Réseaux et Systèmes (MIN 15112)

2025 - 2026

Zhiyi Zhang

zhiyi.zhang@uvsq.fr

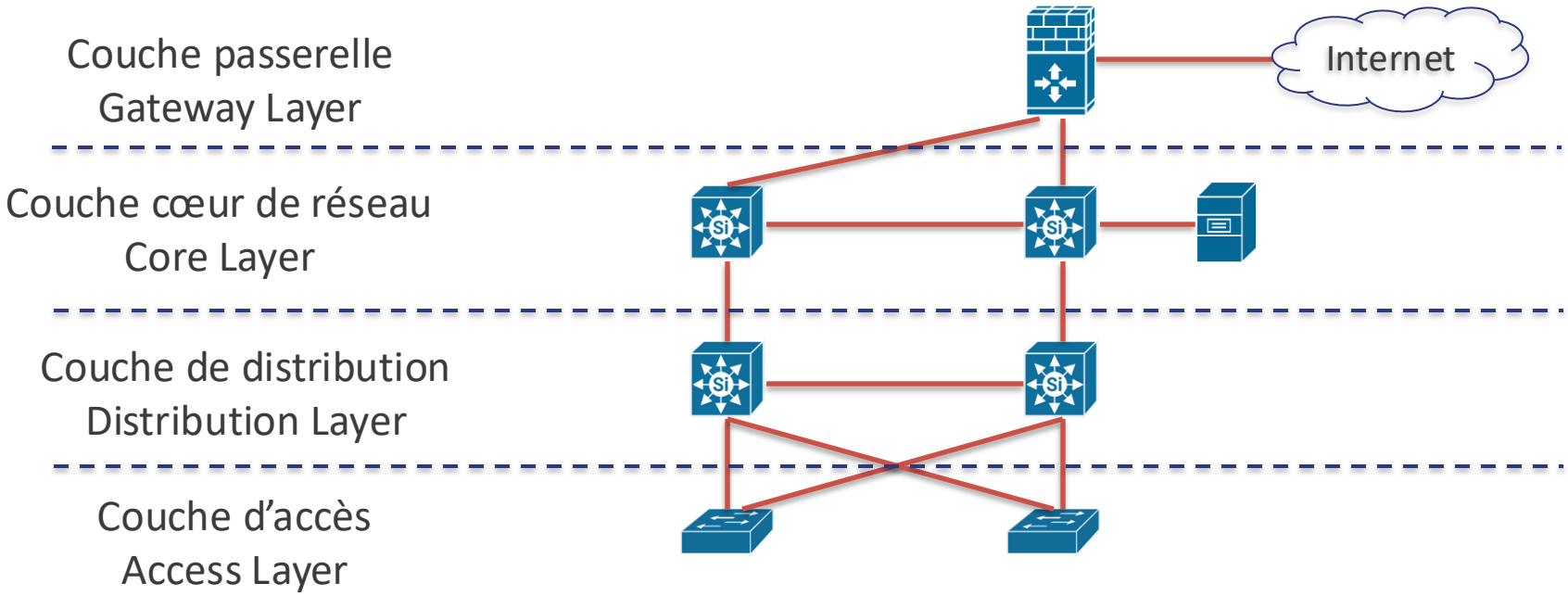
https://www.zhiyizhang.com

UVSQ

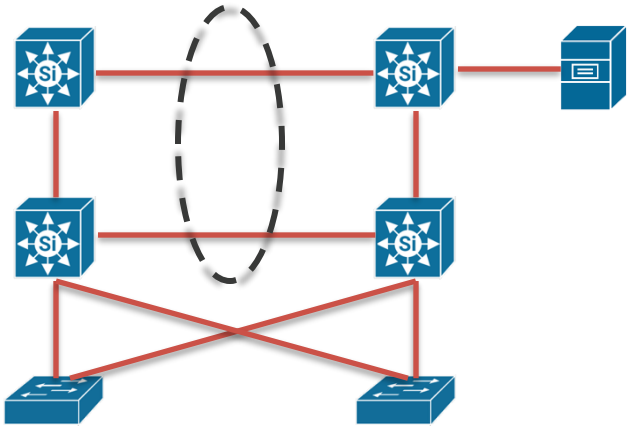
université PARIS-SACLAY

EtherChannel

Rappel : Architecture LAN



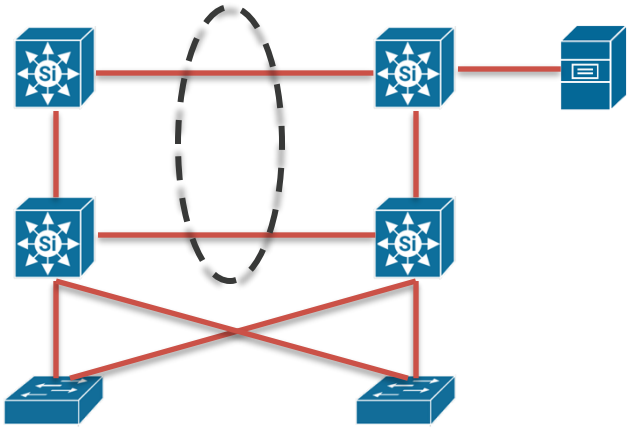
Contexte EtherChannel



Le trafic interne au réseau est important, mais l'interconnexion entre les commutateurs est limitée par le débit physique de leurs ports.

Solution ?

Contexte EtherChannel

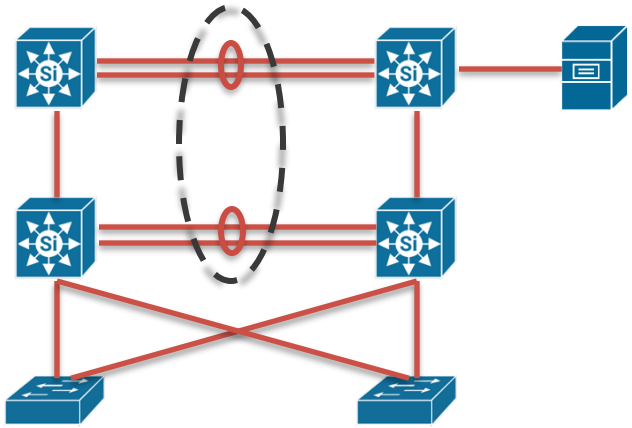


Le trafic interne au réseau est important, mais l'interconnexion entre les commutateurs est limitée par le débit physique de leurs ports.

- Nouveau équipements/ports (investissement financier)

Est-il possible d'avoir une solution à moindre coût tout en conservant l'équipement existant ?

Contexte EtherChannel



Le trafic interne au réseau est important, mais l'interconnexion entre les commutateurs est limitée par le débit physique de leurs ports.

- Nouveau équipements/ports (investissement financier)
- EtherChannel

Avec EtherChannel, plusieurs liaisons physiques relient les commutateurs, mais elles sont considérées comme une seule liaison logique.

EtherChannel

- Solution permettant d’augmenter le débit;
- Agrégation logique de plusieurs liens;
- Vue comme un seul lien logique;
- Load balancing et la redondance.

PAgP & LACP

Protocoles permettant de négocier la création et le maintien d'un lien EtherChannel

- PAgP est un protocole propriétaire Cisco
- LACP correspond au standard IEEE 802.3ad

Configuration EtherChannel statique sans protocole possible

Modes PAgP

- On : Membre d'un EtherChannel sans négociation, sans protocole
- Desirable : Demande activement à l'autre côté s'il peut/veut créer un EtherChannel
- Auto : Attend passivement que l'autre côté initie la création de l'EtherChannel
- Off : Aucun EhterChannel n'est configuré sur l'interface

- On – On : Oui
- On/Desirable/Auto – Off : Non
- Desirable/Auto – Desirable : Oui
- Auto – Auto : Non

Modes LACP

- On : Membre d'un EtherChannel sans négociation, sans protocole
- Active : Demande activement à l'autre côté s'il peut/veut créer un EtherChannel
- Passive : Attend passivement que l'autre côté initie la création de l'EtherChannel
- Off : Aucun EhterChannel n'est configuré sur l'interface

- On – On : Oui
- On/Active/Passive – Off : Non
- Active/Passive – Active : Oui
- On/Passive – Passive : Non

EtherChannel

Une fois la configuration d'un EtherChannel établie, une interface logique est créée. Toutes les configurations ultérieures doivent alors être appliquées sur cette interface logique, et non plus directement sur les interfaces physiques.

Un EtherChannel peut regrouper **jusqu'à 8 liens physiques**.

Les ports physiques du même EtherChannel doivent respecter les conditions suivantes :

- Même débit et même mode duplex
- Même mode d'interface : access/trunk
- Si les interfaces sont en access : même VLAN
- Si les interfaces sont en trunk : même VLAN natif et même liste de VLAN autorisée

UVSQ

université PARIS-SACLAY

VLSM
(Variable Length Subnet Mask)

Adresse IPv4

Une adresse IPv4 est un identifiant de **32 bits** utilisé pour identifier un hôte dans un réseau IP (couche 3). Elle est généralement représentée en notation **décimale pointée**.

Exemple :

192.168.10.1

11000000.10101000.00001010.00000001

Décimal <-> binaire

128	64	32	16	8	4	2	1

Exemple : Convertir 202 en binaire

202 > 128 -> **1** (reste 74)

74 > 64 ->**1** (reste 10)

10 < 32 -> **0**

10 < 16 -> **0**

10 > 8 -> **1** (reste 2)

2 < 4 -> **0**

2 = 2 -> **1**

0 < 1 -> **0**

Décimal <-> binaire

128	64	32	16	8	4	2	1

Exemple : Convertir 202 en binaire

202 > 128 -> **1** (reste 74)

74 > 64 ->**1** (reste 10)

10 < 32 -> **0**

10 < 16 -> **0**

10 > 8 -> **1** (reste 2)

2 < 4 -> **0**

2 = 2 -> **1**

0 < 1 -> **0**

Exemple : Convertir 11001010 en décimal

128 + 64 +8 + 2 = 202

Classes d’adresses IPv4

	8 bits	8 bits	8 bits	8 bits	Premier octet
A	0xxxxxxx	Hôte	Hôte	Hôte	0 – 127
B	10xxxxxx	Réseau	Hôte	Hôte	128 – 191
C	110xxxxx	Réseau	Réseau	Hôte	192 – 223
D	1110xxxx	Multicast	Multicast	Multicast	224 – 239
E	Réservé				240 – 255

Classes d'adresses IPv4

	8 bits	8 bits	8 bits	8 bits	Premier octet
A	**0xxxxxxx**	Hôte	Hôte	Hôte	0 – 127
B	**10xxxxxx**	**Réseau**	Hôte	Hôte	128 – 191
C	**110xxxxx**	**Réseau**	**Réseau**	Hôte	192 – 223
D	1110xxxx	Multicast	Multicast	Multicast	224 – 239
E	Réservé				240 – 255

Bits réseau/hôte

Dans une adresse IPv4, les 32 bits sont divisés en deux parties :

- Les bits du réseau (Net-ID)
- Les bits de l'hôte (Host-ID)

Cette séparation permet d'identifier le réseau et les hôtes (machines) dans ce réseau.

Net-ID : identifiant du réseau, même pour toutes les machines du réseau, déterminé par le masque de sous-réseau

Host-ID : identifiant pour chaque machine dans le réseau

Masque de sous-réseau

Le masque de sous-réseau est un ensemble de 32 bits utilisé en IPv4 pour séparer la partie réseau et la partie hôte d'une adresse IP.

Structure : une suite de 1 continus, suivie de 0

Exemple : 11111111.11111111.11110000.00000000 (/20) -> 12 bits Host-ID

Il permet de déterminer :

- À quel réseau appartient une adresse IP
- Combien d'hôtes peuvent exister dans ce réseau
- Si deux adresses se trouvent dans le même réseau

Types d'adresses IPv4

Trois types d'adresses existent :

- **Adresse réseau** : Net-ID, identifie le sous-réseau lui-même (routage)
  Tous les bits hôte = 0, ne peut pas être attribuée à une machine
  Exemple : 192.168.1.154/24 -> **192.168.1.0/24**
- **Adresse d'hôte** : adresses utilisables par les machines du réseau
  Net-ID identique, bits hôte ≠ tout 0 et ≠ tout 1
  Exemple : 192.168.1.0/24, plage d'adresses : 192.168.1.1/24 – 192.168.1.254/24
- **Adresse broadcast** : permet d'envoyer un message à tous les hôtes du sous-réseau
  Tous les bits hôte = 1, ne peut pas être attribuée à une machine
  Exemple : 192.168.1.0/24 -> **192.168.1.255/24**

Types d'adresses IPv4

Trois types d'adresses existent :

- **Adresse réseau** : Net-ID, identifie le sous-réseau lui-même (routage)
  Tous les bits hôte = 0, ne peut pas être attribuée à une machine
  Exemple : 192.168.1.154/24 -> **192.168.1.0/24**
- **Adresse d'hôte** : adresses utilisables par les machines du réseau
  Net-ID identique, bits hôte ≠ tout 0 et ≠ tout 1
  Exemple : 192.168.1.0/24, plage d'adresses : 192.168.1.1/24 – 192.168.1.254/24
- **Adresse broadcast** : permet d'envoyer un message à tous les hôtes du sous-réseau
  Tous les bits hôte = 1, ne peut pas être attribuée à une machine
  Exemple : 192.168.1.0/24 -> **192.168.1.255/24**

Convention : la passerelle d'un réseau est la première ou la dernière adresse utilisable.

Diviser un réseau en sous-réseaux

- Réduction de la taille des domaines de broadcast
- Meilleure organisation et gestion du réseau
- Optimisation de l’utilisation des adresses IP
- Support de multi-VLAN
- Amélioration du routage

VLSM

Exemple : On dispose 172.16.0.0/16

172.16	**0**								**0**
172.16	0	0	0	0	0	0	0	0	0
Net-ID				…			Host-ID		
172.16	**255**								**255**
172.16	1	1	1	1	1	1	1	1	255

Au départ,
On a un réseau
172.16.0.0/16

Plage d'adresses :
172.16.0.1/16 –
172.16.255.254/16

Adresse broadcast :
172.16.255.255

VLSM

Exemple : On dispose 172.16.0.0/16

172.16		**0**							**0**
172.16	0	0	0	0	0	0	0	0	0

Net-ID Host-ID

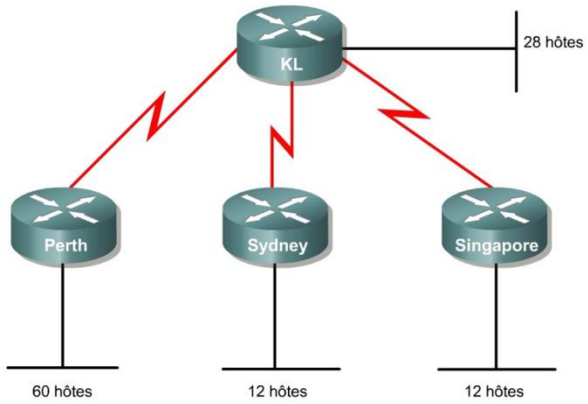
Sous-réseau 1 : 172.16.0.0
Masque : 255.255.128.0
Hôtes : 172.16.0.1 – 172.16.127.254

172.16		**128**							**0**
172.16	1	0	0	0	0	0	0	0	0

Sous-réseau 2 : 172.16.128.0
Masque : 255.255.128.0
Hôtes : 172.16.128.1 – 172.16.255.254

Subnet, **on emprunte des bits à la partie hôte**

VLSM



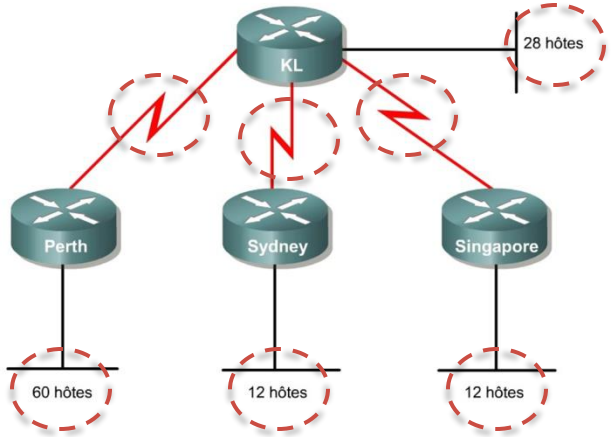
Étape 1 :

On doit d'abord déterminer combien de réseaux sont nécessaires dans la topologie, et le nombre d'adresses hôtes requis pour chaque réseau.

Combien de réseaux ?

Combien d'adresses hôtes requis ?

VLSM

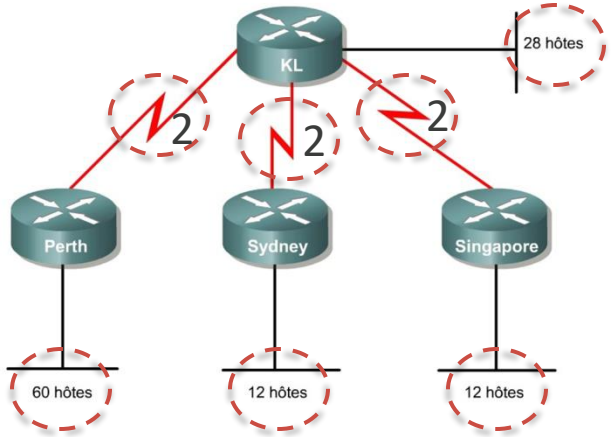


Étape 1 :

On doit d'abord déterminer combien de réseaux sont nécessaires dans la topologie, et le nombre d'adresses hôtes requis pour chaque réseau.

Le nombre de réseaux : 7

VLSM



Étape 1 :

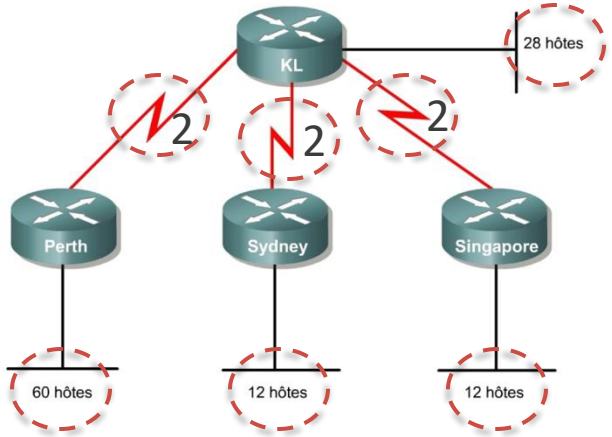
On doit d'abord déterminer combien de réseaux sont nécessaires dans la topologie, et le nombre d'adresses hôtes requis pour chaque réseau.

Le nombre de réseaux : 7

Les nombre d'adresses requis :

28; 60; 12; 12; 2; 2; 2.

VLSM



Étape 2 :

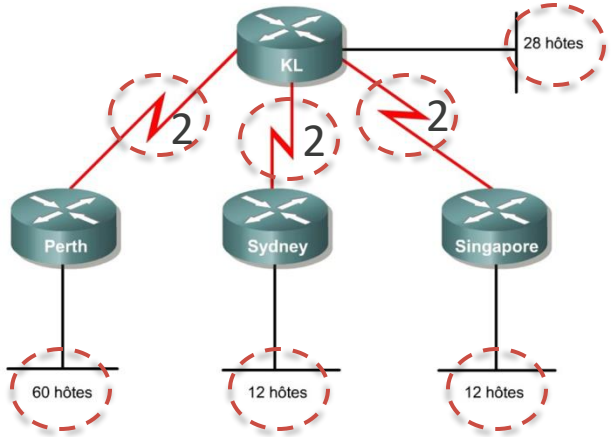
Arranger les besoins en nombre d'adresses hôtes, du plus grand au plus petit.

Les nombre d'adresses requis :

28; 60; 12; 12; 2; 2; 2.

60; 28; 12; 12; 2; 2; 2.

VLSM



Étape 3 :

Déterminer le nombre de bits hôte nécessaires pour chaque sous-réseau, puis en déduire le masque de sous-réseau correspondant.

Les nombre d'adresses requis :

60; 28; 12; 12; 2; 2; 2.

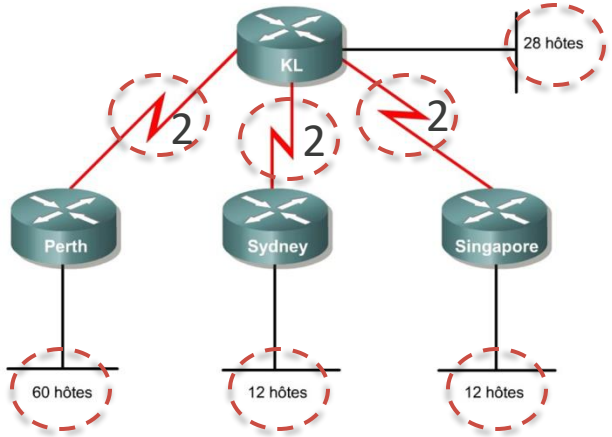
$2^6 - 2 \geq 60$, donc 6 bits nécessaires => /26

$2^5 - 2 \geq 28$, donc 5 bits nécessaires => /27

$2^4 - 2 \geq 12$, donc 4 bits nécessaires => /28

$2^2 - 2 \geq 2$, donc 2 bits nécessaires => /30

VLSM



192.168.10.0/24

Étape 4 :

On commence l'attribution des adresses.

1 sous-réseau /26, 1 sous-réseau /27,

2 sous-réseaux /28, 3 sous-réseaux /30

192.168.10.	0	0	0	0	0	0	0	0
192.168.10.	0	0	1	1	1	1	1	1

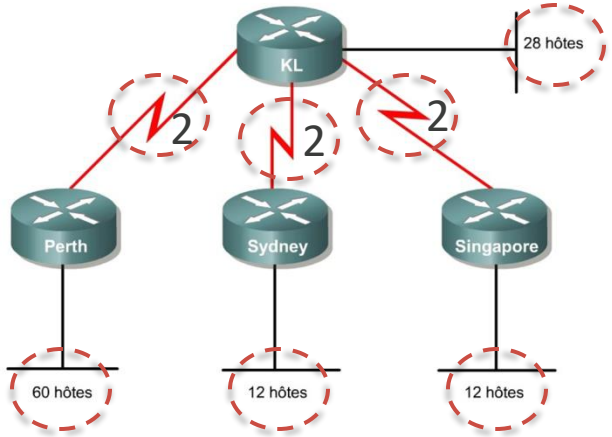
Sous-réseau Perth : 192.168.10.0/26

Masque : 255.255.255.192

Hôtes : 192.168.10.1 – 192.168.10.62

@Broadcast : 192.168.10.63

VLSM



192.168.10.0/24

Étape 4 :

On commence l'attribution des adresses.

1 sous-réseau /26, 1 sous-réseau /27,

2 sous-réseaux /28, 3 sous-réseaux /30

192.168.10.	0	**1**	0	0	0	0	0	0
192.168.10.	0	**1**	0	1	1	1	1	1

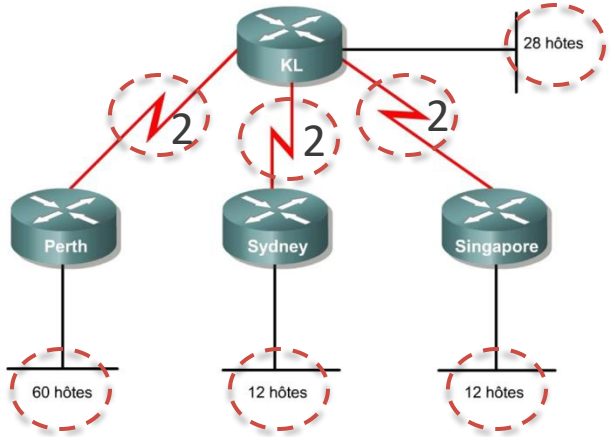
Sous-réseau KL : 192.168.10.64/27

Masque : 255.255.255.224

Hôtes : 192.168.10.64 – 192.168.10.94

@Broadcast : 192.168.10.95

VLSM



192.168.10.0/24

Étape 4 :

On commence l'attribution des adresses.

1 sous-réseau /26, 1 sous-réseau /27,

2 sous-réseaux /28, 3 sous-réseaux /30

192.168.10.	0	1	**1**	0	0	0	0	0
192.168.10.	0	1	**1**	0	1	1	1	1

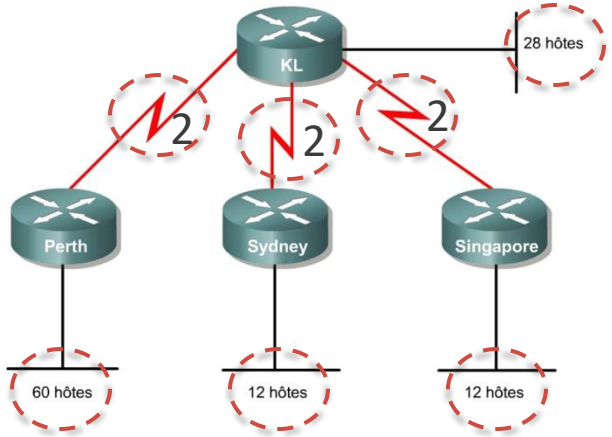
Sous-réseau Sydney : 192.168.10.96/28

Masque : 255.255.255.240

Hôtes : 192.168.10.97 – 192.168.10.110

@Broadcast : 192.168.10.111

VLSM



192.168.10.0/24

Étape 4 :

On commence l'attribution des adresses.

1 sous-réseau /26, 1 sous-réseau /27,

2 sous-réseaux /28, 3 sous-réseaux /30

192.168.10.	0	1	1	**1**	0	0	0	0
192.168.10.	0	1	1	**1**	1	1	1	1

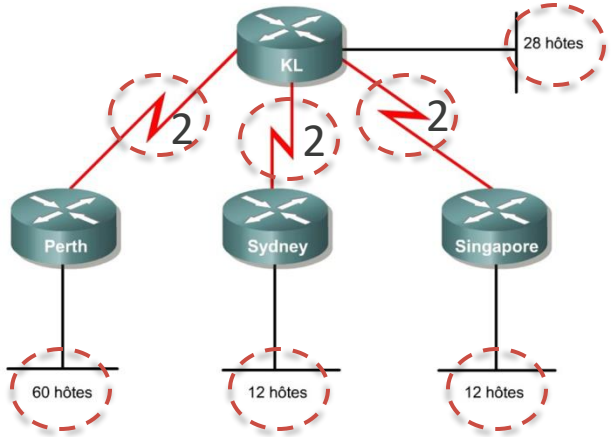
Sous-réseau Singapore : 192.168.10.112/28

Masque : 255.255.255.240

Hôtes : 192.168.10.113 – 192.168.10.126

@Broadcast : 192.168.10.127

VLSM



192.168.10.0/24

Étape 4 :

On commence l'attribution des adresses.

1 sous-réseau /26, 1 sous-réseau /27,

2 sous-réseaux /28, 3 sous-réseaux /30

192.168.10.	**1**	**0**	**0**	**0**	0	0	0	0
192.168.10.	**1**	**0**	**0**	**0**	0	0	1	1

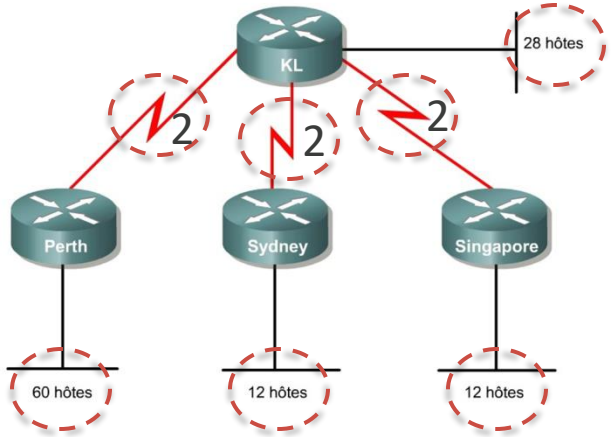
Sous-réseau KL - Perth : 192.168.10.128/30

Masque : 255.255.255.252

Hôtes : 192.168.10.129 – 192.168.10.130

@Broadcast : 192.168.10.131

VLSM



192.168.10.0/24

Étape 4 :

On commence l'attribution des adresses.

1 sous-réseau /26, 1 sous-réseau /27,

2 sous-réseaux /28, 3 sous-réseaux /30

192.168.10.	1	0	0	0	0	**1**	0	0
192.168.10.	1	0	0	0	0	**1**	1	1

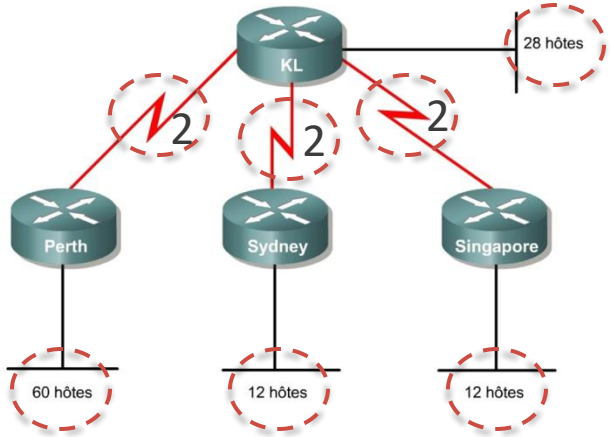
Sous-réseau KL - Sydney : 192.168.10.132/30

Masque : 255.255.255.252

Hôtes : 192.168.10.133 – 192.168.10.134

@Broadcast : 192.168.10.135

VLSM



192.168.10.0/24

Étape 4 :

On commence l'attribution des adresses.

1 sous-réseau /26, 1 sous-réseau /27,

2 sous-réseaux /28, 3 sous-réseaux /30

192.168.10.	1	0	0	0	**1**	**0**	0	0
192.168.10.	1	0	0	0	**1**	**0**	1	1

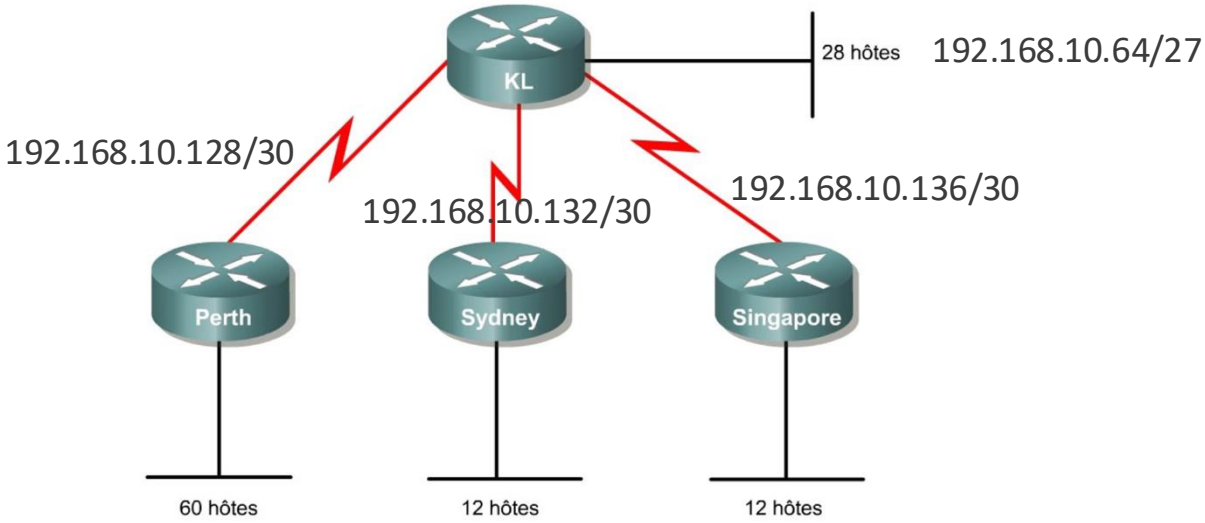
Sous-réseau KL - Singapore : 192.168.10.136/30

Masque : 255.255.255.252

Hôtes : 192.168.10.137 – 192.168.10.138

@Broadcast : 192.168.10.139

VLSM


KL
28 hôtes
192.168.10.64/27
192.168.10.128/30
192.168.10.132/30
192.168.10.136/30
Perth
Sydney
Singapore
60 hôtes
12 hôtes
12 hôtes
192.168.10.0/26
192.168.10.96/28
192.168.10.112/28

UVSQ

université PARIS-SACLAY

DHCP
(Dynamic Host Configuration Protocol)

DHCP

DHCP (Dynamic Host Configuration Protocol)

Basé sur UDP, port 67 (serveur) et 68 (client)

RFC 2131

Le DHCP permet d'attribuer automatiquement des paramètres IP aux hôtes d'un réseau.

La configuration automatique permet de gagner du temps, de simplifier la gestion et de réduire les risques d'erreurs.

Remarque : Le DHCP est principalement utilisé dans IPv4. Avec IPv6, une partie de la configuration d'adresse est assurée par le protocole NDP (Neighbor Discovery Protocol).

Fonctions principales

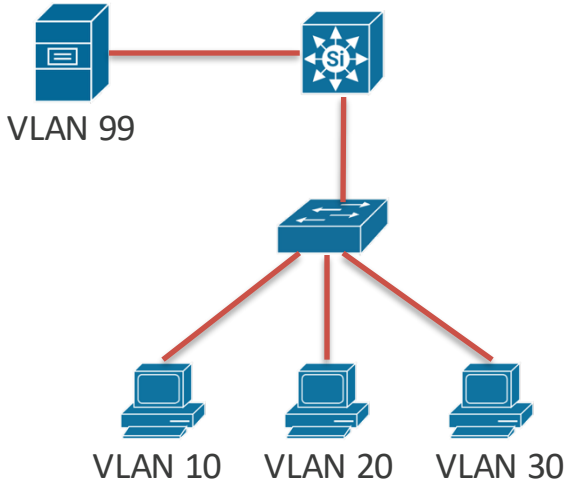
- Attribuer une adresse IP
- Fournir le masque de sous-réseau
- Fournir la passerelle par défaut
- Fournir les serveurs DNS
- Attribuer la durée du bail

Fonctionnement

DHCP suit un échange en 4 étapes

1. DHCP Discover (Broadcast) : le client cherche un serveur DHCP
2. DHCP Offer (Broadcast) : le serveur propose une adresse IP avec les paramètres
3. DHCP Request (Broadcast) : Le client demande officiellement cette adresse
4. DHCP Ack (Broadcast) : Le serveur confirme l'attribution

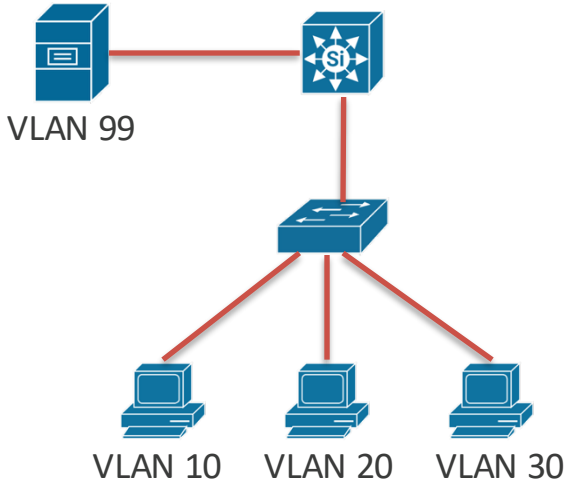
DHCP relay



Le serveur DHCP et les clients ne se trouvent pas forcément dans le même réseau. On peut utiliser un seul serveur DHCP pour attribuer des adresses à des clients situés dans plusieurs VLANs.

Problème ?

DHCP relay



Le serveur DHCP et les clients ne se trouvent pas forcément dans le même réseau. On peut utiliser un seul serveur DHCP pour attribuer des adresses à des clients situés dans plusieurs VLANs.

Le client émet des requêtes en broadcast, mais les équipements de couche 3 filtrent les broadcasts. Donc, les requêtes DHCP ne peuvent pas traverser un équipement de couche 3.

On utilise DHCP relay pour relayer les requêtes DHCP d'un réseau à un autre.

DHCP relay

Du client DHCP jusqu'au serveur DHCP, il est parfois nécessaire de traverser plusieurs réseaux. Mais, il n'est pas nécessaire de configurer le DHCP relay sur toutes les interfaces de couche 3. **On configure le DHCP relay sur la première interface de couche 3 qui bloque les paquets broadcast (la passerelle par défaut du client DHCP).**

DHCP relay

Du client DHCP jusqu'au serveur DHCP, il est parfois nécessaire de traverser plusieurs réseaux. Mais, il n'est pas nécessaire de configurer le DHCP relay sur toutes les interfaces de couche 3. **On configure le DHCP relay sur la première interface de couche 3 qui bloque les paquets broadcast (la passerelle par défaut du client DHCP).**

La communication entre l'agent de relais DHCP et le serveur DHCP se fait en unicast.

Le serveur DHCP utilise l'adresse de la passerelle pour déterminer dans quel réseau se trouve le client et ainsi sélectionner le bon pool d'adresses.

DHCP binding manuel

Dans un réseau, certains équipements (serveurs, imprimantes, caméras …) peuvent nécessiter une adresse IP fixe. On peut leur attribuer cette adresse via le DHCP.

Pour cela, deux actions sont nécessaires :

- Exclure ces adresses IP du pool DHCP destiné aux clients.
- Associer manuellement chaque adresse IP à une adresse MAC ou à un identifiant client (client-ID).

UVSQ

université PARIS-SACLAY

DNS
(Domain Name System)

DNS

DNS (Domain Name System)

RFC 882

Service d'annuaire distribué depuis 1985

- Stocke la hiérarchie des noms de domaine et les espaces d'adressage
- Délègue la gestion directe des domaines aux serveurs de noms faisant autorité

Traduction du nom canonique vers l'adresse IP

- Exemple : www.uvsq.fr -> 193.51.31.90

Informations techniques

- Adresses IP (A pour IPv4, AAAA pour IPv6), serveurs mail (MX), serveurs de noms (NS), alias de noms de domaine (CNAME), *etc.*

Architecture DNS

- Basé sur UDP, port : 53
- 13 serveurs racines (10 US, 2 Europe, 1 Asie)
  200+ serveurs, 50+ pays (Anycast)

List of Root Servers

HOSTNAME	IP ADDRESSES	OPERATOR
a.root-servers.net	198.41.0.4, 2001:503:ba3e::2:30	Verisign, Inc.
b.root-servers.net	170.247.170.2, 2801:1b8:10::b	University of Southern California, Information Sciences Institute
c.root-servers.net	192.33.4.12, 2001:500:2::c	Cogent Communications
d.root-servers.net	199.7.91.13, 2001:500:2d::d	University of Maryland
e.root-servers.net	192.203.230.10, 2001:500:a8::e	NASA (Ames Research Center)
f.root-servers.net	192.5.5.241, 2001:500:2f::f	Internet Systems Consortium, Inc.
g.root-servers.net	192.112.36.4, 2001:500:12::d0d	US Department of Defense (NIC)
h.root-servers.net	198.97.190.53, 2001:500:1::53	US Army (Research Lab)
i.root-servers.net	192.36.148.17, 2001:7fe::53	Netnod
j.root-servers.net	192.58.128.30, 2001:503:c27::2:30	Verisign, Inc.
k.root-servers.net	193.0.14.129, 2001:7fd::1	RIPE NCC
l.root-servers.net	199.7.83.42, 2001:500:9f::42	ICANN
m.root-servers.net	202.12.27.33, 2001:dc3::35	WIDE Project

Source : iana.org

Domain Name

TLD (Top Level Domains)

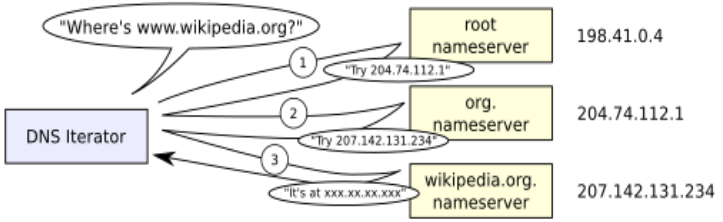
- Géré par l'ICANN
- Infrastructure TLD : ARPA
- Country-code TLD (ccTLD) : .fr /.cn /.jp /.kr …
- Sponsored TLD (sTLD) : .gov /.edu /.biz …
- Generic TLD (gTLD) : .com /.net/ .org …

La délégation est toujours utilisée

DNS itérative

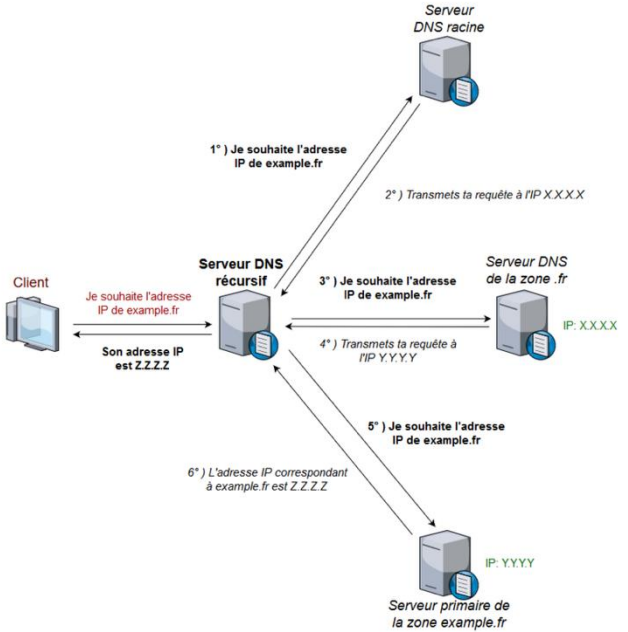
Chaque requête DNS répond directement au client avec une adresse d'au autre serveur DNS à interroger.

Le client continue d'interroger les serveurs DNS jusqu'à ce que l'un d'eux réponde avec l'adresse IP correcte pour le domaine concerné.



DNS récursif

Un serveur DNS communique avec plusieurs autres serveurs DNS pour chercher l'adresse IP et la renvoyer au client.



Cache DNS

Le navigateur, le système d'exploitation, le système DNS … conservent tous un cache DNS. Le cache est conservé pendant la durée définie par le TTL (Time To Live) indiqué dans l'enregistrement DNS.

Pourquoi utiliser un cache DNS ?

- Accélère la résolution DNS
- Réduit la charge sur les serveurs DNS
- Diminue le trafic réseau

Il faut faire attention à la mise à jour.

whois

whois

whois permet d'obtenir des infos sur un nom de domaine ou une adresse IP.

Il affiche généralement :

- Les infos du domaine : date de création, date d'expiration, dernière mise à jour…
- Le registrar : le bureau d'enregistrement responsable du domaine
- Les serveurs DNS associé au domaine
- Les contacts (admin, technique…) : souvent masqué

Pour une adresse IP, whois affiche l'organisation propriétaire, la région Internet (RIPE, ARIN…) et les informations de contact.

nslookup

nslookup est un outil qui permet d'interroger des serveurs DNS.

Il sert principalement à :

- Résoudre un nom de domaine en adresse IP
- Trouver le nom associé à une adresse IP (reverse lookup)
- Interroger un serveur DNS spécifique
- Vérifier les enregistrements DNS (A, AAAA, MX, NS, CNAME …)

```
zhiyi@MacBook-Pro-de-Zhiyi ~ % nslookup -type=MX uvsq.fr
Server:         193.51.24.1
Address:        193.51.24.1#53

uvsq.fr mail exchanger = 50 mx1.relay.renater.fr.
uvsq.fr mail exchanger = 50 mx2.relay.renater.fr.
```

dig

dig (Domain Information Groper) est un outil pour interroger les serveurs DNS.

Il est plus puissant et plus détaillé que nslookup.

```
zhiyi@MacBook-Pro-de-Zhiyi ~ % dig @1.1.1.1 www.uvsq.fr

; <<>> DiG 9.10.6 <<>> @1.1.1.1 www.uvsq.fr
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 50396
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
;; QUESTION SECTION:
;www.uvsq.fr.                   IN      A

;; ANSWER SECTION:
www.uvsq.fr.            0       IN      CNAME   k67-dev.uvsq.fr.
k67-dev.uvsq.fr.        85772   IN      A       193.51.31.90

;; Query time: 6 msec
;; SERVER: 1.1.1.1#53(1.1.1.1)
;; WHEN: Mon Nov 24 16:00:36 CET 2025
;; MSG SIZE  rcvd: 78
```

UVSQ

université PARIS-SACLAY

ICMP
(Internet Control Message Protocol)

ICMP

ICMP (Internet Control Message Protocol)

RFC 792

- Protocole de la couche réseau, utilisé pour le diagnostic et la contrôle
- Principalement utilisé par les routeurs lorsqu'un paquet ne peut pas être acheminé.
  - L'en-tête permet d'identifier le type de problème
- Les paquets ICMP sont principalement renvoyés vers la source
- ICMP ne sert pas seulement aux erreurs
  - ping & traceroute

Message ICMP

ICMP est encapsulé dans paquet IP, mais ICMP n'est pas un protocole de transport, il est considéré comme **une extension du protocole IP**.

L'en-tête ICMP

- Type de message (8 bits)
- Code d'erreur (8 bits)
- Checksum (16 bits)
- Information supplémentaires (le cas échéant, 32 bits incluant le padding)
- Données : copie de l'en-tête IPv4 + les 8 premiers octets de données du paquet IPv4 ayant provoqué le message d'erreur

Message ICMP

Pourquoi on copie le contenu du paquet original dans les données ICMP ?

Ces informations permettent à l'hôte d'associer le message d'erreur au processus concerné.

Dans un ping (ICMP echo), les données contiennent généralement une séquence de caractère ASCII.

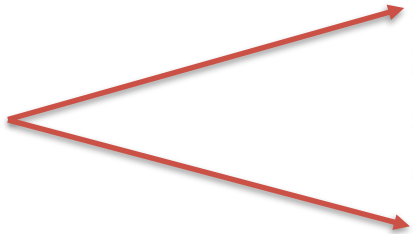
Type message

Summary of Message Types

0	Echo Reply
3	Destination Unreachable
4	Source Quench
5	Redirect
8	Echo
11	Time Exceeded
12	Parameter Problem
13	Timestamp
14	Timestamp Reply
15	Information Request
16	Information Reply

Type message

ping



Summary of Message Types

0	Echo Reply
3	Destination Unreachable
4	Source Quench
5	Redirect
8	Echo
11	Time Exceeded
12	Parameter Problem
13	Timestamp
14	Timestamp Reply
15	Information Request
16	Information Reply

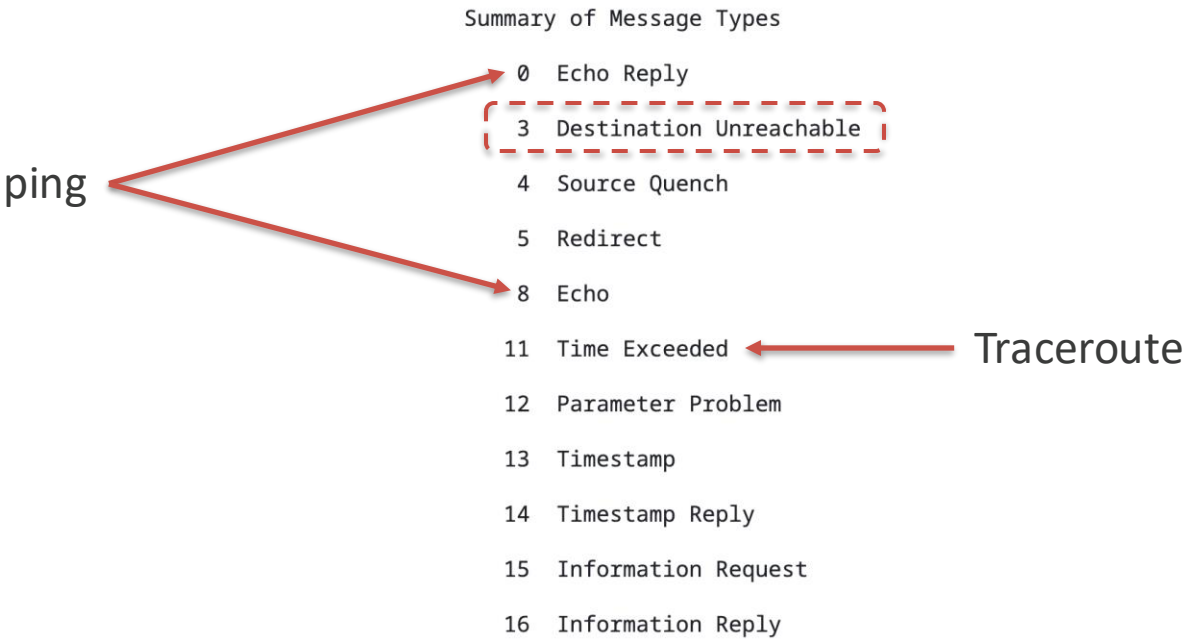
Type message

ping

Summary of Message Types

0	Echo Reply
3	Destination Unreachable
4	Source Quench
5	Redirect
8	Echo
11	Time Exceeded
12	Parameter Problem
13	Timestamp
14	Timestamp Reply
15	Information Request
16	Information Reply

Type message



UVSQ

université PARIS-SACLAY

ARP
(Address Resolution Protocol)

ARP

Lorsqu'on veut accéder à une ressource externe, on peut obtenir son adresse IP grâce au service DNS.

Au moment d'envoyer un paquet, on constate qu'on ne dispose que de l'adresse IP, et non de l'adresse MAC, ce qui empêche l'encapsulation complète de la trame.

Il nous faut donc un mécanisme permettant d'**obtenir l'adresse MAC correspondant à une adresse IP**.

ARP (Address Resolution Protocol) : adresse IP -> adresse MAC

RFC 826

ARP dynamique & statique

ARP dynamique

- Apprises automatiquement avec les échanges (requête/réponse) ARP
- Avec un temps d'expiration (TTL)

ARP statique

- Configurés par l'administrateur
- N'expirent pas

Fonctionnement ARP

Cache ARP : stocke temporairement les correspondances entre adresses IP et adresses MAC

Si on trouve l'entrée dans la cache, on l'encapsule directement dans la trame;

Sinon, on doit chercher l'adresse MAC avec l'ARP.

Fonctionnement ARP

Cache ARP : stocke temporairement les correspondances entre adresses IP et adresses MAC

Si on trouve l'entrée dans la cache, on l'encapsule directement dans la trame;

Sinon, on doit chercher l'adresse MAC avec l'ARP.

Les requêtes ARP sont envoyées en broadcast;
Les réponses ARP sont envoyées en unicast.

Scénarios ARP

- Si une **machine** va envoyer la trame à une autre machine du **même réseau**, elle peut utiliser ARP pour trouver directement **l'adresse MAC de la machine de destination**.
- Si une **machine** va envoyer la trame à une machine située dans **un autre réseau**, elle va utiliser ARP pour trouver **l'adresse MAC du routeur local**, qui se chargera ensuite de l'acheminement vers le réseau suivant.
- Si un **routeur** va transférer un paquet vers une adresse située dans **un réseau directement connecté**, il peut obtenir **l'adresse MAC de la destination** avec l'ARP.
- Si un **routeur** doit transférer un paquer vers **un autre réseau**, il utilise ARP pour obtenir **l'adresse MAC de l'interface du routeur de prochain saut**.

Conclusion ARP

- ARP ne cherche pas forcément l'adresse MAC de la destination finale. Il peut chercher l'adresse MAC de l'interface du routeur local ou l'adresse MAC de l'interface du prochain saut (next hop).
- Lors de l'encapsulation, **les adresses IP source et destination ne changent pas**, mais **les adresses MAC source et destination sont mises à jour à chaque passage par un routeur**.
- ARP (broadcast) ne traverse jamais un routeur.
- ARP ne concerne que IPv4. En IPv6, ARP est remplacé par NDP (Neighbor Discovery Protocol).

Résumé (Cours 7)

- EtherChannel
- VLSM
- DHCP
- DNS
- ICMP
- ARP