

IN2T - Concepts informatiques

Réseaux et internet

Protocole *Ethernet*

Utilisé couramment dans :

- Les réseaux câblés Ethernet
Cuivre et fibre optique
- Les réseaux sans fil Wifi
- Les réseaux sans fil Bluetooth
- ...

Protocole *Ethernet*

- Messages découpés en **paquets** qui contiennent entre autres :
 - Adresse MAC destination
 - Adresse MAC source
 - Somme de contrôle
- Livraison **non garantie**

Protocole *Ethernet* : Adresses MAC

Identifiant **unique** des appareils d'un réseau *Ethernet*

- *Media Access Control*
- Assignée **par le fabricant**
- 48 bits
préfixes de 24 bits dépend du fabricant
- En hexadécimal $\Rightarrow 5E : FF : 56 : A2 : AF : 15$

Protocole *Ethernet* : Réseau à collisions

- Tous les membres sont connectés au même fil
topologie en bus
- Tous les membres voient tous les paquets
- Un appareil à la fois peut émettre
- L'usage de concentrateurs (*hubs*) permet une topologie physique en étoile
mais se comporte toujours comme un bus

Protocole *Ethernet* : Réseau commuté

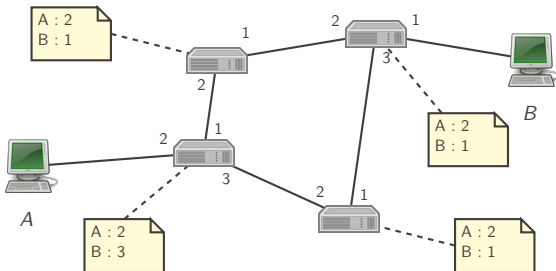
- Utilisation de commutateurs *Ethernet* (*switchs*)
- Chaque membres ne reçoit que ses paquets
- Grosse diminution des collisions
disparition avec le mode Full-duplex

Protocole Internet (IP)

- **Couche** au dessus de l'Ethernet
peut utiliser l'Ethernet ou autre chose
- Permet d'utiliser des **intermédiaires**
Pas besoin d'être directement connecté
- Permet une topologie générale de graphe
- Les membres intermédiaires sont appelés **routeur**
- Les messages sont découpé en **paquets**
Un paquet IP peut être transporté en plusieurs paquets Ethernet

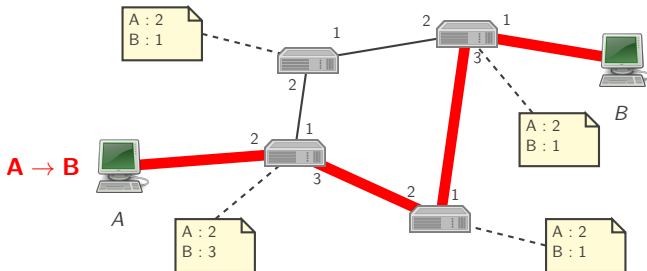
Protocole Internet (IP) : Routage

- Transfert **pas à pas** d'un paquet vers une destination
Le paquet est passé de machine en machine à travers Internet
- Chaque routeur possède une **table de routage**
Détermine sur quel lien envoyer un paquet reçu



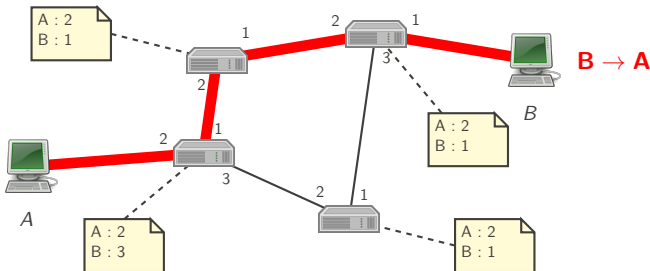
Protocole Internet (IP) : Routage

- Transfert **pas à pas** d'un paquet vers une destination
Le paquet est passé de machine en machine à travers Internet
- Chaque routeur possède une **table de routage**
Détermine sur quel lien envoyer un paquet reçu



Protocole Internet (IP) : Routage

- Transfert **pas à pas** d'un paquet vers une destination
Le paquet est passé de machine en machine à travers Internet
- Chaque routeur possède une **table de routage**
Détermine sur quel lien envoyer un paquet reçu



Protocole Internet (IP) : Adresses

■ IPv4

- Composées de 4 octets (32 bits)
- Généralement écrit au format décimal séparés par des points
Exemple : 192.168.1.1

■ IPv6

- Composées de 128 bits
- représenté en 8 groupe de 16 bits en hexadécimal
Exemple :
2001 : 0db8 : 0000 : 0042 : 0000 : 8a2e : 0370 : 7334

Protocole Internet (IP) : Sous-réseaux

- Division en réseaux local, *Local Area Network* (LAN) et réseaux étendu, *Wide Area Network* (WAN)
- Les machines qui peuvent se parler **directement** sont **dans un même LAN**
Via Ethernet par exemple
- Le **masque** de sous-réseau permet de reconnaître les adresses appartenant au LAN

IP : Masque de sous-réseaux

- Permet d'identifier la partie l'adresse IP qui **désigne le sous-réseau**
- A le même format qu'une adresse IP
- Le sous réseau est obtenu en faisant un **ET logique** entre les bits de l'adresse et ceux du masque

	192	.	168	.	1	.	2
	11000000	.	10101000	.	00000001	.	00000002
	255	.	255	.	255	.	0
ET	11111111	.	11111111	.	11111111	.	00000000
<hr/>							
	192	.	168	.	1	.	0
=	11000000	.	10101000	.	00000001	.	00000000

IP : Masque de sous-réseaux

Lorsqu'une machine veut envoyer un paquet à une autre

- Elle identifie son **propre** sous-réseau en utilisant le masque
- Elle identifie le sous-réseau du **destinataire** en utilisant le masque
- Si les deux sous-réseaux sont **identiques**, l'envoi se fait directement
Via Ethernet par exemple
- Sinon, la **table de routage** est utilisée.

IP : Passerelle par défaut

- C'est une **partie** de la table de routage
- Elle est utilisée lorsqu'**aucune autre** entrée de la table ne correspond au sous-réseau de destination
- Pour les utilisateurs finaux, la table de routage se limite à la passerelle par défaut

Address Resolution Protocol (ARP)

Comment envoyer un paquet à une machine du même sous-réseau dont on connaît l'IP mais pas l'adresse MAC ?

- C'est la fonction de ARP
- ARP est utilisé pour les adresses IPv4
En IPv6 c'est le Neighbor Discovery Protocol (NDP)

Address Resolution Protocol (ARP)

Fonctionnement :

- Envoie d'un *broadcast* (à toute les machines du sous-réseau)
"quelle est l'adresse MAC correspondant à l'adresse IP
X.X.X.X ? signé Y.Y.Y.Y"
- La machine concernée répond à Y.Y.Y.Y "je suis X.X.X.X,
mon adresse MAC est xx : xx : xx : xx : xx : xx"
- Les deux machines sauvent la correspondance dans leur cache
ARP.

Dynamic Host Configuration Protocol (DHCP)

Comment une machine connaît-elle ses propres adresse IP, masque de sous-réseau et passerelle par défaut ?

- Ces informations peuvent être configurées manuellement.
- L'utilisation d'un serveur DHCP permet une configuration automatique.

Dynamic Host Configuration Protocol (DHCP)

Fonctionnement :

- Envoie d'un *broadcast* "je suis xx : xx : xx : xx : xx : xx et j'ai besoin d'une adresse IP"
- Le ou les serveurs DHCP du sous-réseau lui répondent "je suis Y.Y.Y.Y. Je peux te proposer l'adresse X.X.X.X"
- La machine choisit la première réponse DHCP qu'elle reçoit et renvoie un broadcast pour informer tous les serveurs DHCP de son choix.
- Le serveur DHCP choisi envoie les autres paramètres

Transmission Control Protocol (TCP)

Problème : protocole IP ne garanti ni l'**arrivée** ni l'**ordre** des paquets

- Solution : TCP

- Basé sur IP

On parle souvent de TCP/IP

- Protocole **connecté** \Rightarrow 3 phases :

- Connexion

- Envoi de données

- Déconnexion

- garanti l'arrivée et l'ordre des paquets

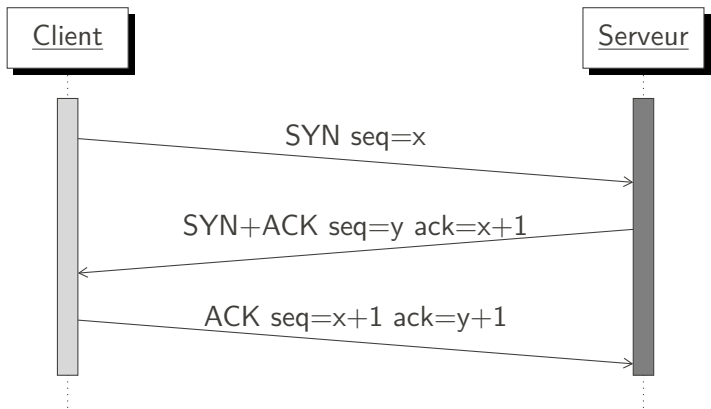
Transmission Control Protocol (TCP)

Les messages contiennent :

- Les numéros de **port** source et destination
- Des *flags* qui peuvent être à **1** ou à **0**
NS, CWR, ECE, URG, ACK, PSH, RST, SYN, FIN
- Un numéro de séquence
Pour maintenir l'ordre des paquets
- Un numéro d'acquittement
Contient le prochain numéro de séquence attendu
- **Somme de contrôle**
- ...

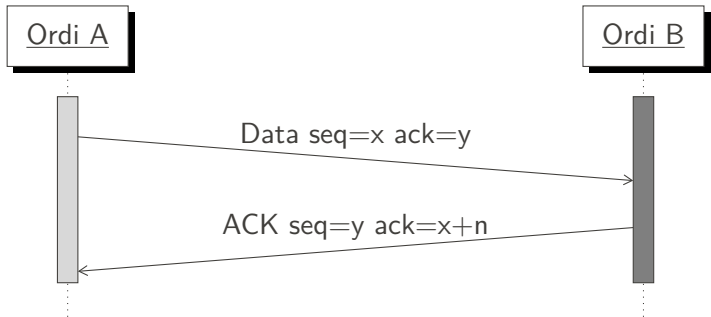
TCP : Connexion

Permet de **syn**chroniser les numéros de séquence. Le client demande une connexion à un serveur qui écoute et peut l'accepter.



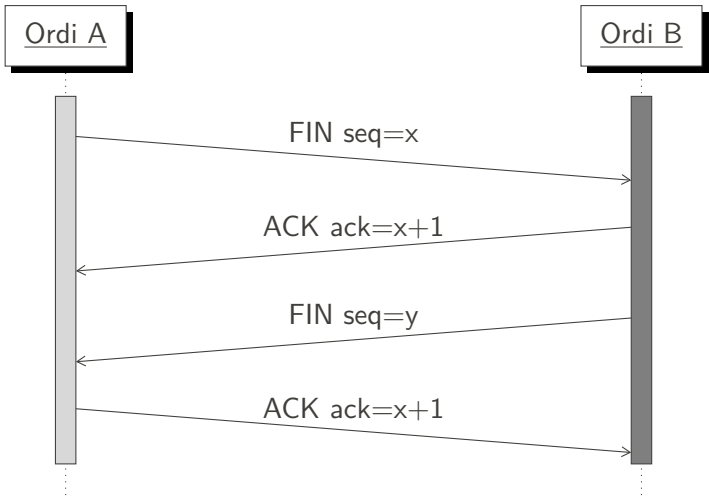
TCP : Envoi de données

Envoi de n octets. Dans un sens comme dans l'autre



TCP : Déconnexion

Peut être initiée par l'un ou l'autre des ordinateurs



Dynamic Name System (DNS)

Les adresses IP ne sont pas pratiques pour les **humains**.

- DNS = répertoire qui associe un nom à une IP

On parle de la résolution d'un nom de domaine

- Les correspondances sont stockées dans une **hiérarchie** de serveurs

- Des serveurs récursifs s'occupent de parcourir l'arborescence et de maintenir une **cache** des résolutions effectuées

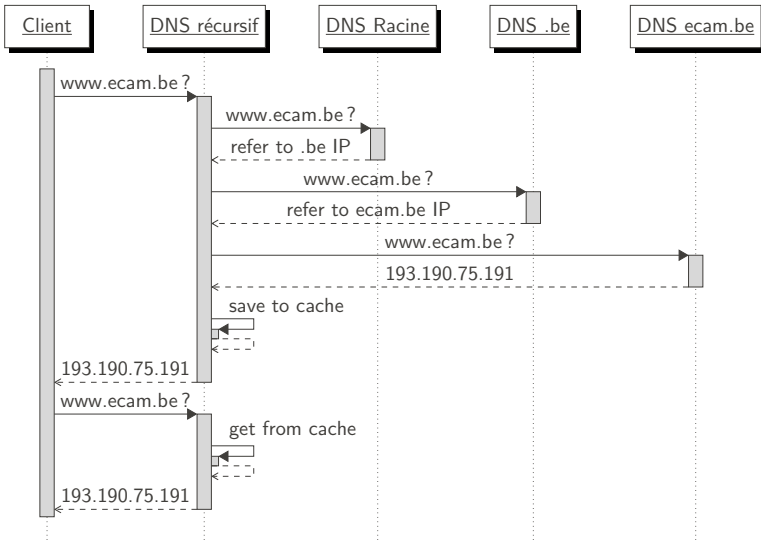
- Un client obtient l'adresse d'un serveur DNS récursif

Généralement par DHCP

- Les messages sont envoyés en UDP ou TCP

*UDP fournit **uniquement** les ports et une somme de contrôle*

DNS : Résolution



HyperText Transfer Protocol (HTTP)

- Protocole de **haut niveau**
- Permet l'accès aux **ressources** sur internet
- Protocole client/serveur
 - Le client envoie une requête, le serveur répond*
- La requête et la réponse sont composées d'un **entête** et optionnellement d'un **corp**.
- L'entête est du texte, peut être du texte ou du binaire.
- Les requêtes et réponses sont généralement envoyée en TCP

HyperText Transfer Protocol (HTTP)

- La requête commence par l'établissement d'une connexion TCP.
- La requête proprement dite :

```
GET / http/1.1  
Host: www.perdu.com
```

- Et la réponse :

```
hTTP/1.1 200 OK  
Date: Thu, 22 Nov 2018 19:53:23 GMT  
Content-Length: 204  
Content-Type: text/html
```

```
<html><head><title>Vous Etes Perdu ?</title></head><body>  
<h1>Perdu sur l'Internet ?</h1><h2>Pas de panique, on va  
vous aider</h2><strong><pre>    * <----- vous &ecirc;tes  
ici</pre></strong></body></html>
```

HyperText Transfer Protocol Secure (HTTPS)

2 problèmes :

- La **confiance** en le serveur DNS
La bonne résolution des IP en dépends
- Le contenu des requêtes et des réponses passent en **clair** sur le réseaux

Pour régler ces problèmes, HTTPS utilise 3 concepts :

- Le cryptage
- La signature numérique
- Des autorités de certification

Le cryptage

- Le cryptage se base sur une fonction à sens unique qui utilise une clé d'encryptage $eKey$

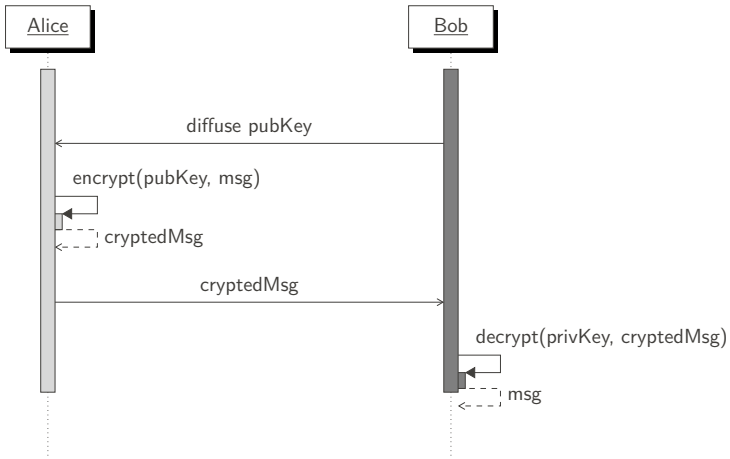
Appelons la $encrypt(eKey, message) \rightarrow cryptedMsg$

- Il est très difficile (long) de retrouver le message à partir de $cryptedMsg$ sauf si on connaît $dKey$, la clé de décryptage.

$decrypt(dKey, cryptedMsg) \rightarrow message$

- La clé d'encryptage est appelée clé publique, la clé de décryptage clé privée
- Ce cryptage est appelé asymétrique. Un cryptage symétrique utilise la même clé pour l'encryptage et le décryptage.

Le cryptage

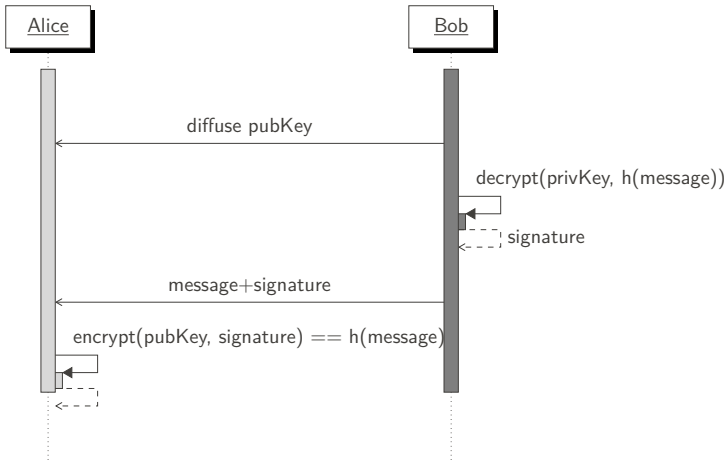


La signature numérique

Permet de s'assurer :

- de l'expéditeur d'un message
- que le message n'a pas été modifié

La signature numérique



Les autorités de certification

Fournit des certificats qui contiennent :

- Une clé privée
- Une clé publique
- Le ou les **noms de domaine** liés au certificat
- La signature de l'autorités de certification
La signature porte sur la clé publique et les noms de domaine

HTTPS Connexion

