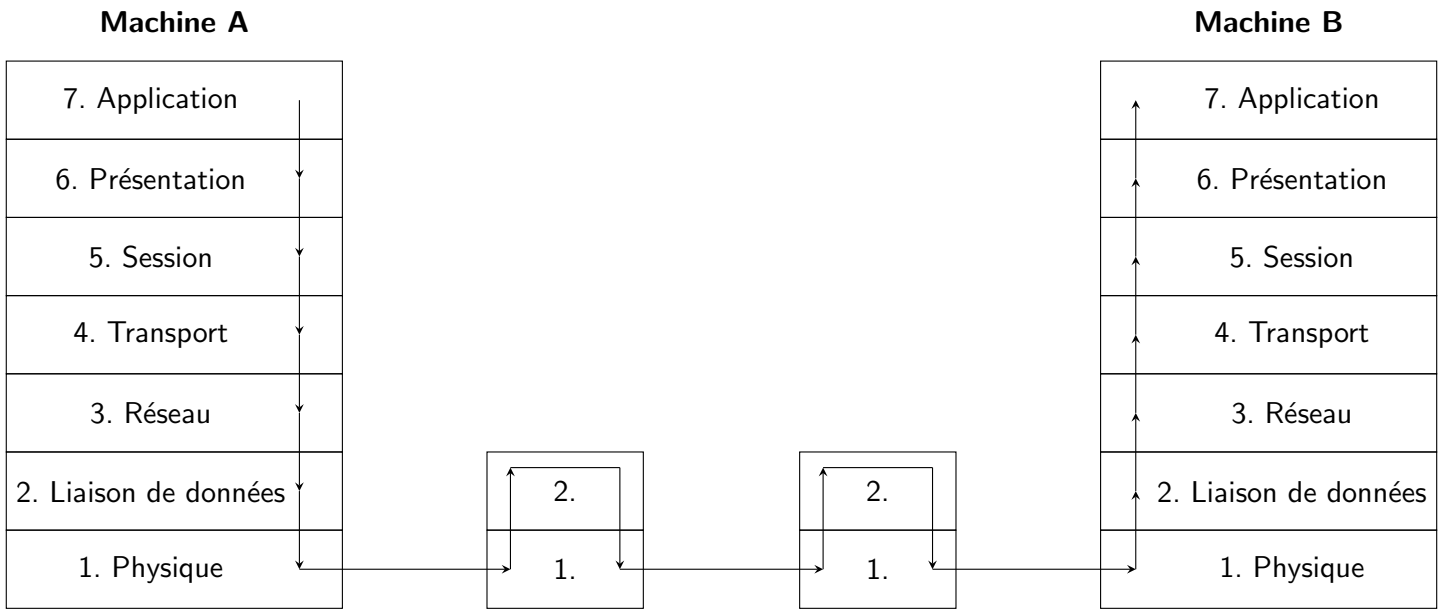
Il reçoit des signaux en entrée sur un port et les diffuse à tous ses ports. Tous les équipements connectés à un hub reçoivent l'intégralité du trafic même si les messages ne leur sont pas destinés.

Liaison de données

La couche liaison de données est responsable de l'acheminement sans erreur des blocs d'information entre les deux machines qui se trouvent aux extrémités d'une liaison de données.

Les messages de niveau 2 sont appelés trames.

Switch : équipement de niveau 2

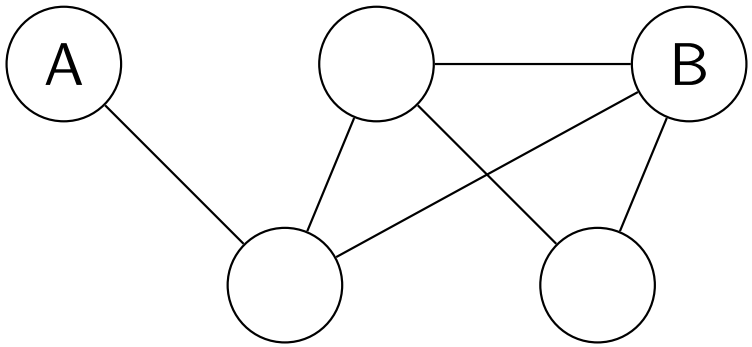


Le switch opère au niveau 2

Il maintient une table de commutation lui permettant de savoir quel port permet d'accéder aux stations du réseau.

Réseau

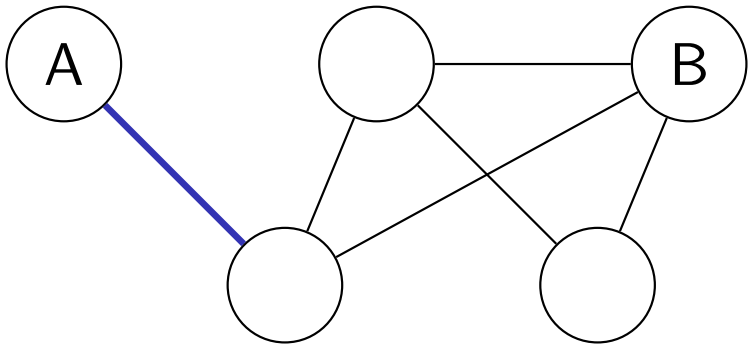
La couche réseau assure l'acheminement des blocs d'information à travers le sous-réseau. Elle choisit le meilleur chemin entre les deux commutateurs d'entrée-sortie du sous-réseau.



Les messages de niveau 3 sont appelés paquets.

Réseau

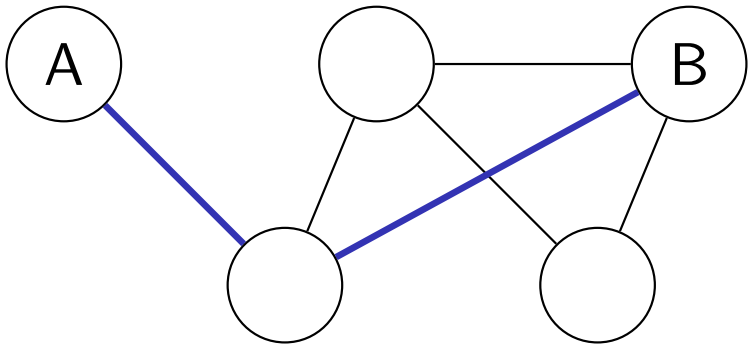
La couche réseau assure l'acheminement des blocs d'information à travers le sous-réseau. Elle choisit le meilleur chemin entre les deux commutateurs d'entrée-sortie du sous-réseau.



Les messages de niveau 3 sont appelés paquets.

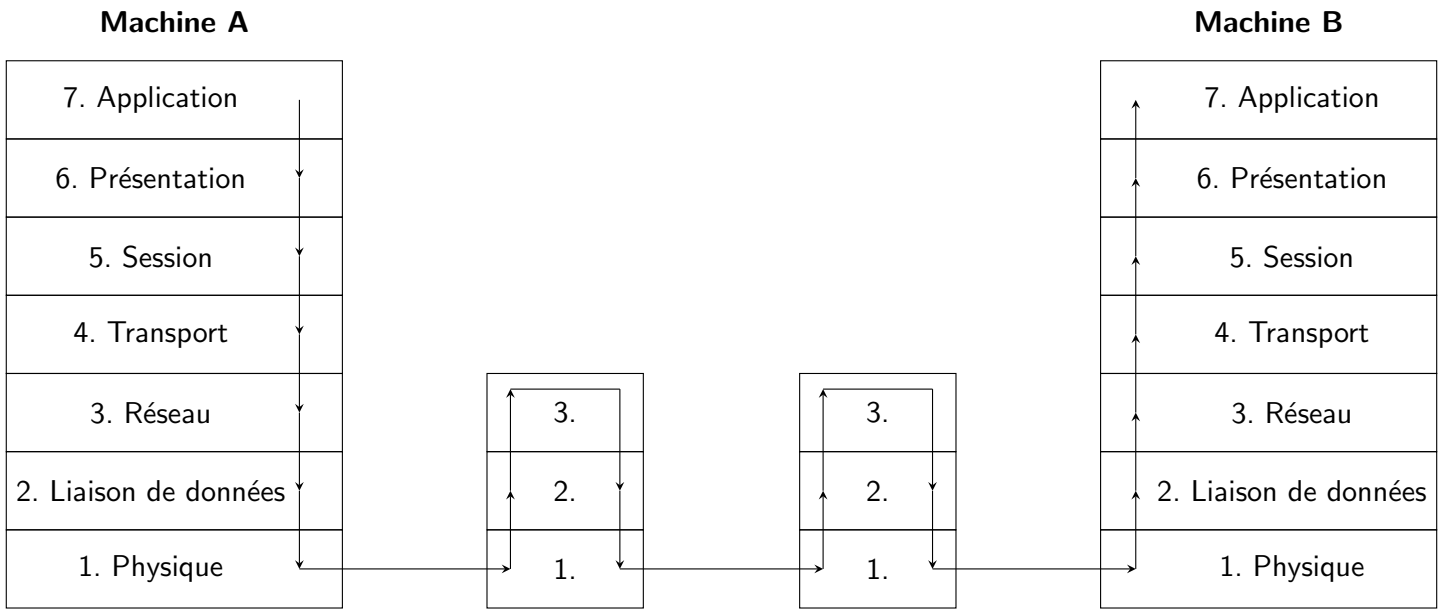
Réseau

La couche réseau assure l'acheminement des blocs d'information à travers le sous-réseau. Elle choisit le meilleur chemin entre les deux commutateurs d'entrée-sortie du sous-réseau.



Les messages de niveau 3 sont appelés paquets.

Routeur : équipement de niveau 3



Le routeur opère au niveau 3

Il décode les paquets reçus de manière à utiliser le meilleur chemin vers le destinataire.

4. La couche transport

Transport

La couche transport assure le transport de bout en bout, c'est-à-dire entre les deux stations qui communiquent.

Elle garantit que le message est acheminé entre les deux stations avec la qualité de service demandée.

QoS

Le terme qualité de service désigne un ensemble de propriétés que le demandeur du service exige du prestataire :

- la garantie d'un débit minimum,
- maximum de temps de livraison de messages, ...

5. La couche session

Session

La couche session contrôle le dialogue entre les machines qui communiquent.

Elle gère en particulier :

- la synchronisation du dialogue et
- la reprise après interruption.

6. La couche présentation

Présentation

La couche présentation réalise la compression et le chiffrement, et vérifie la syntaxe des données échangées.

7. La couche application

Application

La couche application offre aux utilisateurs des services normalisés pour la conception de leurs applications.

Cette couche offre ses services directement aux applications utilisées par l'usager.

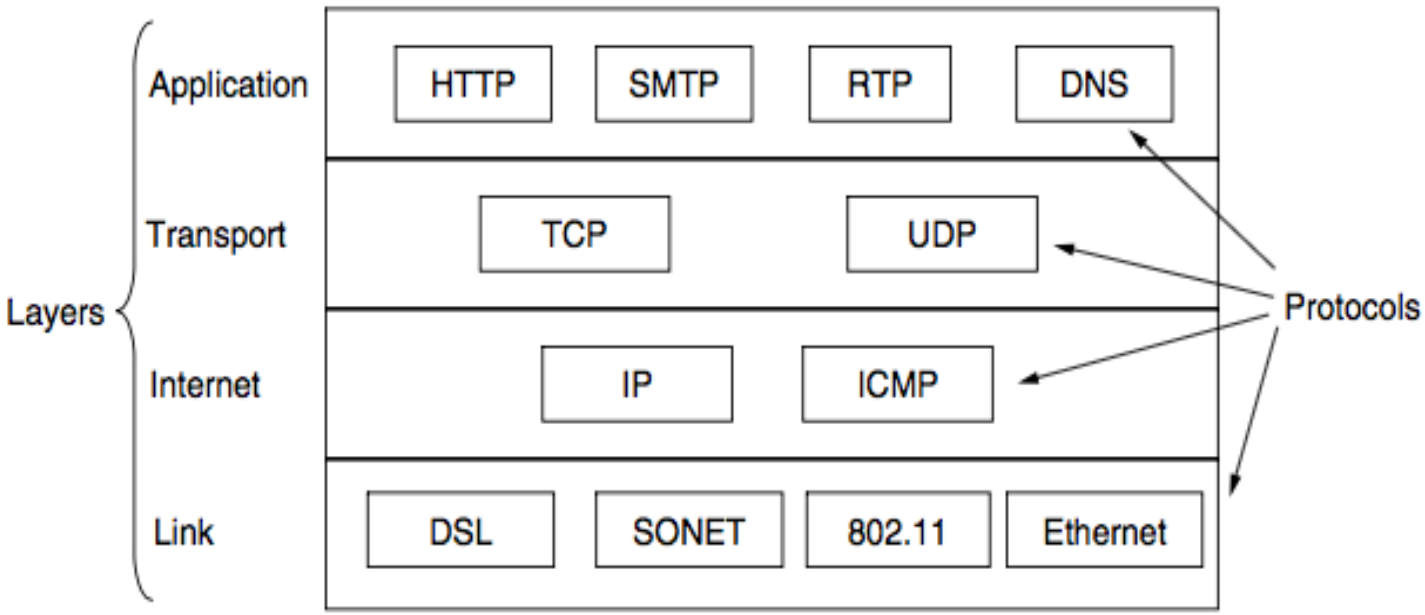
Le modèle TCP/IP

Le Modèle TCP/IP

Il définit un modèle de quatre couches.

Niveau	Couche
4	Application
3	Transport
2	Internet
1	Hôte-réseau

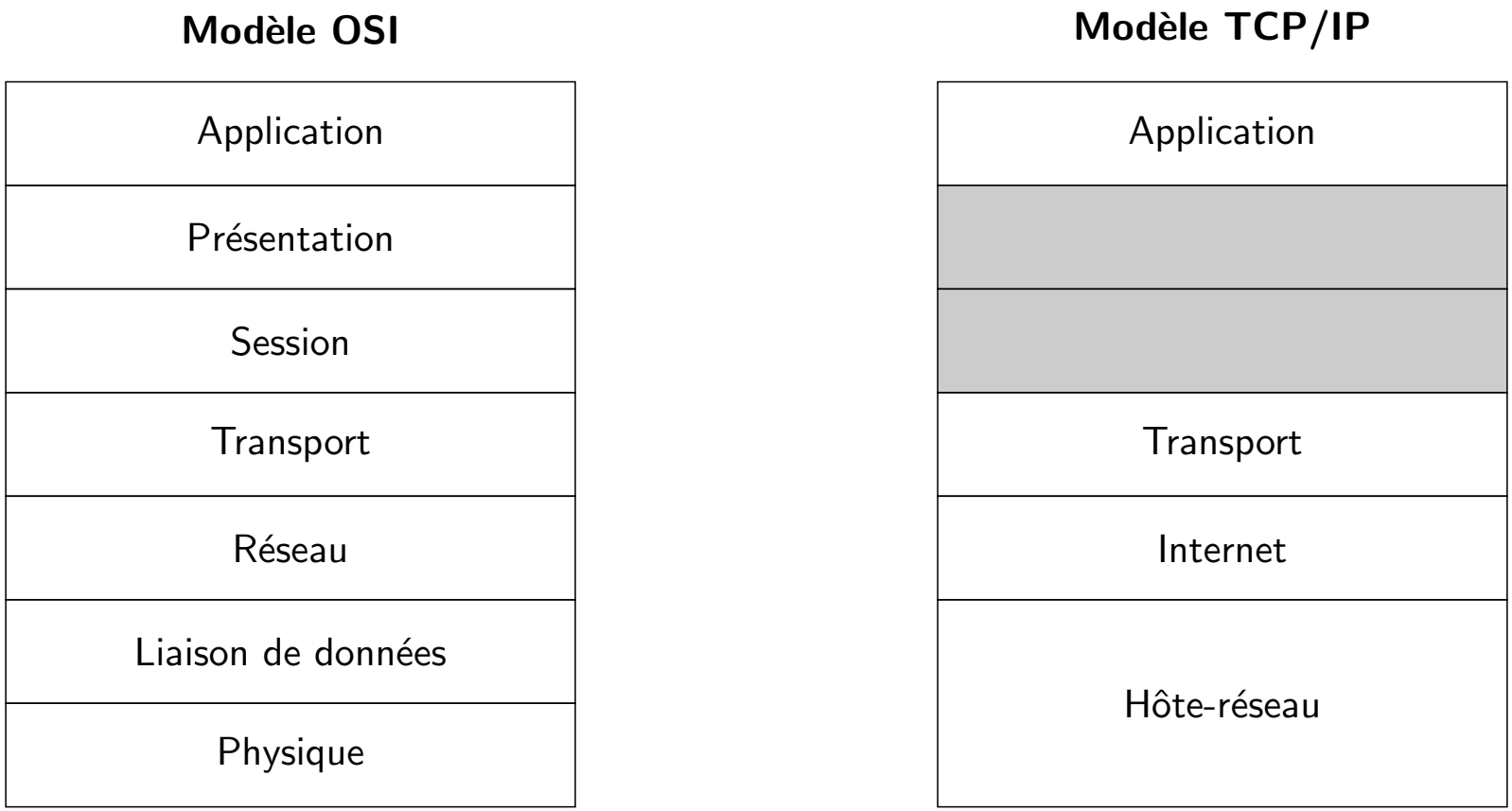
Les protocoles de TCP/IP



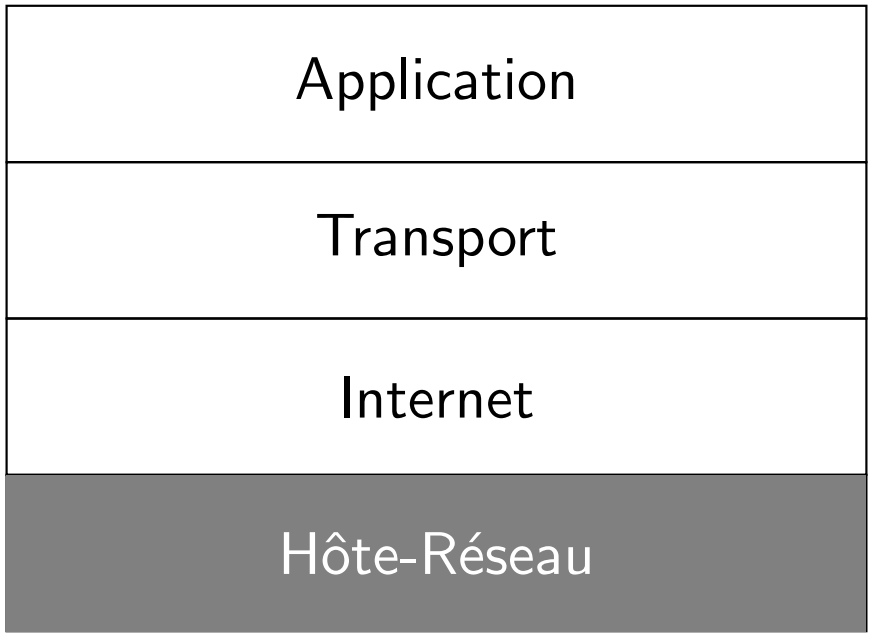
Le modèle TCP/IP

Le modèle TCP/IP est nommé d'après ses deux protocoles principaux TCP et IP, mais il comporte en réalité plusieurs dizaines de protocoles.

Modèle OSI vs Modèle TCP/IP



Modèle TCP/IP



1. La couche hôte-réseau

Les fonctions de la couche hôte-réseau sont :

- Offrir une interface de service à la couche Internet,
- Traiter les erreurs de transmission,
- Réguler le flux des données — éviter que les destinataires lents ne soient submergés par des expéditeurs rapides

Sous-couche Medium Access Control (MAC)

Sous-couche de contrôle d'accès au canal partagé.

- Système d'adressage basé sur les adresses MAC.

1. La couche hôte-réseau

Les fonctions de la couche hôte-réseau sont :

- Offrir une interface de service à la couche Internet,
- Traiter les erreurs de transmission,
- Réguler le flux des données — éviter que les destinataires lents ne soient submergés par des expéditeurs rapides

Sous-couche Medium Access Control (MAC)

Sous-couche de contrôle d'accès au canal partagé.

- Système d'adressage basé sur les adresses MAC.

Adressage MAC

Adresse MAC

Une *adresse MAC* (Media Access Control) est un identifiant physique codé sur 6 octets qui est enregistré dans la carte réseau.

Exemple d'adresse MAC

:

`00:18:DE:10:FA:87`

Remarques :

- Cette adresse identifie de façon unique l'interface réseau de la machine.
- Une machine avec plusieurs cartes réseaux aura plusieurs adresses MAC.
- Une adresse MAC est *unique* sur toute la planète.

Adressage MAC

Adresse MAC

Une *adresse MAC* (Media Access Control) est un identifiant physique codé sur 6 octets qui est enregistré dans la carte réseau.

Exemple d'adresse MAC

:

`00:18:DE:10:FA:87`

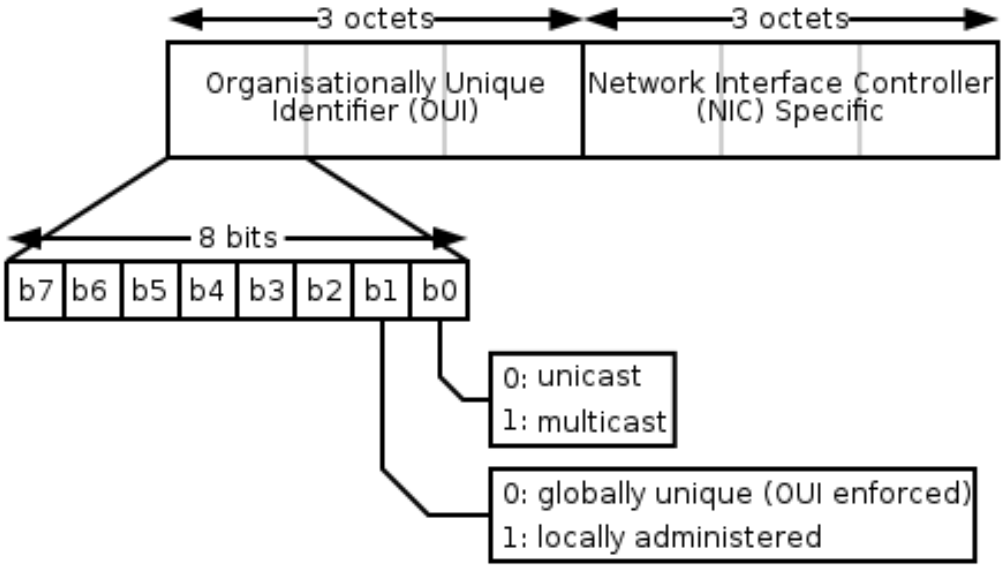
Remarques :

- Cette adresse identifie de façon unique l'interface réseau de la machine.
- Une machine avec plusieurs cartes réseaux aura plusieurs adresses MAC.
- Une adresse MAC est *unique* sur toute la planète.

Adresse MAC — Exemple

`00:18:DE:10:FA:87`

- les trois premiers octets constituent l'identifiant du constructeur de la carte (`00:18:DE` = Intel),



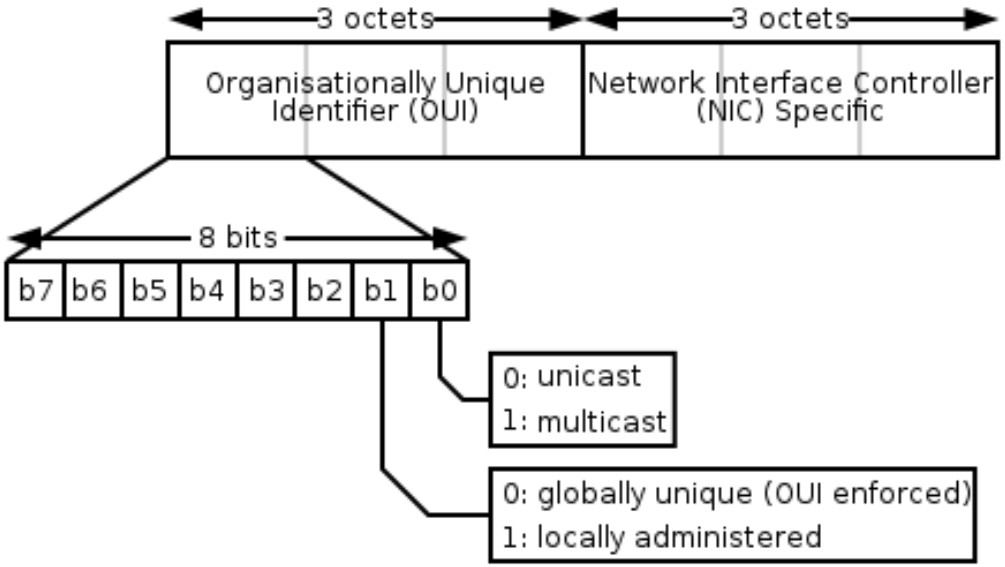
- les trois derniers octets représentent l'identifiant de la carte attribué par le constructeur.

Combien d'adresses peut attribuer chaque constructeur ?

Adresse MAC — Exemple

`00:18:DE:10:FA:87`

- les trois premiers octets constituent l'identifiant du constructeur de la carte (`00:18:DE` = Intel),

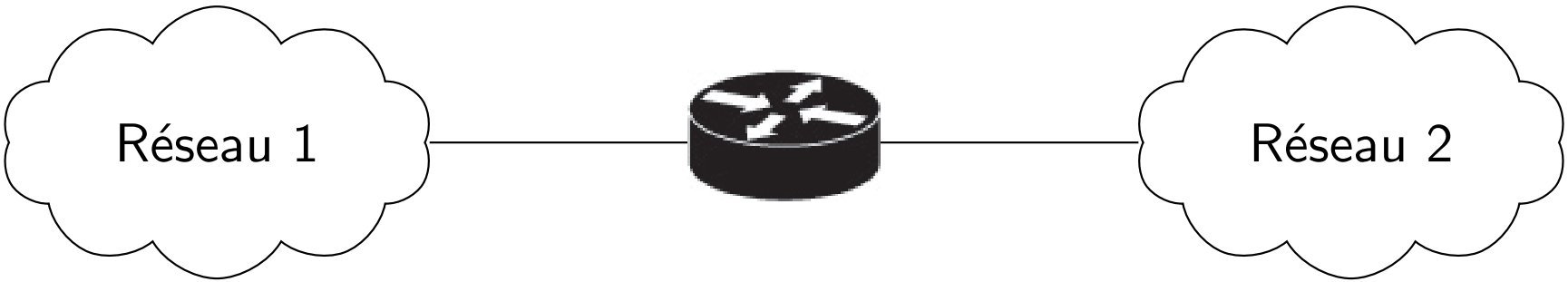


- les trois derniers octets représentent l'identifiant de la carte attribué par le constructeur.

Combien d'adresses peut attribuer chaque constructeur ? $2^{24} = 16\,777\,216$

Adresse MAC

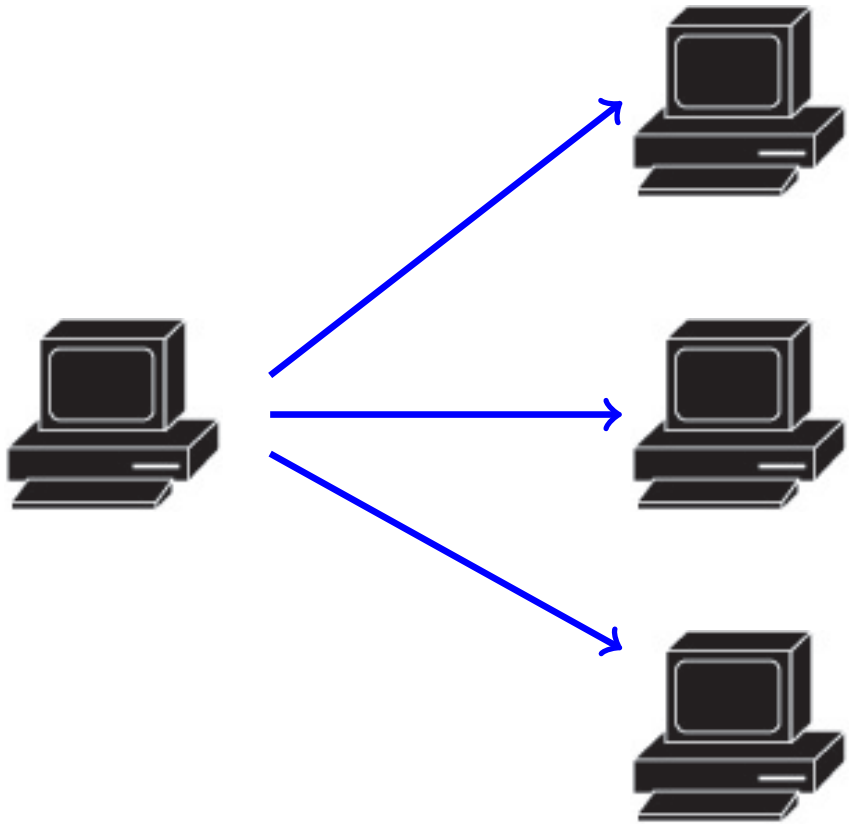
L'adresse MAC est utilisée pour identifier au niveau physique la carte réseau d'une machine sur un réseau local.



Remarque

Les messages qui circulent sur le Réseau 1 ne contiennent pas d'adresse MAC du Réseau 2.

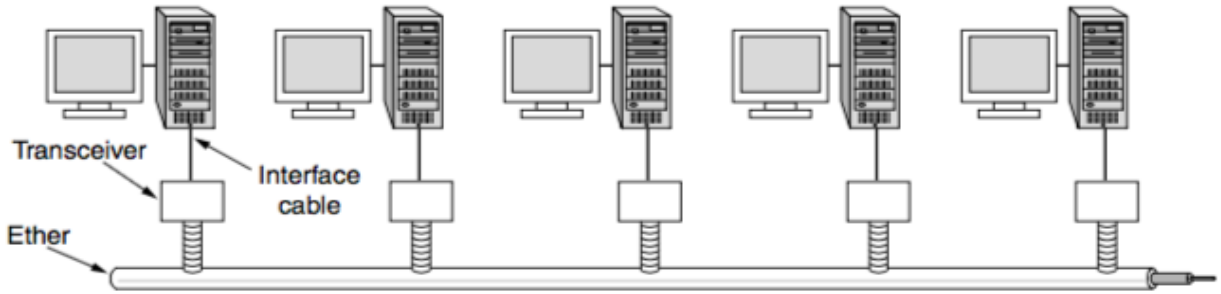
Adresse MAC de diffusion locale



Adresse MAC de diffusion locale : `FF:FF:FF:FF:FF:FF`

Ethernet - IEEE 802.3

Le protocole Ethernet repose sur une topologie physique et logique en bus :



Il y a donc plusieurs machines connectées au même câble.

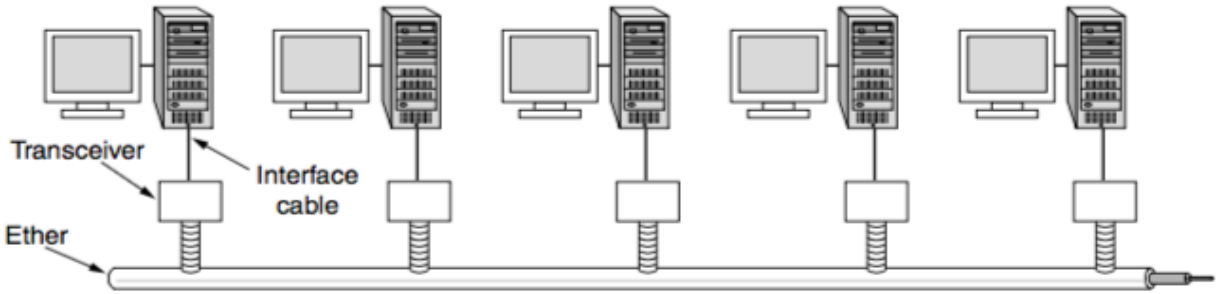
Lorsqu'une machine A envoie une trame à une machine B :

- toutes les stations du réseaux la reçoivent,
- seulement la machine B la traite (les autres l'ignorent).

Pour régler l'accès au canal physique Ethernet utilise une technologie appelée Carrier Sense Multiple Access with Collision Detection (CSMA/CD).

Ethernet - IEEE 802.3

Le protocole Ethernet repose sur une topologie physique et logique en bus :



Il y a donc plusieurs machines connectées au même câble.

Lorsqu'une machine A envoie une trame à une machine B :

- toutes les stations du réseaux la reçoivent,
- seulement la machine B la traite (les autres l'ignorent).

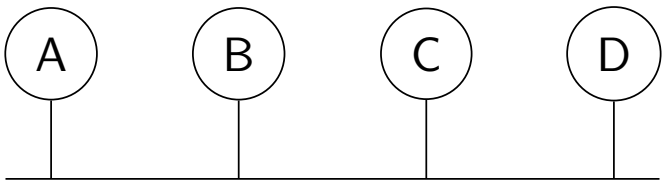
Pour régler l'accès au canal physique Ethernet utilise une technologie appelée Carrier Sense Multiple Access with Collision Detection (CSMA/CD).

CSMA/CD

Carrier Sense Multiple Access

Quand une station doit envoyer des données, elle exécute l'algorithme CSMA :

1. elle commence par écouter le support de transmission pour savoir si une autre station n'est pas déjà en train de transmettre.
2. Si c'est le cas, elle attend que le support se libère.
3. Quand le support est de nouveau disponible elle transmet sa trame.



CSMA/CD

Collision Detection

Si deux stations commencent une transmission au même moment elles provoquent une collision qu'elles détectent en écoutant le support.

Collision

La station émettrice d'une trame lance la procédure de gestion des collisions :

1. elle continue la transmission à pour s'assurer que toutes les stations détectent la collision,
2. elle observe un temps de pause aléatoire dépendant du nombre de tentatives de transmission,
3. elle redémarre l'envoi de la trame.

CSMA/CD

Collision Detection

Si deux stations commencent une transmission au même moment elles provoquent une collision qu'elles détectent en écoutant le support.

Collision

La station émettrice d'une trame lance la procédure de gestion des collisions :

1. elle continue la transmission à pour s'assurer que toutes les stations détectent la collision,
2. elle observe un temps de pause aléatoire dépendant du nombre de tentatives de transmission,
3. elle redémarre l'envoi de la trame.

Trame Ethernet

La couche Ethernet ajoute à son SDU :

- 22 octets d'en-tête
- 4 octets d'en-queue,

donc un total de 26 octets de PCI.

Si la taille des données est

< 46 octets alors on rajoute des *octets de bourrage (0000 0000)*.

> 1500 octets alors il faut envoyer plusieurs trames.

Trame Ethernet

La couche Ethernet ajoute à son SDU :

- 22 octets d'en-tête
- 4 octets d'en-queue,

donc un total de 26 octets de PCI.

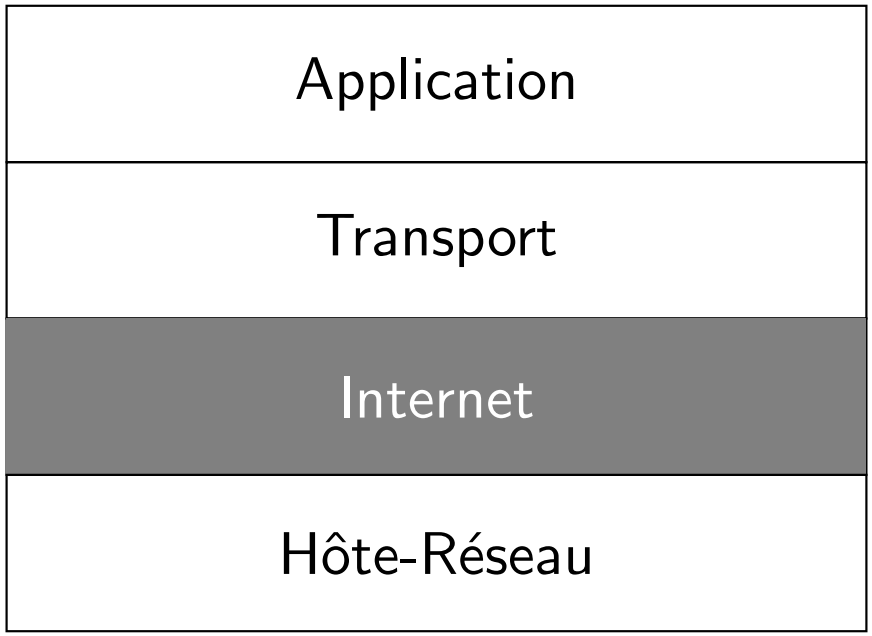
Si la taille des données est
< 46 octets alors on rajoute des *octets de bourrage (0000 0000)*.
> 1500 octets alors il faut envoyer plusieurs trames.

Structure d'une trame Ethernet

- Préambule (7o) : séquence de 7 octets à 1010 1010. Il permet la synchronisation des horloges de l'émetteur et du récepteur.
- SFD (Start Frame Delimiter, 1o) : un octet à 1010 1011 indiquant le début de la trame.
- DA (6o) : adresse MAC destination.
- SA (6o) : adresse MAC source.
- DL/Etype (2o) :
  - Si $\leq 1\,500 \Rightarrow$ c'est la taille des données (Ethernet I),
  - Si $> 1\,500 \Rightarrow$ c'est le type de la trame (Ethernet II).
- FCS (Frame Check Sequence, 4o) : utilisé par le code polynomial.

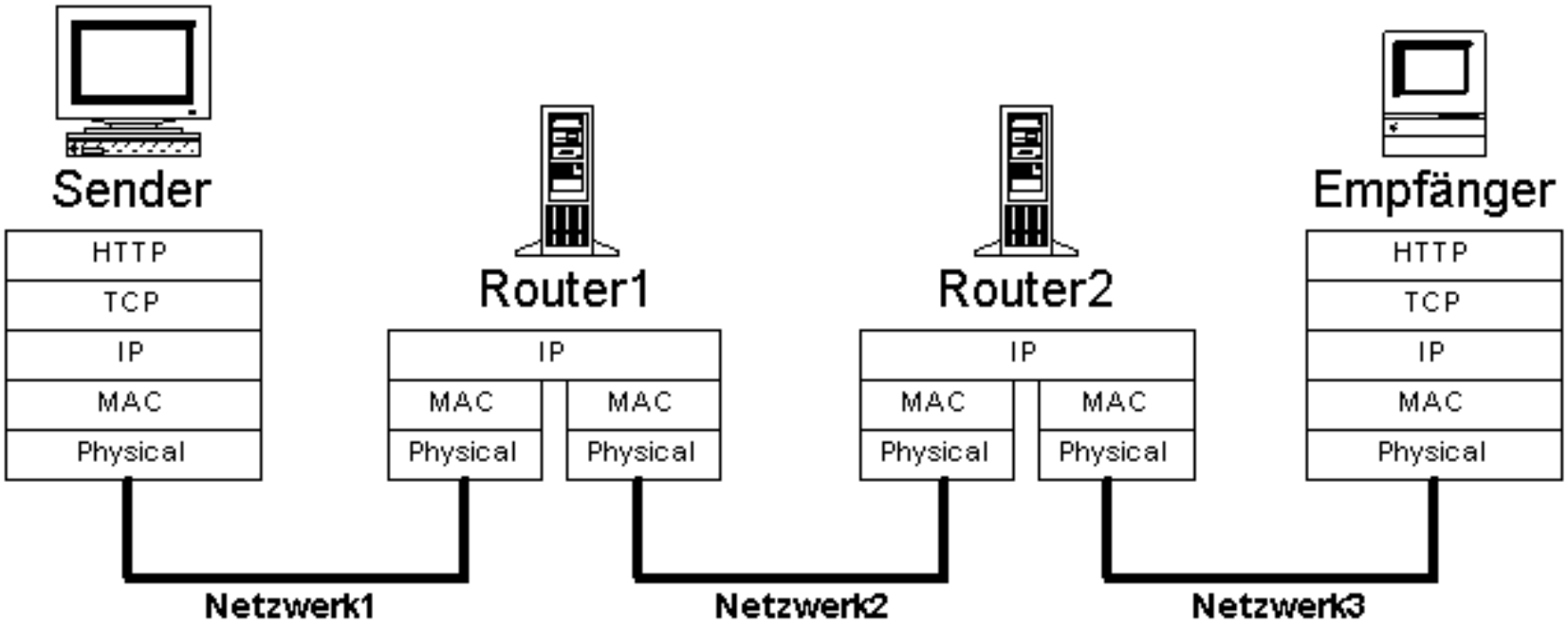
2. Couche Internet

Modèle TCP/IP



Internet Protocol

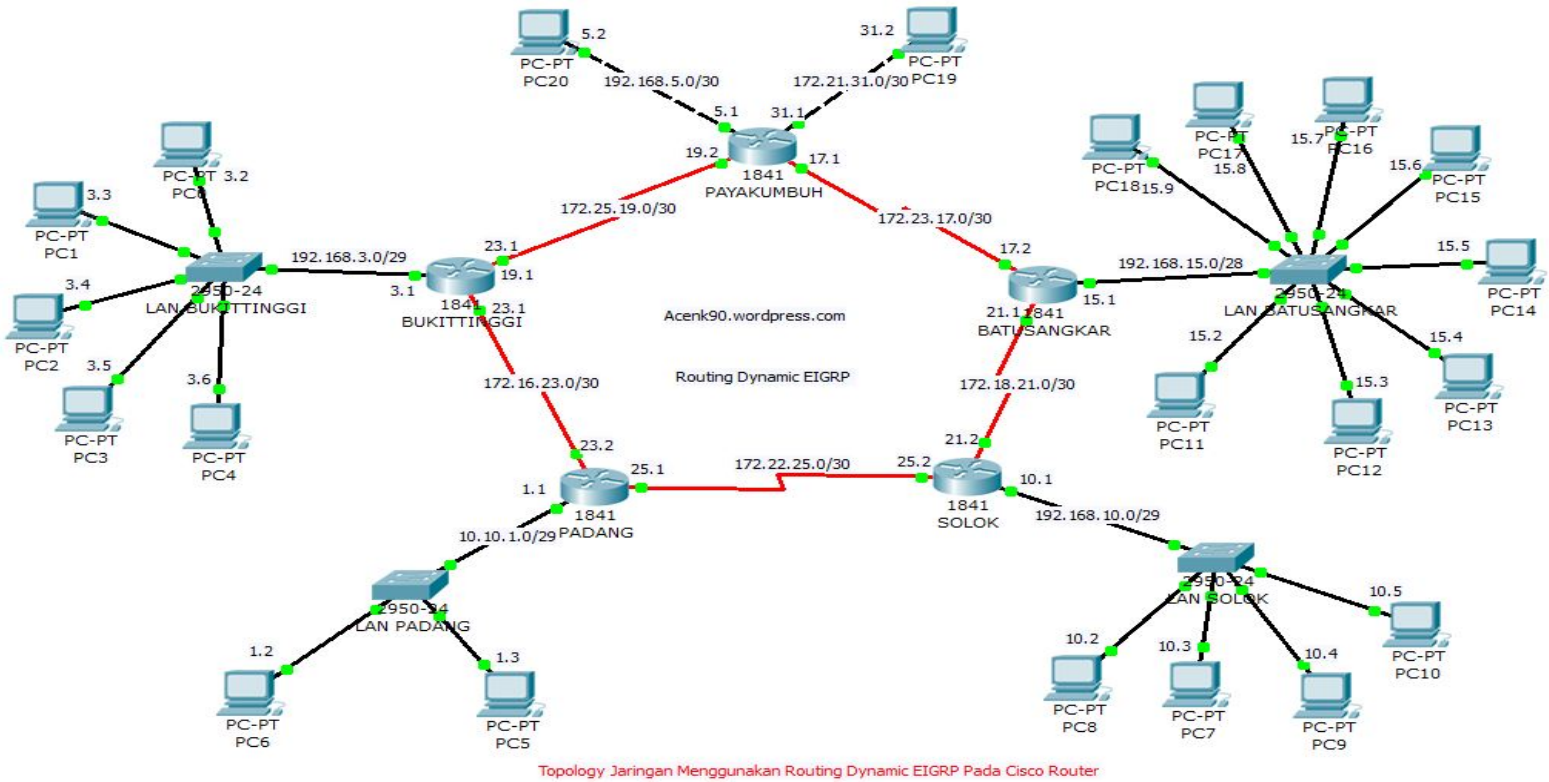
Entre la machine A (émetteur) et la machine B (destinataire) il peut y avoir un ou plusieurs routeurs.



Les routeurs sont des équipements de niveau 3 (couche réseau du modèle OSI)

Couche Internet : Routage

Problème : choisir la route suivant laquelle les paquets de données seront relayés de proche en proche jusqu'au destinataire.



Topology Jaringan Menggunakan Routing Dynamic EIGRP Pada Cisco Router

Internet Protocol (IP) fournit un système de livraison de paquets, sans connexion et non fiable.

Couche Internet : Routage

Routage

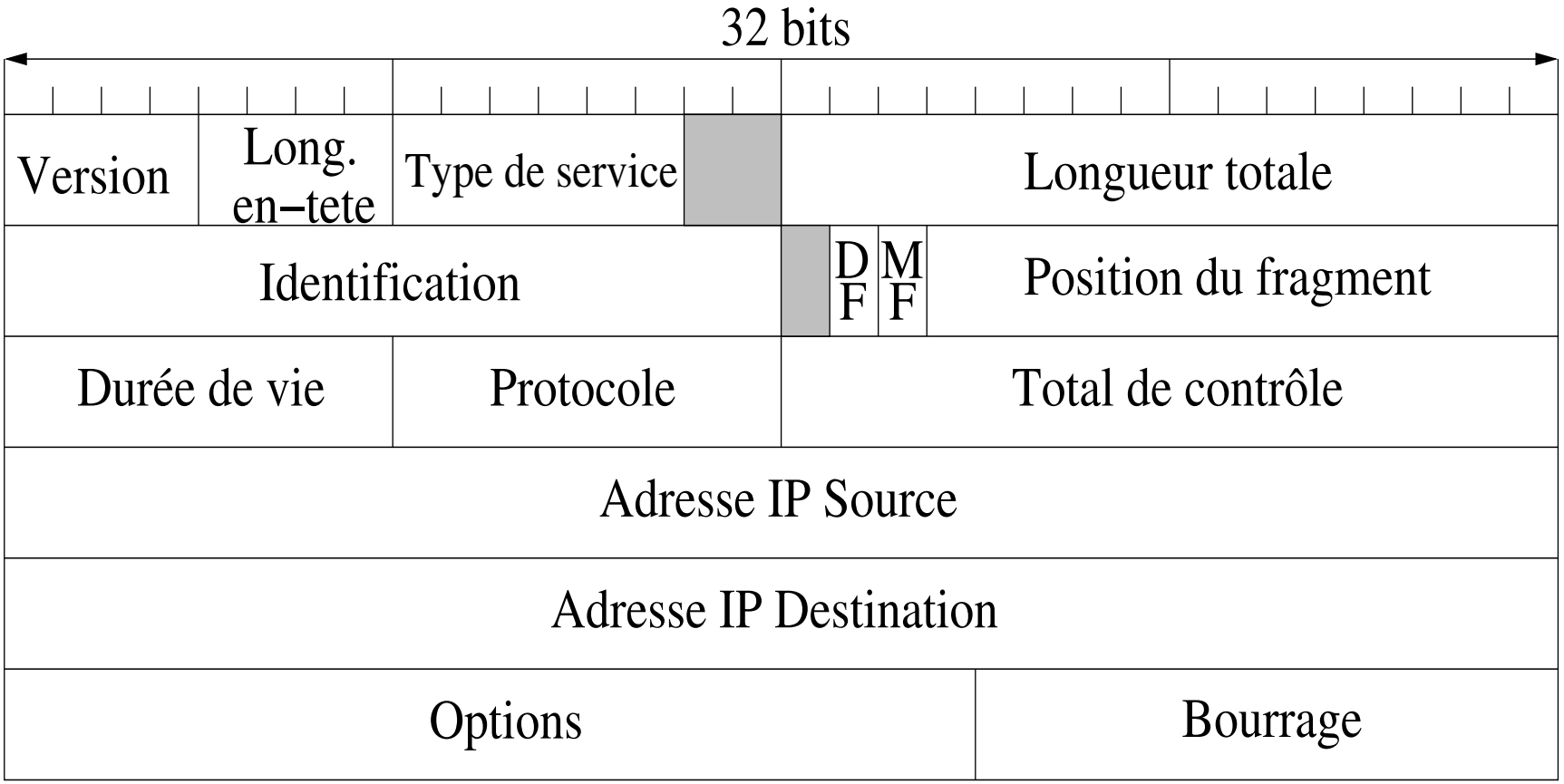
- Statique :
  - l'administrateur réseau doit initialiser la tables de routage de chaque routeur,
  - les routes ne changent que si l'administrateur les modifie manuellement.
- Dynamique :
  - les routes sont calculées automatiquement par un algorithme exécuté sur chaque routeur.
  - le protocole OSPF permet aux routeurs d'échanger des informations concernant l'état du réseau
  - les routeurs utilisent ces informations afin de calculer la meilleure route possible et mettre à jour leurs tables de routage.

Couche Internet : Routage

Routage

- Statique :
  - l'administrateur réseau doit initialiser la tables de routage de chaque routeur,
  - les routes ne changent que si l'administrateur les modifie manuellement.
- Dynamique :
  - les routes sont calculées automatiquement par un algorithme exécuté sur chaque routeur.
  - le protocole OSPF permet aux routeurs d'échanger des informations concernant l'état du réseau
  - les routeurs utilisent ces informations afin de calculer la meilleure route possible et mettre à jour leurs tables de routage.

Format du Paquet IP



Le paquet IP est encapsulé dans Ethernet/Token Ring

Adressage IP

Adresse IP

Une adresse IP (version 4) est un numéro d'identification de quatre octets qui est attribué de façon permanente ou provisoire à chaque carte réseau utilisant le protocole IP.

192.168.1.2

11000000	10101000	00000001	00000010

Une adresse IP se compose de deux parties :

- NET-ID = la partie réservée au réseau,
- HOST-ID = la partie réservée à la machine,

Les deux parties ont une longueur variable, qui dépend de la classe de l'adresse IP.

Adressage IP

Adresse IP

Une adresse IP (version 4) est un numéro d'identification de quatre octets qui est attribué de façon permanente ou provisoire à chaque carte réseau utilisant le protocole IP.

192.168.1.2

11000000	10101000	00000001	00000010

Une adresse IP se compose de deux parties :

- NET-ID = la partie réservée au réseau,
- HOST-ID = la partie réservée à la machine,

Les deux parties ont une longueur variable, qui dépend de la classe de l'adresse IP.

Classes IP

Classe A (Grands Réseaux) : le bit de poids fort est à 0.
Net-id = 1 octet, host-id = 3 octets.
- Plage : de 1.0.0.0 à 127.255.255.255.

Classe B (Réseaux Moyens) : les deux bits de poids fort sont 10.
Net-id = 2 octets, host-id = 2 octets.
- Plage : de 128.0.0.0 à 191.255.255.255.

Classe C (Petits Réseaux) : les trois bits de poids forts sont 110.
Net-id = 3 octets, host-id = 1 octet
- Plage : de 192.0.0.0 à 223.255.255.255.

Classe D (Multicast) : les quatre bits de poids forts sont 1110.
Les 28 bits qui restent désignent dans ce cas un groupe de multicast.
- Plage : de 224.0.0.0 à 239.255.255.255.

Classe E (Classe Réservée) : les cinq bits de poids forts sont 1111.
Classe Réservée pour des usages futurs.
- Plage : de 240.0.0.0 à 255.255.255.255.

Classes IP

Classe A (Grands Réseaux) : le bit de poids fort est à 0.
Net-id = 1 octet, host-id = 3 octets.
- Plage : de 1.0.0.0 à 127.255.255.255.

Classe B (Réseaux Moyens) : les deux bits de poids fort sont 10.
Net-id = 2 octets, host-id = 2 octets.
- Plage : de 128.0.0.0 à 191.255.255.255.

Classe C (Petits Réseaux) : les trois bits de poids forts sont 110.
Net-id = 3 octets, host-id = 1 octet
- Plage : de 192.0.0.0 à 223.255.255.255.

Classe D (Multicast) : les quatre bits de poids forts sont 1110.
Les 28 bits qui restent désignent dans ce cas un groupe de multicast.
- Plage : de 224.0.0.0 à 239.255.255.255.

Classe E (Classe Réservée) : les cinq bits de poids forts sont 1111.
Classe Réservée pour des usages futurs.
- Plage : de 240.0.0.0 à 255.255.255.255.

Classes IP

Classe A (Grands Réseaux) : le bit de poids fort est à 0.
Net-id = 1 octet, host-id = 3 octets.
- Plage : de 1.0.0.0 à 127.255.255.255.

Classe B (Réseaux Moyens) : les deux bits de poids fort sont 10.
Net-id = 2 octets, host-id = 2 octets.
- Plage : de 128.0.0.0 à 191.255.255.255.

Classe C (Petits Réseaux) : les trois bits de poids forts sont 110.
Net-id = 3 octets, host-id = 1 octet
- Plage : de 192.0.0.0 à 223.255.255.255.

Classe D (Multicast) : les quatre bits de poids forts sont 1110.
Les 28 bits qui restent désignent dans ce cas un groupe de multicast.
- Plage : de 224.0.0.0 à 239.255.255.255.

Classe E (Classe Réservée) : les cinq bits de poids forts sont 1111.
Classe Réservée pour des usages futurs.
- Plage : de 240.0.0.0 à 255.255.255.255.

Classes IP

Classe A (Grands Réseaux) : le bit de poids fort est à 0.
Net-id = 1 octet, host-id = 3 octets.
- Plage : de 1.0.0.0 à 127.255.255.255.

Classe B (Réseaux Moyens) : les deux bits de poids fort sont 10.
Net-id = 2 octets, host-id = 2 octets.
- Plage : de 128.0.0.0 à 191.255.255.255.

Classe C (Petits Réseaux) : les trois bits de poids forts sont 110.
Net-id = 3 octets, host-id = 1 octet
- Plage : de 192.0.0.0 à 223.255.255.255.

Classe D (Multicast) : les quatre bits de poids forts sont 1110.
Les 28 bits qui restent désignent dans ce cas un groupe de multicast.
- Plage : de 224.0.0.0 à 239.255.255.255.

Classe E (Classe Réservée) : les cinq bits de poids forts sont 1111.
Classe Réservée pour des usages futurs.
- Plage : de 240.0.0.0 à 255.255.255.255.

Classes IP

Classe A (Grands Réseaux) : le bit de poids fort est à 0.
Net-id = 1 octet, host-id = 3 octets.
- Plage : de 1.0.0.0 à 127.255.255.255.

Classe B (Réseaux Moyens) : les deux bits de poids fort sont 10.
Net-id = 2 octets, host-id = 2 octets.
- Plage : de 128.0.0.0 à 191.255.255.255.

Classe C (Petits Réseaux) : les trois bits de poids forts sont 110.
Net-id = 3 octets, host-id = 1 octet
- Plage : de 192.0.0.0 à 223.255.255.255.

Classe D (Multicast) : les quatre bits de poids forts sont 1110.
Les 28 bits qui restent désignent dans ce cas un groupe de multicast.
- Plage : de 224.0.0.0 à 239.255.255.255.

Classe E (Classe Réservée) : les cinq bits de poids forts sont 1111.
Classe Réservée pour des usages futurs.
- Plage : de 240.0.0.0 à 255.255.255.255.

Adresses réservées

Certaines adresses IP sont réservées pour des usages particuliers :

- 255.255.255.255 : adresse de diffusion locale.
- NET-ID $1 \cdots 1$: adresse de diffusion sur le sous-réseau identifié par le NET-ID.
- NET-ID $0 \cdots 0$: adresse d'un sous-réseau.
- 127.X.Y.Z : adresse de la boucle locale.
- 0.0.0.0 : Valeur qui indique l'absence d'adresse IP.

Si on veut changer la longueur du NET-ID il faut utiliser le masque.

Couche Internet

Chaque réseau a un certain MTU (*Maximum Transmission Unit*).
MTU = taille maximale d'un paquet pouvant circuler sur le réseau

Problème : et si la taille d'un paquet est plus grande que le MTU ?

taille en-tête IP + SDU_{IP} > MTU $\Rightarrow$ Problème

Couche Internet : Fragmentation

Chaque réseau a un certain MTU (*Maximum Transmission Unit*).
MTU = taille maximale d'un paquet pouvant circuler sur le réseau

Problème : et si la taille d'un paquet est plus grande que le MTU ?

Fragmentation

devient

où $SDU_{IP} = SDU1_{IP} + SDU2_{IP} + SDU3_{IP}$

Protocole ARP (Address Resolution Protocol, niv. 2/3)

ARP traduit une adresse réseau en une adresse physique :

$$\text{ARP : Adresse Logique} \rightarrow \text{Adresse Physique}$$

Typiquement : adresse IPv4 $\rightarrow$ adresse MAC.

Service

Permettre à une machine de trouver l'adresse physique d'une autre machine (connectée sur le même réseau IP) à partir de son adresse IP.

Niveau 2/3 : Violation du principe d'indépendance des couches.

Protocole ARP (Address Resolution Protocol, niv. 2/3)

ARP traduit une adresse réseau en une adresse physique :

$$\text{ARP : Adresse Logique} \rightarrow \text{Adresse Physique}$$

Typiquement : adresse IPv4 $\rightarrow$ adresse MAC.

Service

Permettre à une machine de trouver l'adresse physique d'une autre machine (connectée sur le même réseau IP) à partir de son adresse IP.

Niveau 2/3 : Violation du principe d'indépendance des couches.

Protocole ARP (Address Resolution Protocol, niv. 2/3)

Fonctionnement

- La machine source envoie un paquet spécifique : `ARP-Request`, en mode diffusion.
- Le paquet `ARP-Request` contient l'adresse IP de la machine recherchée ainsi que les adresses IP et physique de la machine source.
- Si la machine recherchée se trouve sur le réseau local alors elle répond directement à la machine source en envoyant un paquet `ARP-Reply` fournissant son adresse physique.

Un mécanisme de cache permet garder en mémoire (pour un temps limité) les correspondances obtenues entre adresses logiques et adresses physiques (commande `arp -a`).

ARP est encapsulé dans Ethernet/Token Ring

Protocole ARP (Address Resolution Protocol, niv. 2/3)

Fonctionnement

- La machine source envoie un paquet spécifique : `ARP-Request`, en mode diffusion.
- Le paquet `ARP-Request` contient l'adresse IP de la machine recherchée ainsi que les adresses IP et physique de la machine source.
- Si la machine recherchée se trouve sur le réseau local alors elle répond directement à la machine source en envoyant un paquet `ARP-Reply` fournissant son adresse physique.

Un mécanisme de cache permet garder en mémoire (pour un temps limité) les correspondances obtenues entre adresses logiques et adresses physiques (commande `arp -a`).

ARP est encapsulé dans Ethernet/Token Ring

Protocole ARP (Address Resolution Protocol, niv. 2/3)

Fonctionnement

- La machine source envoie un paquet spécifique : `ARP-Request`, en mode diffusion.
- Le paquet `ARP-Request` contient l'adresse IP de la machine recherchée ainsi que les adresses IP et physique de la machine source.
- Si la machine recherchée se trouve sur le réseau local alors elle répond directement à la machine source en envoyant un paquet `ARP-Reply` fournissant son adresse physique.

Un mécanisme de cache permet garder en mémoire (pour un temps limité) les correspondances obtenues entre adresses logiques et adresses physiques (commande `arp -a`).

ARP est encapsulé dans Ethernet/Token Ring

Format du message ARP

Type Mat.	Type Prot.	Long. Adr. Phy.	Long. Adr. Prot.	OP	Adr. Phy. Source	Adr. Prot. Source	Adr. Phy. Dest.	Adr. Prot. Dest.
2 o.	2 o.	1 o.	1 o.	2 o.	6 o.	4 o.	6 o.	4 o.

- Type Mat. : type de materiel physique (2o),
- Type Prot. : type de protocole (2o),
- Long. Adr. Phy. : longueur de l'adresse physique (1o),
- Long. Adr. Prot. : longueur de l'adresse du protocole (1o),
- OP : opération (2o),
- Adr. Phy. Source : adresse physique de l'émetteur (6o),
- Adr. Prot. Source : adresse du protocole de l'émetteur (4o),
- Adr. Phy. Dest. : adresse physique du destinataire (6o),
- Adr. Prot. Dest. : adresse du protocole du destinataire (4o).

Structure de la trame ARP

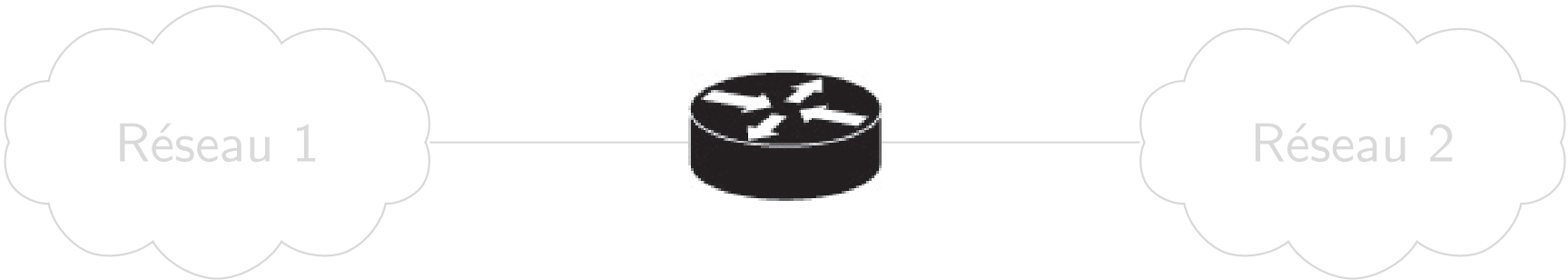
Le message ARP est encapsulé dans Ethernet (ou Token Ring) :

en-tête Ethernet	message ARP	en-queue Ethernet

Remarque : Il n'y a pas d'en-tête IP !

Remarque

Les messages ARP ne traversent jamais un routeur.



Structure de la trame ARP

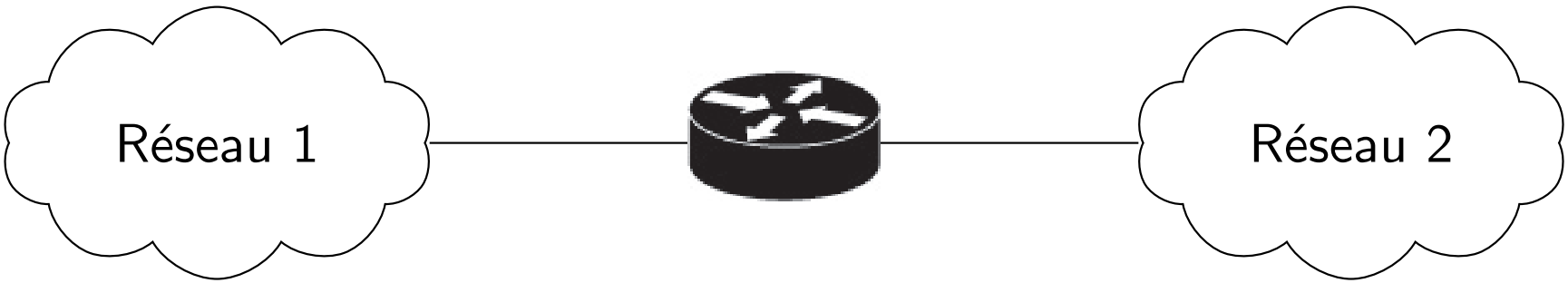
Le message ARP est encapsulé dans Ethernet (ou Token Ring) :

en-tête Ethernet	message ARP	en-queue Ethernet

Remarque : Il n'y a pas d'en-tête IP !

Remarque

Les messages ARP ne traversent jamais un routeur.

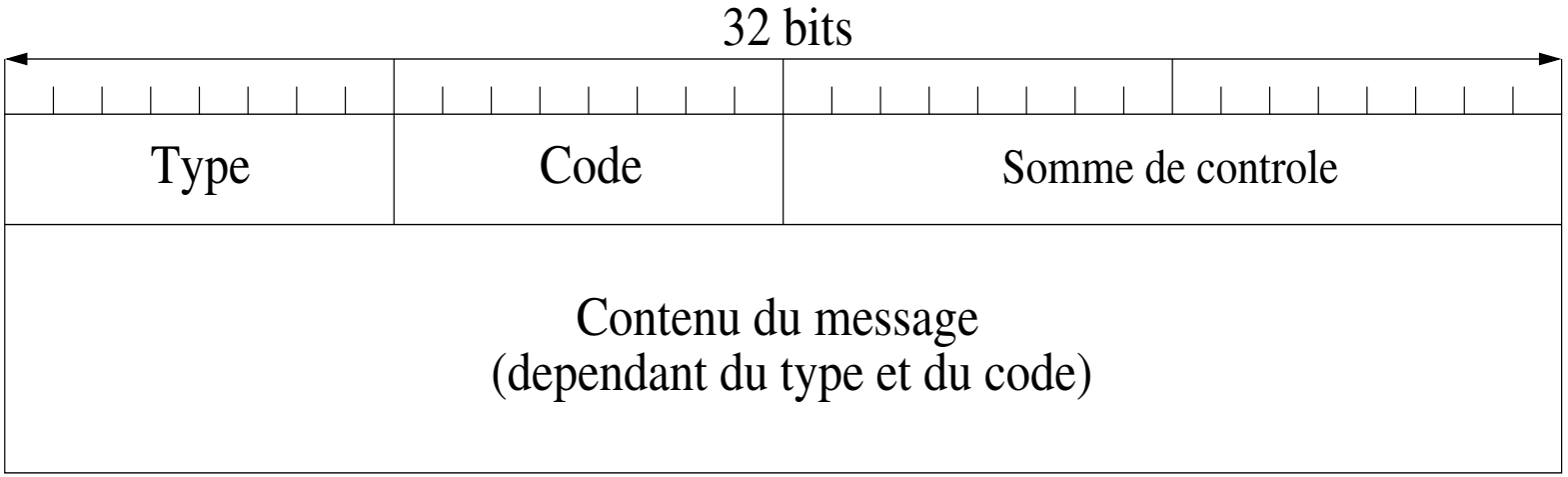


Internet Control Message Protocol (ICMP)

Services

- Messages de collecte d'informations sur l'état du réseau
- Messages de signalisation et de notification d'erreurs

Message ICMP



- type (1 oct) : spécifie le type de message ICMP.
- code (1 oct) : précise la nature de l'erreur signalée.
- somme de contrôle (2 oct) : calculé sur l'intégralité du message ICMP
- données : le contenu du message (dépend du type et du code de l'erreur).

Le message ICMP est encapsulé dans IP

Principaux messages ICMP : Demande d'information

Message	Type	Code	Description
Echo Request	8	0	envoi d'une requête Echo
Echo Reply	0	0	réponse à un Echo Request
Timestamp Request	13	0	demander le temps actuel
Timestamp Reply	14	0	réponse à un Timestamp Request

D'autres messages ICMP : Notification d'erreurs

- Destination inaccessible
- Réseau inaccessible
- Machine inaccessible
- Fragmentation requise mais non autorisée
- Expiration TTL...

Principaux messages ICMP : Demande d'information

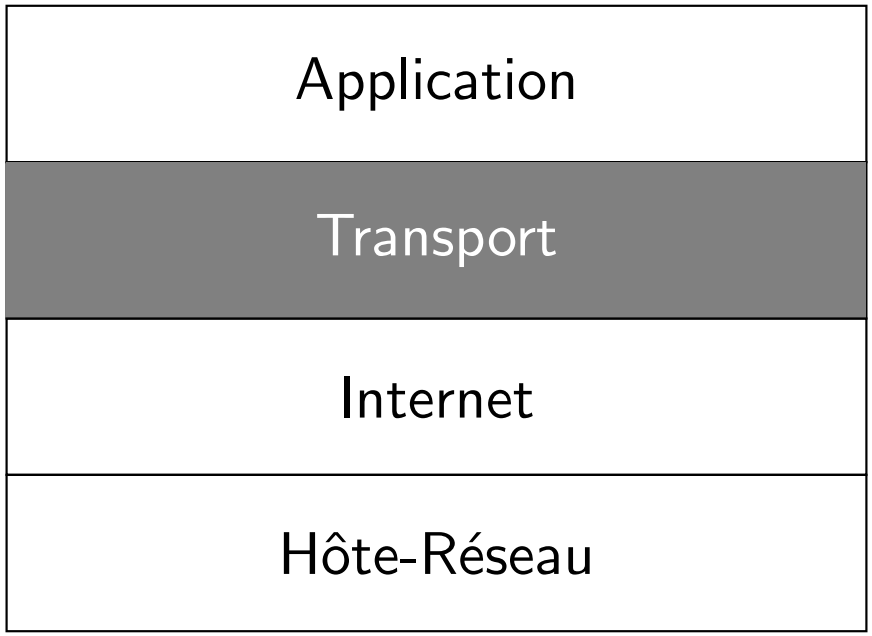
Message	Type	Code	Description
Echo Request	8	0	envoi d'une requête Echo
Echo Reply	0	0	réponse à un Echo Request
Timestamp Request	13	0	demander le temps actuel
Timestamp Reply	14	0	réponse à un Timestamp Request

D'autres messages ICMP : Notification d'erreurs

- Destination inaccessible
- Réseau inaccessible
- Machine inaccessible
- Fragmentation requise mais non autorisée
- Expiration TTL...

Les couches « de bout en bout »

Modèle TCP/IP



La couche transport

La couche transport :

- gère une conversation directe (sans intermédiaires) entre source (machine A) et destination (machine B)
- est présente seulement sur les machines, pas sur les routers qui les relient.
- les services principaux offerts au niveaux supérieurs concernent les différents types de transport des informations entre
  - une entité de niveau transport de la machine A et
  - son entité homologue de la machine B.
- Les services peuvent être :
  - fiables orientés à la connexion (typiques de ce niveau)
  - non-fiables et non connectés.

La couche transport

La couche transport :

- gère une conversation directe (sans intermédiaires) entre source (machine A) et destination (machine B)
- est présente seulement sur les machines, pas sur les routers qui les relient.
- les services principaux offerts au niveaux supérieurs concernent les différents types de transport des informations entre
  - une entité de niveau transport de la machine A et
  - son entité homologue de la machine B.
- Les services peuvent être :
  - fiables orientés à la connexion (typiques de ce niveau)
  - non-fiables et non connectés.

La couche transport

La couche transport :

- gère une conversation directe (sans intermédiaires) entre source (machine A) et destination (machine B)
- est présente seulement sur les machines, pas sur les routers qui les relient.
- les services principaux offerts au niveaux supérieurs concernent les différents types de transport des informations entre
  - une entité de niveau transport de la machine A et
  - son entité homologue de la machine B.
- Les services peuvent être :
  - fiables orientés à la connexion (typiques de ce niveau)
  - non-fiables et non connectés.

La couche transport

La couche transport :

- gère une conversation directe (sans intermédiaires) entre source (machine A) et destination (machine B)
- est présente seulement sur les machines, pas sur les routers qui les relient.
- les services principaux offerts au niveaux supérieurs concernent les différents types de transport des informations entre
  - une entité de niveau transport de la machine A et
  - son entité homologue de la machine B.
- Les services peuvent être :
  - fiables orientés à la connexion (typiques de ce niveau)
  - non-fiables et non connectés.

La couche transport

La couche transport comporte deux principaux protocoles :

- TCP
  - fiable,
  - avec connexion et
  - acquittements groupés,
- UDP
  - non-fiable,
  - sans connexion,
  - ni confirmation.

Transmission Control Protocol (TCP)

TCP : protocole de transmission fiable.

- Services :
  - Contrôler les données transmises (données endommagées, perdues, dupliquée)
  - Délivrer les données dans l'ordre de transmission même si la couche IP ne les délivre pas dans cet ordre.
  - Contrôle de flux : l'émetteur ne doit pas saturer le récepteur,
  - Contrôle de congestion : émetteur module son débit en fonction de la congestion du réseau
- Mode d'opération : connecté
  - Trois phases : connexion, échange, déconnexion.

Ne fournit pas de garanties de délai ou de débit.

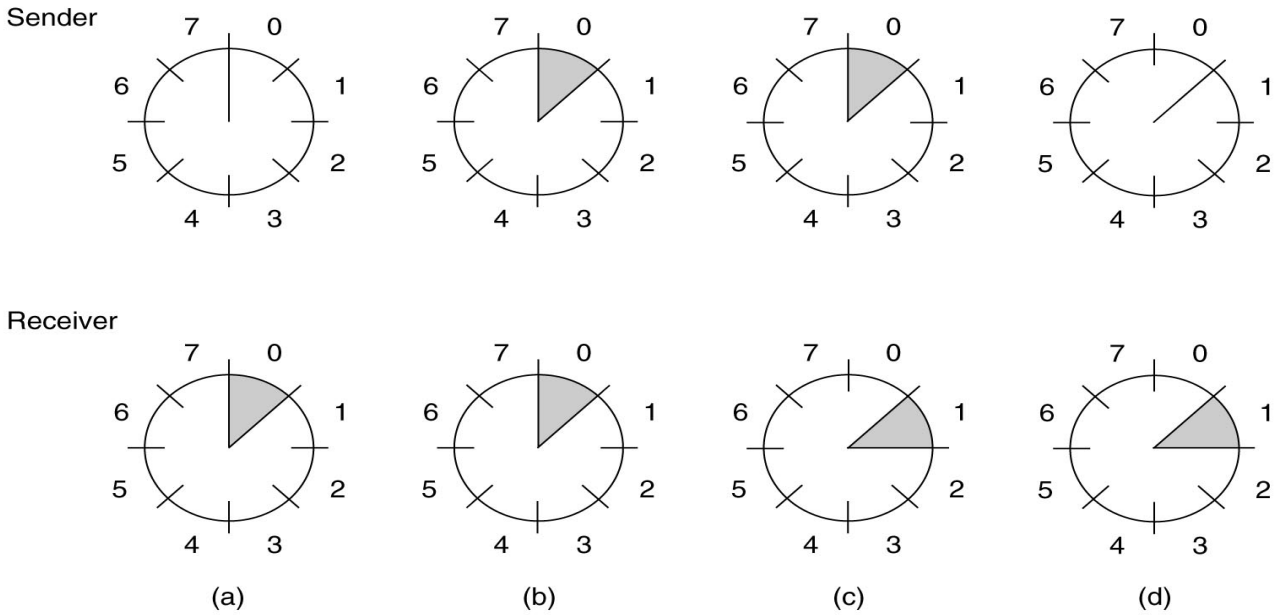
Connexion

Une connexion TCP/IP est caractérisée par 4 informations :

- adresse source ;
- adresse destination ;
- port source ;
- port destination ;

Le numéro de port représente le Service Access Point de niveau Application.

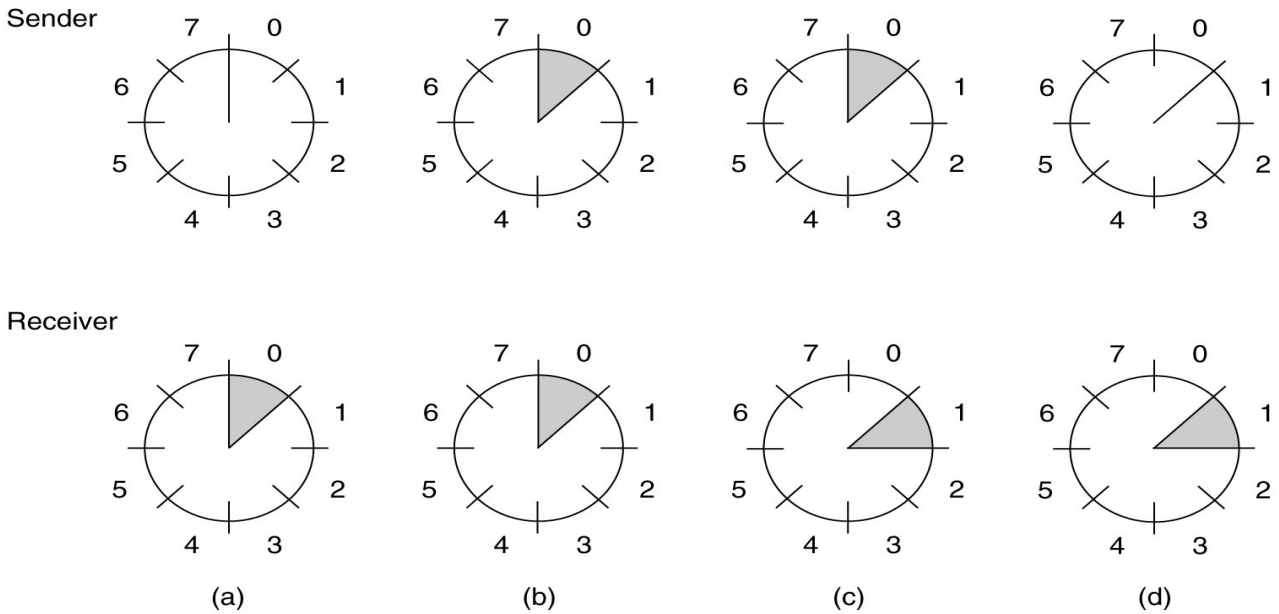
Sliding Window



Le protocole utilise une fenêtre à anticipation :

- Lorsqu'un émetteur envoie un segment (message) TCP
  - le segment M est sauvegardé dans une zone de mémoire (fenêtre de l'émetteur)
  - un temporisateur T_1 est armé.

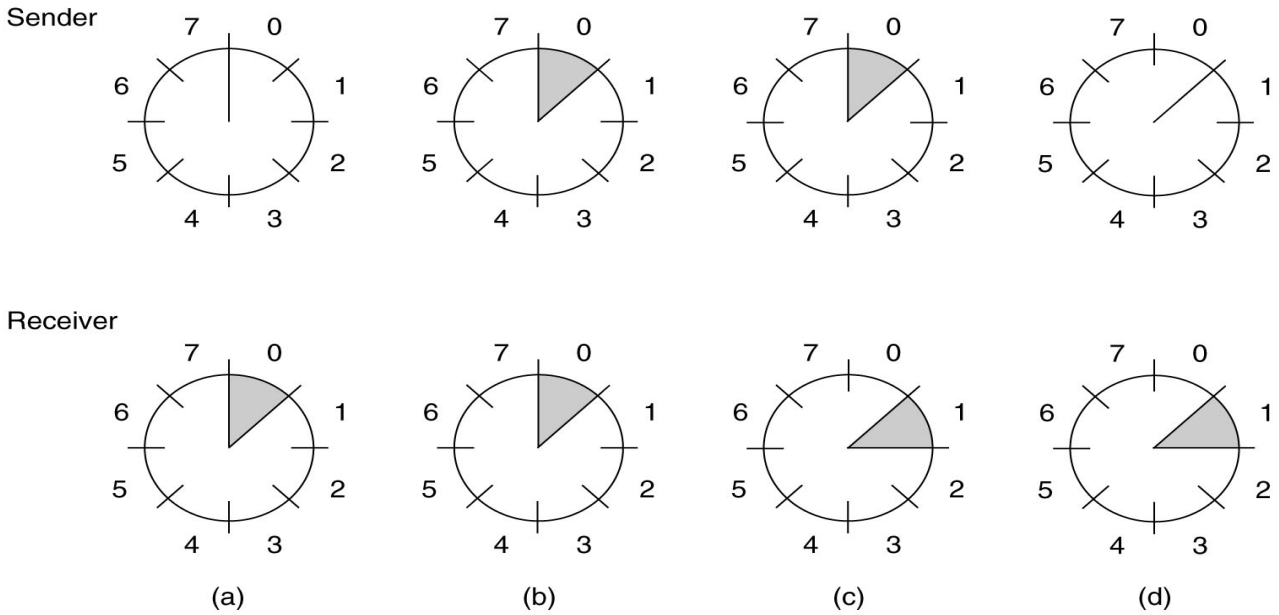
Sliding Window



Le protocole utilise une fenêtre à anticipation :

- Lorsque le destinataire reçoit le segment M
  - la fenêtre du récepteur glisse,
  - un acquittement est envoyé à l'émetteur.

Sliding Window



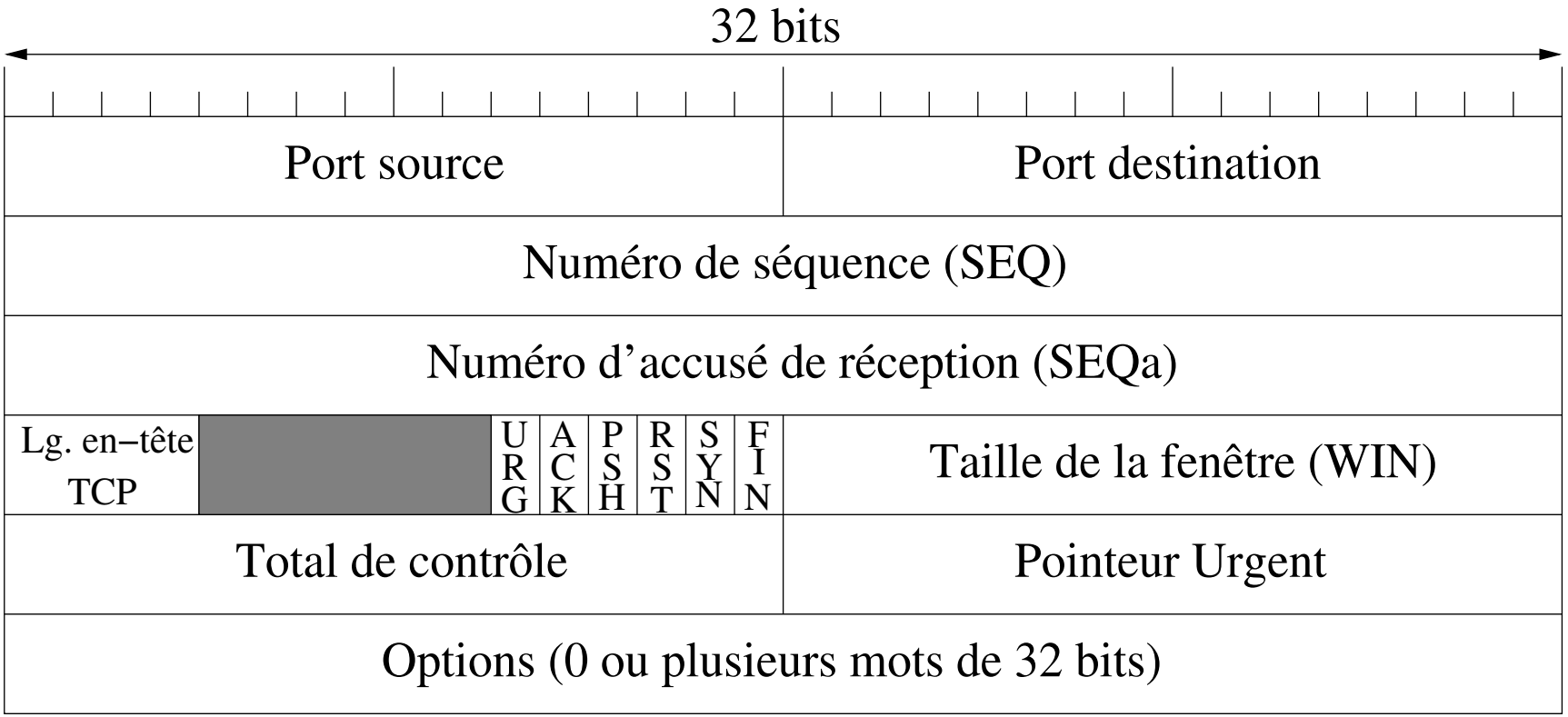
Le protocole utilise une fenêtre à anticipation :

- Lorsque l'émetteur reçoit l'acquittement de M
  - le segment sauvegardé est supprimé,
  - la fenêtre de l'émetteur glisse,
  - le temporisateur T_1 est desarmé.

Plus généralement. . .

- La fenêtre peut contenir plusieurs segments de taille différente,
- Le destinataire peut acquitter plusieurs segments avec un message d'acquittement.
- L'acquittement peut être transmis dans un segment contenant des données (piggybacking),
- Si la fenêtre de l'émetteur est pleine, il attend un acquittement avant d'envoyer d'autres segments.
- Si la fenêtre du récepteur est pleine, il envoie à l'émetteur un message demandant de ne pas envoyer d'autres segments.

Segment TCP



Le message TCP est encapsulé dans IP

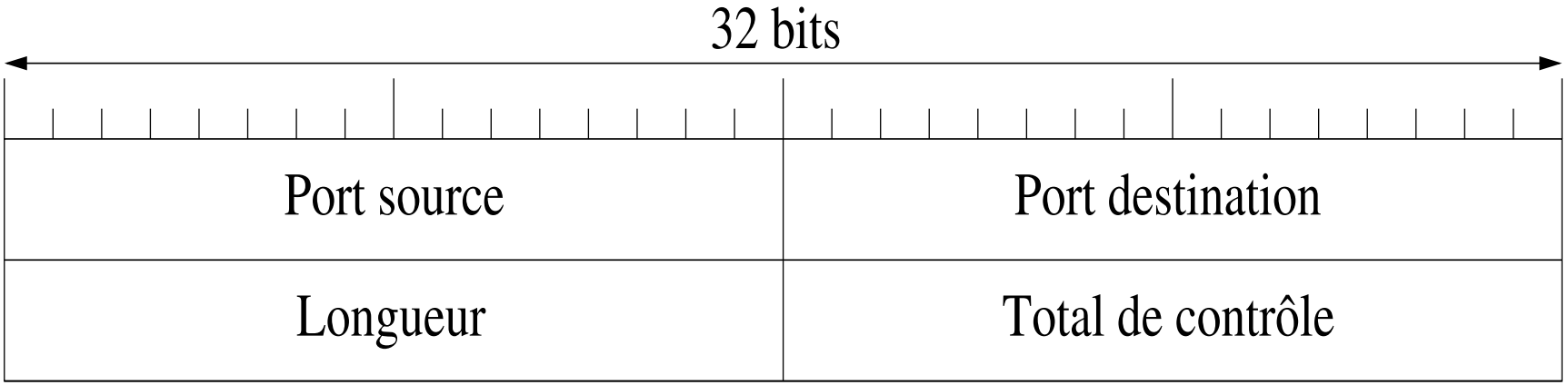
Protocole UDP (couche Transport)

UDP :

- transport en mode non connecté, sans confirmation.

Permet de gagner en débit pour les transmissions gourmandes, telles que vidéos et sons.

Datagramme UDP

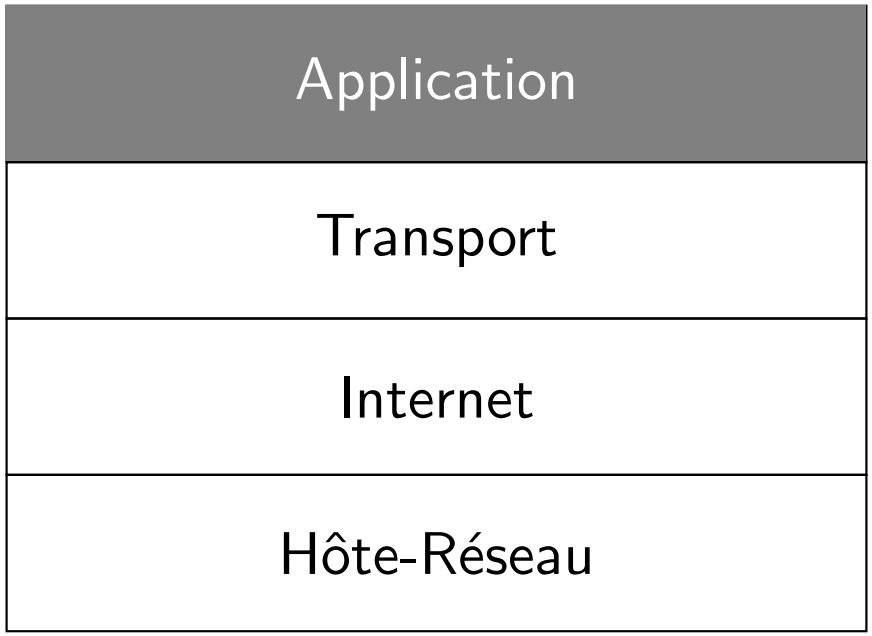


- Port source (2 octets),
- Port destination (2 octets),
- Longueur (2 octets) : indique la longueur totale du message (donnée et en-tête compris),
- Total de contrôle (2 octets) : FCS pour protéger le message UDP.

Le message UDP est encapsulé dans IP

Les couches « de bout en bout »

Modèle TCP/IP



La couche application

La couche application :

- DNS : associer des noms d'hôtes aux adresses de réseaux
- FTP : transfert de fichiers
- Telnet : ouverture de session distante
- SMTP : E-mail
- TFTP : protocole de transfert de fichiers
- HTTP : World Wide Web
- HTTPS : World Wide Web sécurisé
- POP3 : Accès e-mail à distance

Détection et correction d'erreurs

Détection et correction d'erreurs

Le service rendu par la *couche liaison de données* fait croire à la couche réseau qu'elle utilise une couche physique parfaite.

- En réalité, le support matériel utilisé par la couche physique n'est pas fiable à 100%.
- Certains bits reçus peuvent être erronés (un 0 à la place d'un 1 ou inversement).

Détection et correction d'erreurs

La couche liaison utilise principalement deux méthodes :

- la détection/correction d'erreurs ;
- un ≪ dialogue ≫ entre émetteur et récepteur, visant à s'assurer que la transmission se passe correctement : envoi d'accusé de réception, retransmission éventuelle de messages, etc.

Frame Check Sequence

Code détecteur/correcteur d'erreurs

Le FCS (Frame Check Sequence) est un groupe de bits que l'émetteur ajoute au message à transmettre pour introduire de la redondance.

Exemple : Trame Ethernet

Préambule	SFD	DA	SA	DL/EType	Données	Bourrage	FCS

Le FCS doit permettre de détecter des erreurs parmi les bits reçus, et éventuellement les localiser et les corriger.

Frame Check Sequence

Code détecteur/correcteur d'erreurs

Le FCS (Frame Check Sequence) est un groupe de bits que l'émetteur ajoute au message à transmettre pour introduire de la redondance.

Exemple : Trame Ethernet

Préambule	SFD	DA	SA	DL/EType	Données	Bourrage	FCS

Le FCS doit permettre de détecter des erreurs parmi les bits reçus, et éventuellement les localiser et les corriger.

S'il y a trop d'erreurs, la détection et la correction deviennent impossibles.

Rendement d'un code

Tous les codes ne sont pas équivalents.

Definition (Rendement d'un code)

Si le message comporte m bits et le FCS a une longueur de k bits, alors on dit que le rendement du code est

$$\frac{m}{m+k}$$

.

Idéalement on veut :

- minimiser le nombre k de bits à rajouter au message m,
- maximiser le nombre d'erreurs que l'on peut détecter et corriger.

Rendement d'un code

Tous les codes ne sont pas équivalents.

Definition (Rendement d'un code)

Si le message comporte m bits et le FCS a une longueur de k bits, alors on dit que le rendement du code est

$$\frac{m}{m+k}$$

.

Idéalement on veut :

- minimiser le nombre k de bits à rajouter au message m,
- maximiser le nombre d'erreurs que l'on peut détecter et corriger.

Un code simple : la répétition

Une approche naïve consiste à répéter k fois le message à transmettre.

Si l'on envoie :

- 2 exemplaires du message :
  on cherche les différences entre les première et seconde moitiés du message.
  Impossible de corriger une erreur détectée !
  Rendement $= 0.5$.
- 3 exemplaires du message :
  Le bit correct est probablement celui qui apparaît en deux exemplaires.
  On peut cette fois corriger l'erreur.
  Rendement $= 0.33$.

Un code simple : la répétition

Une approche naïve consiste à répéter k fois le message à transmettre.

Si l'on envoie :

- 2 exemplaires du message :
  on cherche les différences entre les première et seconde moitiés du message.
  Impossible de corriger une erreur détectée !
  Rendement $= 0.5$.
- 3 exemplaires du message :
  Le bit correct est probablement celui qui apparaît en deux exemplaires.
  On peut cette fois corriger l'erreur.
  Rendement $= 0.33$.

Un code simple : la répétition

Une approche naïve consiste à répéter k fois le message à transmettre.

Si l'on envoie :

- 2 exemplaires du message :
  on cherche les différences entre les première et seconde moitiés du message.
  Impossible de corriger une erreur détectée !
  Rendement $= 0.5$.
- 3 exemplaires du message :
  Le bit correct est probablement celui qui apparaît en deux exemplaires.
  On peut cette fois corriger l'erreur.
  Rendement $= 0.33$.

Inconvénients de la répétition

Le code de répétition est simple, mais présente de nombreux inconvénients :

- le rendement est faible (respectivement 0.5 et 0.33) ;
- certaines erreurs peuvent ne pas être détectées ;
- certaines erreurs détectées ne peuvent pas être corrigées, voire être mal corrigées ;
- la correction nécessite plus de redondance que la détection d'erreurs.

Les differents types de codes

Deux types de codes :

- codes de protection des caractères ASCII :
  - VRC,
  - LRC,
  - VRC+LRC.
- codes à blocs :
  - basés sur la distance de Hamming,
  - basés sur les polynômes.

VRC (Vertical Redundancy Check)

Utilisé pour protéger les codes ASCII.

VRC

Un code ASCII est défini sur 7 bits, le $8^{\text{ème}}$ est utilisé comme bit de parité.

$$1 \text{ octet} = \underbrace{\mathbf{b_7}}_{\text{bit de parité}} \underbrace{b_6 b_5 b_4 b_3 b_2 b_1 b_0}_{ASCII}$$

Envoi de la chaine de caractères "IUT"

$$\underbrace{1001001}_{I} \underbrace{1010101}_{U} \underbrace{1010100}_{T}$$

VRC (Vertical Redundancy Check)

Utilisé pour protéger les codes ASCII.

VRC

Un code ASCII est défini sur 7 bits, le $8^{\text{ème}}$ est utilisé comme bit de parité.

$$1 \text{ octet} = \underbrace{\mathbf{b_7}}_{\text{bit de parité}} \underbrace{b_6 b_5 b_4 b_3 b_2 b_1 b_0}_{ASCII}$$

Envoi de la chaine de caractères "IUT"

$$\underbrace{1001001}_{I} \underbrace{1010101}_{U} \underbrace{1010100}_{T}$$

VRC Pair

VRC Pair

Le bit de parité est ajouté de sorte que la séquence finale

$$\mathbf{b_7}\, b_6\, b_5\, b_4\, b_3\, b_2\, b_1\, b_0$$

contienne un nombre pair de bits à 1.

Envoi de la chaine de caractères "IUT" avec un VRC pair

$$\underbrace{\mathbf{1}1001001}_{I}\ \underbrace{\mathbf{0}1010101}_{U}\ \underbrace{\mathbf{1}1010100}_{T}$$

VRC Pair

VRC Pair

Le bit de parité est ajouté de sorte que la séquence finale

$$\mathbf{b_7}\, b_6\, b_5\, b_4\, b_3\, b_2\, b_1\, b_0$$

contienne un nombre pair de bits à 1.

Envoi de la chaine de caractères "IUT" avec un VRC pair

$$\underbrace{\mathbf{1}1001001}_{I}\ \underbrace{\mathbf{0}1010101}_{U}\ \underbrace{\mathbf{1}1010100}_{T}$$

LRC (Longitudinal Redundancy Check)

LRC

Au lieu de protéger les caractères un par un, on protège l'ensemble des bits de même rang de tous les caractères :

$$\begin{array}{c} b_6^0\, b_5^0\, b_4^0\, b_3^0\, b_2^0\, b_1^0\, b_0^0 \\ b_6^1\, b_5^1\, b_4^1\, b_3^1\, b_2^1\, b_1^1\, b_0^1 \\ b_6^2\, b_5^2\, b_4^2\, b_3^2\, b_2^2\, b_1^2\, b_0^2 \\ \vdots \\ b_6^n\, b_5^n\, b_4^n\, b_3^n\, b_2^n\, b_1^n\, b_0^n \\ \hline \mathbf{b_6 b_5 b_4 b_3 b_2 b_1 b_0} \end{array}$$

Séquence envoyée :

$$b_6^0\, b_5^0\, b_4^0\, b_3^0\, b_2^0\, b_1^0\, b_0^0 \cdots b_6^n\, b_5^n\, b_4^n\, b_3^n\, b_2^n\, b_1^n\, b_0^n\ \mathbf{b_6 b_5 b_4 b_3 b_2 b_1 b_0}$$

On obtient alors un code de protection sur 7 bits.

LRC Pair

LRC Pair

Le bit de parité est ajouté de sorte que la séquence finale de chaque colonne

$$b_i^0$$
$$b_i^1$$
$$\vdots$$
$$b_i^n$$
$$\mathbf{b_i}$$

contienne un nombre pair de bits à 1.

Envoi de la chaine de caractères “IUT” avec un VRC pair

I	1001001
U	1010101
T	1010100
LRC pair	**1001000**

LRC Pair

LRC Pair

Le bit de parité est ajouté de sorte que la séquence finale de chaque colonne

$$b_i^0$$
$$b_i^1$$
$$\vdots$$
$$b_i^n$$
$$\mathbf{b_i}$$

contienne un nombre pair de bits à 1.

Envoi de la chaine de caractères “IUT” avec un VRC pair

I	1001001
U	1010101
T	1010100
LRC pair	**1001000**

VRC + LRC

On peut également combiner les deux techniques précédentes.

Envoi de la chaine de caractères “IUT” avec un VRC+LRC pair

On code chaque lettre en VRC puis en LRC :

		VRC pair
I	1001001	**1**1001001
U	1010101	**0**1010101
T	1010100	**1**1010100
LRC		**01001000**

$$\underbrace{\mathbf{1}1001001}_{I}\ \underbrace{\mathbf{0}1010101}_{U}\ \underbrace{\mathbf{1}1010100}_{T}\ \underbrace{\mathbf{01001000}}_{LRC}$$

VRC, LRC, VRC + LRC — Version Impaire

Même chose : mais on s'assure que le nombre de "1" dans la séquence finale soit impair.

Le code de Hamming

Distance de Hamming

La distance de Hamming $d(s_1, s_2)$ entre deux séquences binaires s_1 et s_2 de même taille est le nombre de bits de même rang par lesquels ces deux séquences diffèrent.

$$d(1100110, 1010111) = 3$$

Le code de Hamming

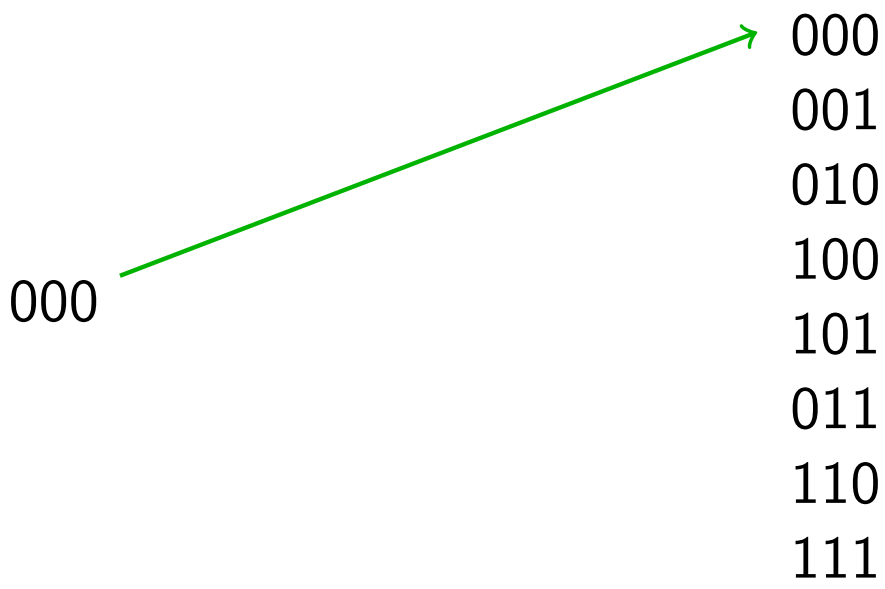
Distance de Hamming

La distance de Hamming $d(s_1, s_2)$ entre deux séquences binaires s_1 et s_2 de même taille est le nombre de bits de même rang par lesquels ces deux séquences diffèrent.

$$d(1100110, 1010111) = \mathbf{3}$$

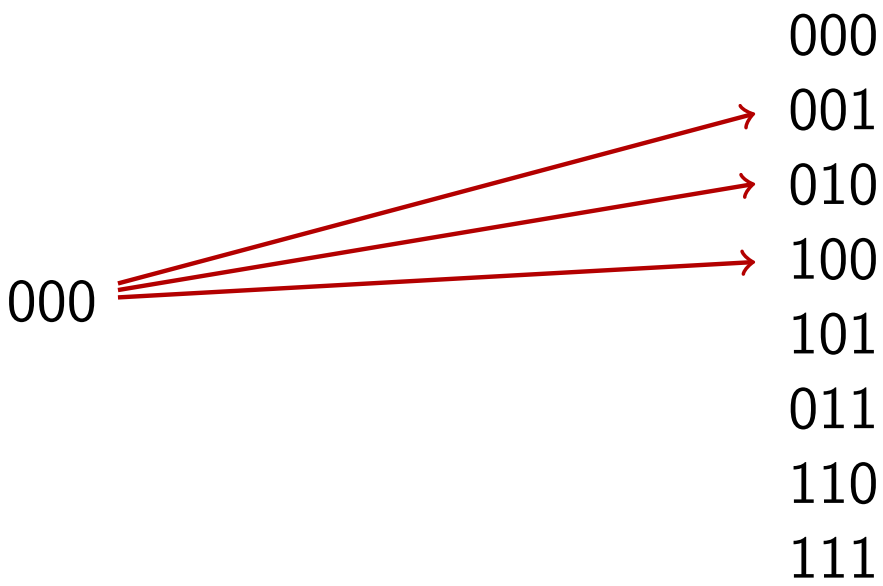
Principe du code de Hamming

Transmission correcte :



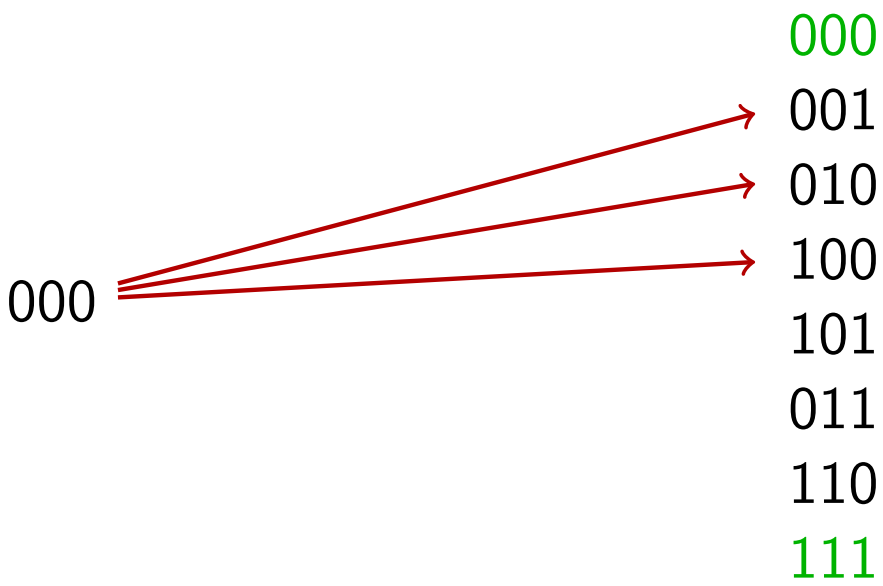
Principe du code de Hamming

1 erreur : distance de Hamming 1.



Principe du code de Hamming

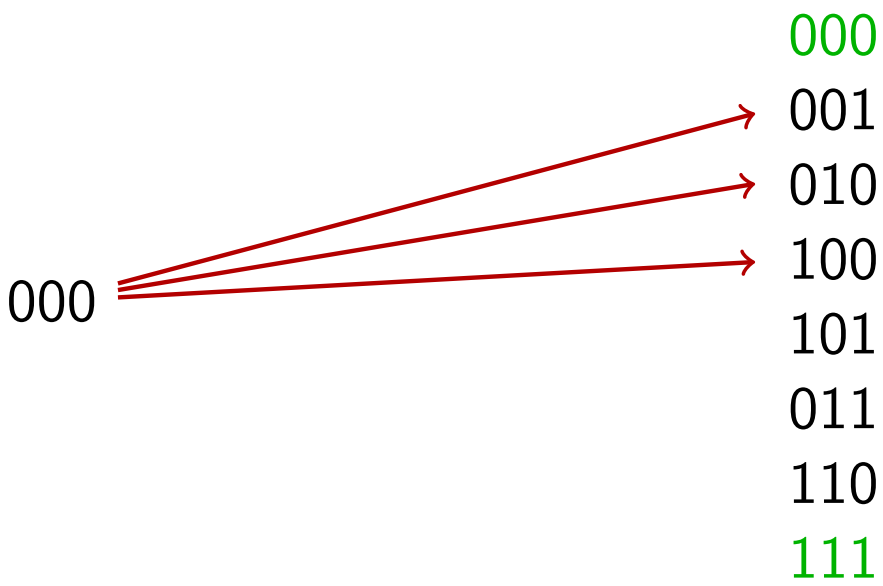
1 erreur : distance de Hamming 1.



Supposons que les mots valides du code soient dans $\mathcal{C} = \{000, 111\}$

Principe du code de Hamming

1 erreur : distance de Hamming 1.

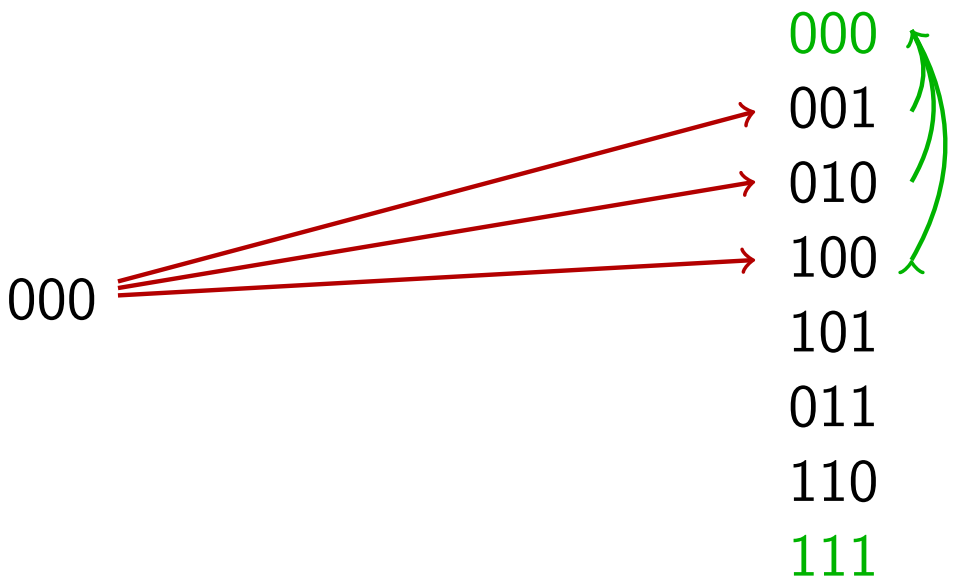


Supposons que les mots valides du code soient dans $\mathcal{C} = \{000, 111\}$

- on peut détecter qu'il y a eu 1 erreur

Principe du code de Hamming

1 erreur : distance de Hamming 1.

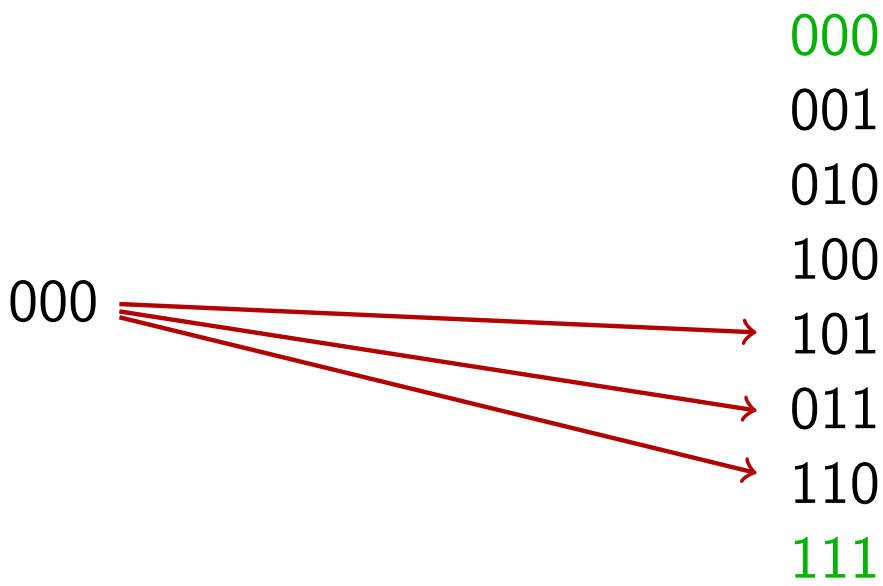


Supposons que les mots valides du code soient dans $\mathcal{C} = \{000, 111\}$

- on peut détecter qu'il y a eu 1 erreur
- on peut corriger en cherchant le mot valide le plus proche

Attention

2 erreurs : distance de Hamming 2.

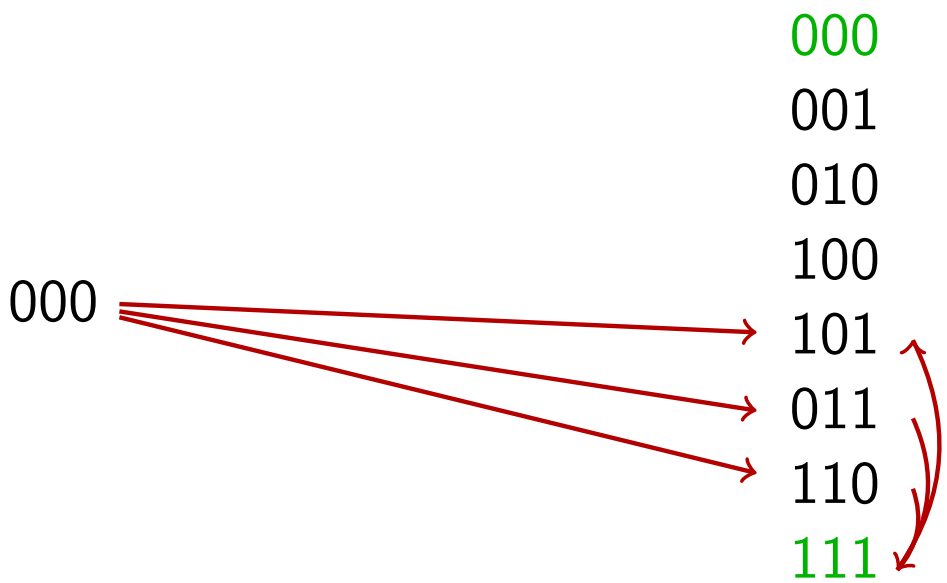


Mots valides du code $\mathcal{C} = \{000, 111\}$

- on peut détecter qu'il y a eu une erreur

Attention

2 erreurs : distance de Hamming 2.

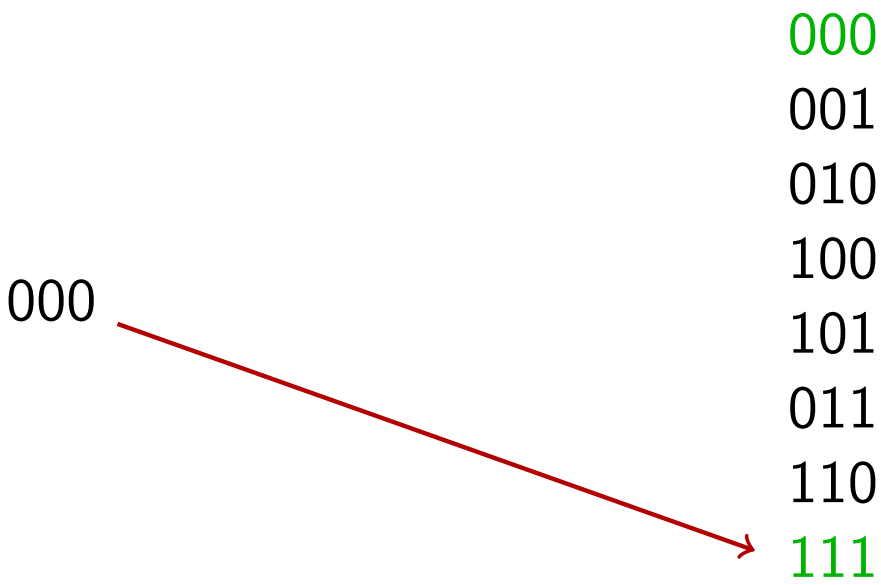


Mots valides du code $\mathcal{C} = \{000, 111\}$

- on peut détecter qu'il y a eu une erreur
- lorsqu'on corrige on trouve le mauvais mot

Attention

3 erreurs : distance de Hamming 3.



Mots valides du code $\mathcal{C} = \{000, 111\}$

- on ne peut pas détecter qu'il y a eu 3 erreurs, donc pas corriger. C'est parce que $d(000, 111) = 3$.

Le code de Hamming (7,4)

message	bits de contrôle	mot du code
0000	000	0000 000
0001	101	0001 101
0010	111	0010 111
0011	010	0011 010
0100	011	0100 011
0101	110	0101 110
0110	100	0110 100
0111	001	0111 001
1000	110	1000 110
1001	011	1001 110
1010	001	1010 001
1011	100	1011 100
1100	101	1100 101
1101	000	1101 000
1110	010	1110 010
1111	111	1111 111

Procédure de détection et correction d'erreurs

Une machine reçoit le message M :

1. Si $M \in \mathcal{C}$ alors le message M est correct.
2. Si $M \notin \mathcal{C}$ alors une erreur s'est produite :
   1. On calcule pour chaque $w \in \mathcal{C}$ la distance $d(w, M)$.
   2. S'il existe un mot w tel que $d(w, M) < d(w', M)$ pour tout $w' \in \mathcal{C}$ alors on considère que le message correct était w
   3. Sinon, on a détecté l'erreur mais on ne peut pas le corriger.

Exemple

On utilise le code de Hamming (7,4), on reçoit le message 1110011 qui n'est pas un mot valide du code.

Procédure de détection et correction d'erreurs

Une machine reçoit le message M :

1. Si $M \in \mathcal{C}$ alors le message M est correct.
2. Si $M \notin \mathcal{C}$ alors une erreur s'est produite :
   1. On calcule pour chaque $w \in \mathcal{C}$ la distance $d(w, M)$.
   2. S'il existe un mot w tel que $d(w, M) < d(w', M)$ pour tout $w' \in \mathcal{C}$ alors on considère que le message correct était w
   3. Sinon, on a détecté l'erreur mais on ne peut pas le corriger.

Exemple

On utilise le code de Hamming (7,4), on reçoit le message 1110011 qui n'est pas un mot valide du code.

Procédure de détection et correction d'erreurs

Une machine reçoit le message M :

1. Si $M \in \mathcal{C}$ alors le message M est correct.
2. Si $M \notin \mathcal{C}$ alors une erreur s'est produite :
   1. On calcule pour chaque $w \in \mathcal{C}$ la distance $d(w, M)$.
   2. S'il existe un mot w tel que $d(w, M) < d(w', M)$ pour tout $w' \in \mathcal{C}$ alors on considère que le message correct était w'
   3. Sinon, on a détecté l'erreur mais on ne peut pas le corriger.

Exemple

On utilise le code de Hamming (7,4), on reçoit le message 1110011 qui n'est pas un mot valide du code.

Procédure de détection et correction d'erreurs

Une machine reçoit le message M :

1. Si $M \in \mathcal{C}$ alors le message M est correct.
2. Si $M \notin \mathcal{C}$ alors une erreur s'est produite :
   1. On calcule pour chaque $w \in \mathcal{C}$ la distance $d(w, M)$.
   2. S'il existe un mot w tel que $d(w, M) < d(w', M)$ pour tout $w' \in \mathcal{C}$ alors on considère que le message correct était w'
   3. Sinon, on a détecté l'erreur mais on ne peut pas le corriger.

Exemple

On utilise le code de Hamming (7,4), on reçoit le message 1110011 qui n'est pas un mot valide du code.

Procédure de détection et correction d'erreurs

Une machine reçoit le message M :

1. Si $M \in \mathcal{C}$ alors le message M est correct.
2. Si $M \notin \mathcal{C}$ alors une erreur s'est produite :
   1. On calcule pour chaque $w \in \mathcal{C}$ la distance $d(w, M)$.
   2. S'il existe un mot w tel que $d(w, M) < d(w', M)$ pour tout $w' \in \mathcal{C}$ alors on considère que le message correct était w'
   3. Sinon, on a détecté l'erreur mais on ne peut pas le corriger.

Exemple

On utilise le code de Hamming (7,4), on reçoit le message 1110011 qui n'est pas un mot valide du code.

Procédure de détection et correction d'erreurs

Une machine reçoit le message M :

1. Si $M \in \mathcal{C}$ alors le message M est correct.
2. Si $M \notin \mathcal{C}$ alors une erreur s'est produite :
   1. On calcule pour chaque $w \in \mathcal{C}$ la distance $d(w, M)$.
   2. S'il existe un mot w tel que $d(w, M) < d(w', M)$ pour tout $w' \in \mathcal{C}$ alors on considère que le message correct était w'
   3. Sinon, on a détecté l'erreur mais on ne peut pas le corriger.

Exemple

On utilise le code de Hamming (7,4), on reçoit le message 1110011 qui n'est pas un mot valide du code.

Le code de Hamming

mot du code	distance avec 1110011
0000 000	5
0001 101	6
0010 111	3
0011 010	2
0100 011	2
0101 110	5
0110 100	4
0111 001	3
1000 110	4
1001 110	3
1010 001	2
1011 100	5
1100 101	3
1101 000	3
1110 010	1
1111 111	2

Le code de Hamming

mot du code	distance avec 1110011
0000 000	5
0001 101	6
0010 111	3
0011 010	2
0100 011	2
0101 110	5
0110 100	4
0111 001	3
1000 110	4
1001 110	3
1010 001	2
1011 100	5
1100 101	3
1101 000	3
1110 010	1
1111 111	2

Distance de Hamming d'un Code

Distance dj code C : $d(C)$

Soit un code C comportant n séquences valides. La distance de Hamming $d(C)$ du code C est la distance minimale séparant deux mots valides du code.

$\mathcal{C} = \{0000, 0011, 1100, 1111\}$.

0000	X			
0011	2	X		
1100	2	4	X	
1111	4	2	2	X
	0000	0011	1100	1111

$d(\mathcal{C}) = 2$

Distance de Hamming d'un Code

Distance dj code C : $d(C)$

Soit un code C comportant n séquences valides. La distance de Hamming $d(C)$ du code C est la distance minimale séparant deux mots valides du code.

$\mathcal{C} = \{0000, 0011, 1100, 1111\}$.

0000	X			
0011	2	X		
1100	2	4	X	
1111	4	2	2	X
	0000	0011	1100	1111

$d(\mathcal{C}) = 2$

Distance de Hamming d'un Code

Distance dj code C : $d(C)$

Soit un code C comportant n séquences valides. La distance de Hamming $d(C)$ du code C est la distance minimale séparant deux mots valides du code.

$\mathcal{C} = \{0000, 0011, 1100, 1111\}$.

0000	X			
0011	2	X		
1100	2	4	X	
1111	4	2	2	X
	0000	0011	1100	1111

$d(\mathcal{C}) = 2$

Distance de Hamming d'un code

Un code de distance $d(C)$ est capable de :

- détecter *au moins* $d(C) - 1$ erreurs
- corriger *au moins* $k = \lfloor \frac{d(C)-1}{2} \rfloor$ erreurs.

Codes polynomiaux

Les codes polynomiaux se basent sur l'utilisation de polynômes à coefficients dans $\mathbb{Z}/2\mathbb{Z}$, donc :

- tous les coefficients sont 0 ou 1,
- $1 \oplus 1 = 0$
- $x \oplus x = 0$
- $x = -x$

Codes polynomiaux

Un polynôme de degré k s'écrit sous la forme :

$$G(x) = a_0 \oplus a_1.x \oplus a_2.x^2 \oplus a_3.x^3 \oplus \ldots \oplus a_k.x^k \text{ où } a_i \in \{0, 1\}$$

Tout polynôme correspond à une séquence binaire de longueur $k + 1$:

$$a_k a_{k-1} \cdots a_1 a_0$$

et *vice versa*.

$P(x) = x^7 \oplus x^5 \oplus x \oplus 1$, degré 7

$$P = 10100011$$

M = 100101

$$M(x) = x^5 \oplus x^2 \oplus 1, \text{ degré } 5$$

Codes polynomiaux

Un polynôme de degré k s'écrit sous la forme :

$$G(x) = a_0 \oplus a_1.x \oplus a_2.x^2 \oplus a_3.x^3 \oplus \ldots \oplus a_k.x^k \text{ où } a_i \in \{0, 1\}$$

Tout polynôme correspond à une séquence binaire de longueur $k + 1$:

$$a_k a_{k-1} \cdots a_1 a_0$$

et *vice versa*.

$P(x) = x^7 \oplus x^5 \oplus x \oplus 1$, degré 7

$$P = 10100011$$

M = 100101

$$M(x) = x^5 \oplus x^2 \oplus 1, \text{ degré } 5$$

Codes polynomiaux

Un polynôme de degré k s'écrit sous la forme :

$$G(x) = a_0 \oplus a_1.x \oplus a_2.x^2 \oplus a_3.x^3 \oplus \ldots \oplus a_k.x^k \text{ où } a_i \in \{0, 1\}$$

Tout polynôme correspond à une séquence binaire de longueur $k + 1$:

$$a_k a_{k-1} \cdots a_1 a_0$$

et *vice versa*.

$P(x) = x^7 \oplus x^5 \oplus x \oplus 1$, degré 7

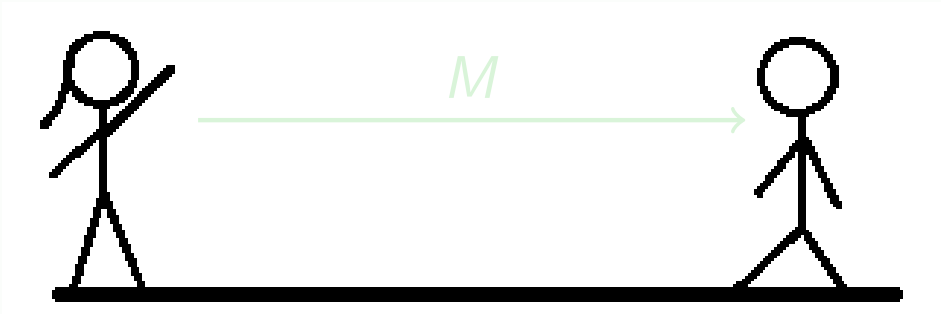
$$P = 10100011$$

M = 100101

$$M(x) = x^5 \oplus x^2 \oplus 1, \text{ degré } 5$$

Codes polynomiaux

Un code polynômial utilise un polynôme générateur $G(x)$.

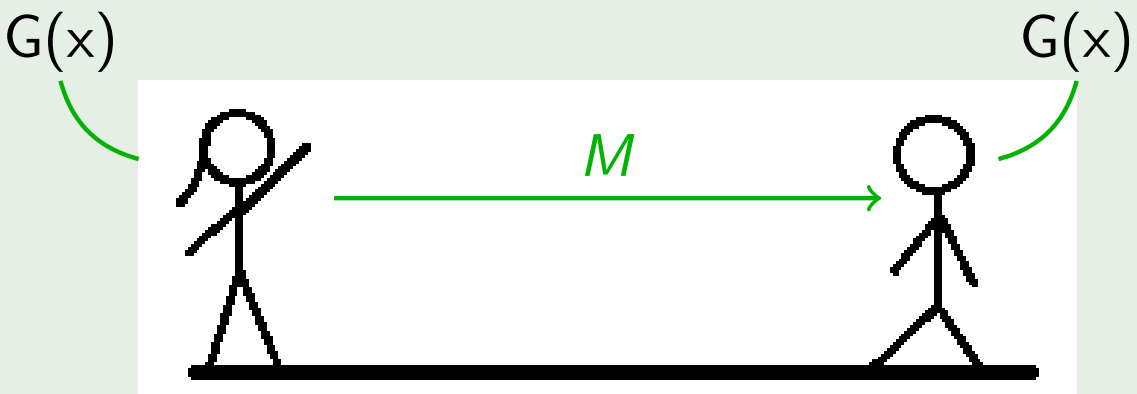


Alice envoie à Bob un message M

Codes polynomiaux

Un code polynômial utilise un polynôme générateur $G(x)$.

Alice et Bob connaissent $G(x)$

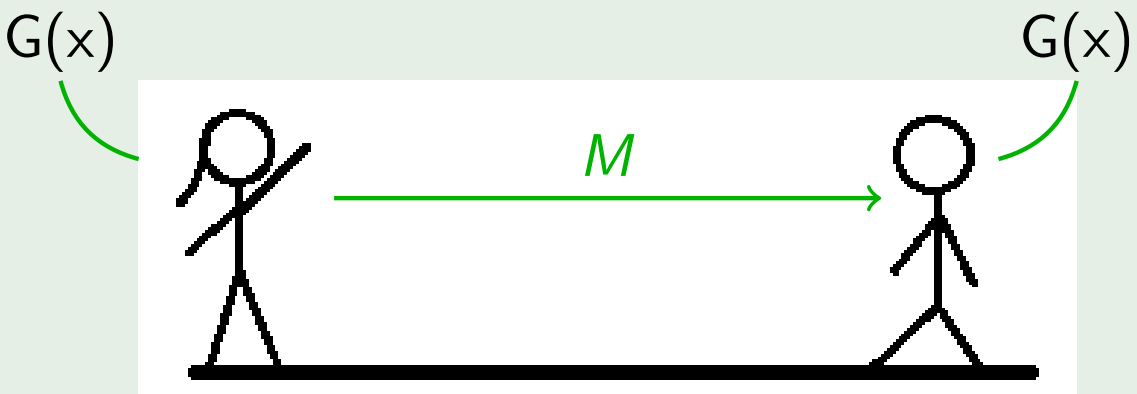


Alice envoie à Bob un message M

Codes polynomiaux

Un code polynômial utilise un polynôme générateur $G(x)$.

Alice et Bob connaissent $G(x)$



M(x)	G(x)
R(x)=0	Q(x)

Alice envoie à Bob un message M qui a la propriété suivante :
M(x) est divisible par G(x).

Division de polynômes dans $\mathbb{Z}/2\mathbb{Z}$

Pour calculer $\frac{x^3 \oplus x \oplus 1}{x \oplus 1}$ il faut se rappeler que $1 \oplus 1 = 0$

$$
\begin{array}{rrrrr|l}
 & x^3 & & \oplus x & \oplus 1 & x \oplus 1 \\
\cline{1-5}
\oplus & x^3 & \oplus x^2 & & & \\
\cline{1-3}
 & & \oplus x^2 & \oplus x & \oplus 1 & \\
 & & \oplus x^2 & \oplus x & & \\
\cline{3-4}
 & & & & 1 & \\
\end{array}
$$

$$
\underbrace{x^3 \oplus x \oplus 1}_{P(x)} = \underbrace{(x \oplus 1)}_{G(x)} \underbrace{(x^2 \oplus x)}_{Q(x)} \oplus \underbrace{1}_{R(x)}
$$

Division de polynômes dans $\mathbb{Z}/2\mathbb{Z}$

Pour calculer $\frac{x^3 \oplus x \oplus 1}{x \oplus 1}$ il faut se rappeler que $1 \oplus 1 = 0$

$$
\begin{array}{rrrrr|l}
 & x^3 & & \oplus x & \oplus 1 & x \oplus 1 \\
\cline{6-6}
\oplus & x^3 & \oplus x^2 & & & x^2 \\
\cline{1-4}
 & & \oplus x^2 & \oplus x & \oplus 1 & \\
 & & \oplus x^2 & \oplus x & & \\
\cline{3-4}
 & & & & 1 &
\end{array}
$$

$$
\underbrace{x^3 \oplus x \oplus 1}_{P(x)} = \underbrace{(x \oplus 1)}_{G(x)} \underbrace{(x^2 \oplus x)}_{Q(x)} \oplus \underbrace{1}_{R(x)}
$$

Division de polynômes dans $\mathbb{Z}/2\mathbb{Z}$

Pour calculer $\frac{x^3 \oplus x \oplus 1}{x \oplus 1}$ il faut se rappeler que $1 \oplus 1 = 0$

$$
\begin{array}{rrrrr|l}
 & x^3 & & \oplus x & \oplus 1 & x \oplus 1 \\
\cline{6-6}
\oplus & x^3 & \oplus x^2 & & & x^2 \oplus x \\
\cline{1-4}
 & & \oplus x^2 & \oplus x & \oplus 1 & \\
 & & \oplus x^2 & \oplus x & & \\
\cline{3-5}
 & & & & 1 &
\end{array}
$$

$$
\underbrace{x^3 \oplus x \oplus 1}_{P(x)} = \underbrace{(x \oplus 1)}_{G(x)} \underbrace{(x^2 \oplus x)}_{Q(x)} \oplus \underbrace{1}_{R(x)}
$$

Division de polynômes dans $\mathbb{Z}/2\mathbb{Z}$

Pour calculer $\frac{x^3 \oplus x \oplus 1}{x \oplus 1}$ il faut se rappeler que $1 \oplus 1 = 0$

$$
\begin{array}{rrrrr|l}
 & x^3 & & \oplus x & \oplus 1 & x \oplus 1 \\
\cline{6-6}
\oplus & x^3 & \oplus x^2 & & & x^2 \oplus x \\
\cline{1-3}
 & & \oplus x^2 & \oplus x & \oplus 1 & \\
 & & \oplus x^2 & \oplus x & & \\
\cline{2-4}
 & & & & 1 &
\end{array}
$$

$$\underbrace{x^3 \oplus x \oplus 1}_{P(x)} = \underbrace{(x \oplus 1)}_{G(x)}\underbrace{(x^2 \oplus x)}_{Q(x)} \oplus \underbrace{1}_{R(x)}$$

Division de polynômes dans $\mathbb{Z}/2\mathbb{Z}$

Pour calculer $\frac{x^3 \oplus x \oplus 1}{x \oplus 1}$ il faut se rappeler que $1 \oplus 1 = 0$

$$
\begin{array}{rrrrr|l}
 & x^3 & & \oplus x & \oplus 1 & x \oplus 1 \\
\cline{6-6}
\oplus & x^3 & \oplus x^2 & & & x^2 \oplus x \\
\cline{1-3}
 & & \oplus x^2 & \oplus x & \oplus 1 & \\
 & & \oplus x^2 & \oplus x & & \\
\cline{2-4}
 & & & & 1 &
\end{array}
$$

$$
\underbrace{x^3 \oplus x \oplus 1}_{P(x)} = \underbrace{(x \oplus 1)}_{G(x)} \underbrace{(x^2 \oplus x)}_{Q(x)} \oplus \underbrace{1}_{R(x)}
$$

Codage

La procédure de codage consiste à :

- calculer $P(x) = M(x).x^k$.
  - ceci correspond à un décalage de k bits (vers la gauche) du message M.
- diviser le polynôme $P(x)$ par $G(x)$. Soient $Q(x)$ et $R(x)$ les polynômes quotient et reste ainsi obtenus :

$$P(x) = Q(x).G(x) \oplus R(x)$$

- le CRC (Cyclic Redundancy Check) est le reste $R(x)$ ainsi calculé. On remarque que le reste est forcément au maximum de degré $k - 1$.
- le message effectivement transmis est associé au polynôme $M'(x) = P(x) \oplus R(x)$.

Le message envoyé est donné par le message initial M suivi de la séquence de k bits correspondant à $R(x)$.

Codage

La procédure de codage consiste à :

- calculer $P(x) = M(x).x^k$.
  - ceci correspond à un décalage de k bits (vers la gauche) du message M.
- diviser le polynôme $P(x)$ par $G(x)$. Soient $Q(x)$ et $R(x)$ les polynômes quotient et reste ainsi obtenus :

$$P(x) = Q(x).G(x) \oplus R(x)$$

- le CRC (Cyclic Redundancy Check) est le reste $R(x)$ ainsi calculé. On remarque que le reste est forcément au maximum de degré $k - 1$.
- le message effectivement transmis est associé au polynôme $M'(x) = P(x) \oplus R(x)$.

Le message envoyé est donné par le message initial M suivi de la séquence de k bits correspondant à $R(x)$.

Codage

La procédure de codage consiste à :

- calculer $P(x) = M(x).x^k$.
  - ceci correspond à un décalage de k bits (vers la gauche) du message M.
- diviser le polynôme $P(x)$ par $G(x)$. Soient $Q(x)$ et $R(x)$ les polynômes quotient et reste ainsi obtenus :

$$P(x) = Q(x).G(x) \oplus R(x)$$

- le CRC (Cyclic Redundancy Check) est le reste $R(x)$ ainsi calculé. On remarque que le reste est forcément au maximum de degré $k - 1$.
- le message effectivement transmis est associé au polynôme $M'(x) = P(x) \oplus R(x)$.

Le message envoyé est donné par le message initial M suivi de la séquence de k bits correspondant à $R(x)$.

Codage

La procédure de codage consiste à :

- calculer $P(x) = M(x).x^k$.
  - ceci correspond à un décalage de k bits (vers la gauche) du message M.
- diviser le polynôme $P(x)$ par $G(x)$. Soient $Q(x)$ et $R(x)$ les polynômes quotient et reste ainsi obtenus :

$$P(x) = Q(x).G(x) \oplus R(x)$$

- le CRC (Cyclic Redundancy Check) est le reste $R(x)$ ainsi calculé. On remarque que le reste est forcément au maximum de degré $k - 1$.
- le message effectivement transmis est associé au polynôme $M'(x) = P(x) \oplus R(x)$.

Le message envoyé est donné par le message initial M suivi de la séquence de k bits correspondant à $R(x)$.

Codage

La procédure de codage consiste à :

- calculer $P(x) = M(x).x^k$.
  - ceci correspond à un décalage de k bits (vers la gauche) du message M.
- diviser le polynôme $P(x)$ par $G(x)$. Soient $Q(x)$ et $R(x)$ les polynômes quotient et reste ainsi obtenus :

$$P(x) = Q(x).G(x) \oplus R(x)$$

- le CRC (Cyclic Redundancy Check) est le reste $R(x)$ ainsi calculé. On remarque que le reste est forcément au maximum de degré $k - 1$.
- le message effectivement transmis est associé au polynôme $M'(x) = P(x) \oplus R(x)$.

Le message envoyé est donné par le message initial M suivi de la séquence de k bits correspondant à $R(x)$.

Décodage

La procédure de décodage de M' consiste à :

- calculer le polynôme $M'(x)$,
- diviser le polynôme $M'(x)$ par $G(x)$,
  - $M'(x)$ est divisible par $G(x)$:
    le CRC n'a pas détecté d'erreurs.
    Le message reçu est traité comme correct.
    Le message initial M est obtenu en ignorant les k derniers bits de M'
  - $M'(x)$ n'est pas divisible par $G(x)$: une erreur est détectée.

Décodage

La procédure de décodage de M' consiste à :

- calculer le polynôme $M'(x)$,
- diviser le polynôme $M'(x)$ par $G(x)$,
  - $M'(x)$ est divisible par $G(x)$:
    le CRC n'a pas détecté d'erreurs.
    Le message reçu est traité comme correct.
    Le message initial M est obtenu en ignorant les k derniers bits de M'
  - $M'(x)$ n'est pas divisible par $G(x)$: une erreur est détectée.

Décodage

La procédure de décodage de M' consiste à :

- calculer le polynôme $M'(x)$,
- diviser le polynôme $M'(x)$ par $G(x)$,
  - $M'(x)$ est divisible par $G(x)$:
    le CRC n'a pas détecté d'erreurs.
    Le message reçu est traité comme correct.
    Le message initial M est obtenu en ignorant les k derniers bits de M'
  - $M'(x)$ n'est pas divisible par $G(x)$: une erreur est détectée.

Décodage

La procédure de décodage de M' consiste à :

- calculer le polynôme $M'(x)$,
- diviser le polynôme $M'(x)$ par $G(x)$,
  - $M'(x)$ est divisible par $G(x)$:
    le CRC n'a pas détecté d'erreurs.
    Le message reçu est traité comme correct.
    Le message initial M est obtenu en ignorant les k derniers bits de M'
  - $M'(x)$ n'est pas divisible par $G(x)$: une erreur est détectée.

Propriétés du code polynomial

Propriétés

1. Un message M' transmis correctement a un polynôme $M'(x)$ divisible par le polynôme générateur $G(x)$.
2. Si $G(x)$ comporte au moins 2 termes, les erreurs simples sont détectables.
3. Si $G(x)$ a un facteur irréductible de 3 termes, les erreurs doubles sont détectables.
4. Si $G(x)$ est un multiple de $x \oplus 1$, les erreurs en nombre impair sont détectables.

$G(x)$ a un facteur irréductible de 3 termes ?

Definition

- Un polynôme $P(x)$ est irréductible s'il ne peux pas se décomposer en produit de deux polynômes non constants :

$$P(x) = P'(x) \cdot P''(x)$$

- Un polynôme $P(x)$ a un facteur irréductible de trois termes s'il peut être décomposé en

$$P(x) = P'(x) \cdot P''(x)$$

où $P'(x)$ est irreductible et a 3 termes.

Exemples de polynôme irréductible ayant 3 termes

- $x^2 \oplus x \oplus 1$
- $x^4 \oplus x^3 \oplus 1$

CRC : Cyclic Redundancy Check

Les polynômes utilisés en pratique sont assez longs :

CRC-16 : $x^{16} \oplus x^{15} \oplus x^2 \oplus 1$

CRC-32 : $x^{32} \oplus x^{26} \oplus x^{23} \oplus x^{22} \oplus x^{16} \oplus x^{12} \oplus x^{11} \oplus x^{10} \oplus x^8 \oplus x^7 \oplus x^5 \oplus x^4 \oplus x^2 \oplus x \oplus 1$