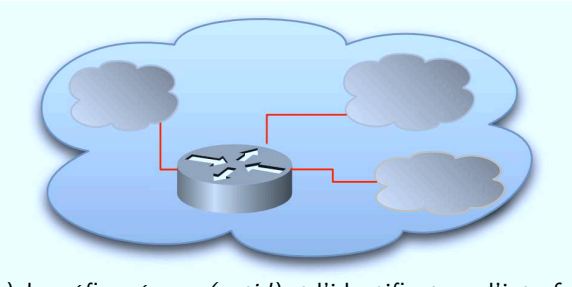
ARes - Lab n°6

Couche Réseau IPv4 (1) : Adressage et contrôle

Dans ce lab, nous nous intéressons à la couche réseau dans l'environnement TCP/IP. Lors des séances précédentes, nous avons observé de nombreux paquets IP dans les traces. Nous allons ici les analyser en détail. Après quelques exercices d'adressages, nous étudierons le découpage en sous-réseau et ferons évoluer la plateforme pour y intégrer des routeurs. Nous nous attacherons alors à analyser les caractéristiques et mécanismes spécifiques à la couche 3 à travers deux outils fondamentaux pour la supervision des réseaux TCP/IP : `ping` et `traceroute`.



1 Exercices d'adressage (sans machine)

1. Pour les adresses IPv4 suivantes, précisez le masque réseau *(netmask)*, le préfixe réseau *(netid)* et l'identificateur d'interface *(hostid)* : `192.33.182.182 /24` ; `81.217.9.35 /20` ; `192.19.67.59 /22` ; `203.19.40.199 /26`.
2. Pour chacun des réseaux suivants, indiquez les adresses IPv4 de la première machine, de la dernière et celle de diffusion *(broadcast)* : `192.33.182.0 /24` ; `10.0.0.0 /16` ; `81.188.160.128 /26` ; `81.188.160.0 /19`.
3. Je dispose du bloc CIDR `150.44.0.0/16` pour mon entreprise. J'ai au plus 1000 machines par sous-réseau. Quel découpage en sous-réseaux dois-je utiliser pour maximiser le nombre de sous-réseaux. Pour le 1^r, le 2^{nd} et le dernier sous-réseau, indiquez le préfixe, les adresses de la première machine, de la dernière et de diffusion.
4. Quelle est la perte d'adresses liée au découpage en sous-réseaux (en nombre d'adresses utilisables pour des interfaces) ?
5. J'ai 20 réseaux de 50 machines à adresser. Quel taille de bloc CIDR vais-je demander à mon fournisseur de service ? Supposons que l'on m'attribue le premier bloc de taille adéquat du préfixe `60.44.32.0 /20`. Pour le 1^r, le 2^{nd} et le dernier sous-réseau, indiquez le préfixe, les adresses de la première machine, de la dernière et de diffusion.
6. Dans l'exercice précédent, si l'on indique directement le nombre de machines que l'on souhaite adresser à son fournisseur, celui-ci nous donnera-t-il un bloc CIDR de la taille choisie ci-dessus ?

2 Plan d'adressage et découpage en sous-réseau (sans machine)

Une entreprise souhaite intégrer son réseau dans l'environnement TCP/IP. Elle possède un site central avec 6 réseaux de 50 machines maximum. Elle souhaite aussi intégrer ses trois succursales, chacune avec 20 machines maximum.

1. Dans le cadre d'un adressage basé sur CIDR, quelle plan proposez vous pour le site principal ? Illustrez votre solution par un schéma où vous utiliserez le premier bloc d'adresse adapté du préfixe `88.5.100.0`.
2. Nous intégrons les 3 succursales. Sachant qu'elles sont reliées chacune par une liaison spécialisée entre leur routeur et celui du site principal, étendez votre solution et complétez le schéma.
3. Quelle sera la table de routage du routeur central ? D'un routeur de succursale ? D'une machine ?
4. Par souci d'économie, l'entreprise décide d'abandonner les lignes spécialisés et connecte directement les succursales à l'Internet. Quelles modifications cela introduit-il au niveau de l'adressage, du routage et de la sécurité ?
5. Finalement, l'entreprise opte pour un adressage privé et des VPN. Quelle est sa motivation ? Quelles modifications cela introduit-il au niveau de l'adressage, du routage, de la sécurité et de l'accès à Internet ?

3 Mise en œuvre de sous-réseaux sur la plateforme

3.1 Généralités sur les équipements CISCO

CISCO utilise dans la plupart de ses équipements le logiciel IOS *(Internetwork Operating System)*. Celui-ci réalise l'intégration des fonctions de configuration d'interface, de commutation et de routage grâce à un système d'exploitation propriétaire de conception relativement ancienne (multitâche monolithique non-préemptif). Des versions plus sophistiquées existent pour les matériels haut de gamme. L'interface est textuelle et basée sur des lignes de commandes *(CLI, Command Line Interface)*, similaire aux *shells* UNIX, même si elle n'atteint pas la souplesse d'utilisation de ces derniers. Un ensemble déterminé de commandes avec de multiples paramètres est proposé en fonction du mode et du niveau de privilège de l'utilisateur (toutes les commandes ont un niveau de privilège, de 0 à 15, et ne pourront être exécutées que par un utilisateur avec les privilèges suffisants).

3.1.1 Les différents modes d'accès IOS

Sur les équipements CISCO, il existe donc plusieurs modes dans lesquels un utilisateur peut se trouver. Nous en utiliserons principalement deux :

user EXEC (level 1) mode initial dans lequel l'utilisateur se retrouve par défaut à la connexion de la machine. Les commandes disponibles sont très restreintes (pas de configuration). Le *prompt* est de la forme "`nom_equipement>`". Pour sortir de ce mode, utilisez la commande `logout` ou la commande `quit`.

privilege EXEC (level 15) accès via la commande `enable`. Un mot de passe est demandé pour accéder à ce mode des utilisateurs avec privilèges (similaire au compte `root` sous UNIX). De nombreuses commandes deviennent accessibles dont celle pour passer en sous-mode de configuration. Le *prompt* est de la forme "`nom_equipement#`". Pour sortir de ce mode, utilisez la commande `disable`.

global configuration accès via la commande `configure` depuis le mode *privilege EXEC*. Dans le niveau avec privilèges, ce sous-mode permet de configurer des fonctionnalités ayant une portée générale sur l'équipement. Le *prompt* est de la forme "`nom_equipement(config)#`". Pour sortir de ce sous-mode, utilisez la commande `exit`, la commande `end` ou appuyez sur les touches **ctrl**+**Z**.

dedicated configuration accès via une commande du type `interface`, `vlan`, `router`... depuis le sous-mode *global configuration*. Toujours dans le niveau avec privilèges, ce sous-sous-mode permet de configurer des paramètres spécifique à une interface, à un VLAN, au routage... Le *prompt* est de la forme "`nom_equipement(config-xxx)#`", ou `xxx` dépend de l'élément à configurer. Pour sortir de ce sous-sous-mode et retourner vers le sous-mode de *global configuration*, utilisez la commande `exit`. Pour retourner directement vers le mode *privilege EXEC*, utilisez la commande `end` ou appuyez sur les touches **ctrl**+**Z**.

Voici l'enchaînement des différents modes cités ci-dessus :

user EXEC ⇔ **privilege EXEC** ⇔ **global configuration** ⇔ **dedicated configuration**

3.1.2 Accès au commutateur et au routeur de la plateforme

Les commutateurs et routeurs présents dans chacune des deux baies sont directement accessibles en TELNET via :

- le réseau d'administration (VLAN 200, adresse IPv4 `10.0./16`) à partir du PC de l'ARI sur lequel vous êtes connecté ;
- les réseaux d'expérimentation (VLAN **N**1, VLAN **N**2...) à partir de vos VM de la plateforme.

Les adresses IPv4 avec lesquelles vous pouvez vous connecter aux équipements de la plateforme via le réseau d'administration sont celles représentées sur la FIGURE 1. Ainsi, le poste ARI **N** peut accéder directement à 5 éléments de la plateforme :

- le commutateur **N**, adresse IPv4 `10.0.2.`**N**
- le routeur **N**, adresse IPv4 `10.0.3.`**N**
- les VM "client" (**N1**), "sonde" **N2** et "serveur" (**N3**) aux adresses habituelles.

Remarque : il va vous être demandé un mot de passe, celui-ci vous sera fourni par votre encadrant.

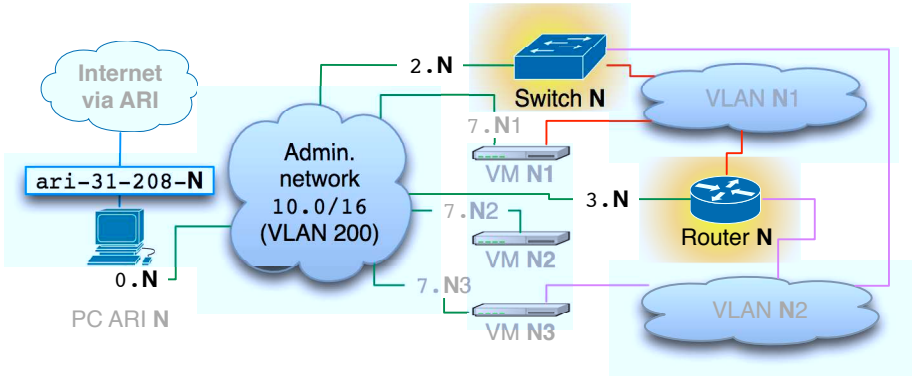


FIGURE 1 : Adressage d'administration des équipements de la plateforme

3.1.3 Nom des interfaces des équipements CISCO sur la plateforme

Les interfaces des commutateurs et des routeurs de la plateforme sont de type 100BaseTX *(Fast Ethernet)*. Le nom des interfaces indiquent explicitement leur type. Ici, nous utiliserons le préfixe "`FastEthernet`" (ou "`fa`" en abrégé). Deux numéros séparés par '/' suivent pour différencier les interfaces de même type. Les équipements pouvant intégrer plusieurs cartes d'extension, le premier numéro indique la carte et le second le numéro d'interface sur la carte (sur la plateforme, il n'y a pas de carte supplémentaire, le premier numéro a donc toujours la valeur "0"). Suite à l'intégration de plusieurs lignes de produits, CISCO numérote de manière différente les interfaces sur les commutateurs et sur les routeurs :

- A partir de 1 sur les commutateurs (donc de 1 à 12 sur la plateforme). Exemple : "`fa0/3`"
- A partir de 0 sur les routeurs (donc 0 et 1 sur la plateforme). Exemple : "`fa0/0`" ; "`fa0/1`"

3.1.4 Remarques générales pour les commandes IOS

La TABLE 1 présente quelques commandes générales qui sont valables pour les commutateurs et routeurs CISCO.

TABLE 1 : Commandes IOS générales

Action	Commandes	Mode
pinger une adresse IP	`ping <adresse-ip>`	user
passer en mode avec privilèges	`enable`	user
afficher la configuration actuelle	`show running-config`	privilege
afficher les interfaces	`show interface status`	privilege
afficher une interface	`show interface fa0/<num-if>`	privilege
afficher les utilisateurs connectés	`show users`	privilege
passer en mode configuration	`configure terminal`	privilege
copier la configuration sur un PC via TFTP	`copy running-config tftp://<adressePC>/<nom-cf>`	privilege

Négation : une commande ou sa réciproque ne sont pas forcément disponibles. Par exemple, il existe la commande "`shutdown`" pour éteindre une interface mais pas celle pour allumer une interface. Il suffit simplement d'utiliser l'opérateur "`no`", qui réalise la négation d'une commande. Ainsi, "`no shutdown`" indique le fonctionnement d'une interface.

Aide : les deux principales aides qui sont fournies par l'IOS sont les touches **tabulation** et **?**. La première assure la complétion automatique et la seconde permet d'afficher tous les choix de commandes ou paramètres disponibles à tout moment, ainsi qu'une petite phrase explicative.

Par exemple, **?** tapé en début de ligne liste toutes les commandes disponibles dans le mode actuel, et **?** tapé après la commande `show` listera tous les paramètres pouvant suivre cette commande.

3.2 Configuration du commutateur

Pour réaliser la topologie de ce sujet, nous allons utiliser deux réseaux d'expérimentation. Dans le cadre de la plateforme, nous ne disposons que d'un commutateur, nous allons donc mettre en place ces réseaux à l'aide de VLAN.

Les VLAN sont des subdivisions virtuelles d'un réseau local commuté (subdivision du réseau Ethernet associé au commutateur dans le cas de la plateforme). La discrimination la plus simple des trafics est généralement réalisée grâce à une attribution de certaines interfaces à certains VLAN (appelés VLAN statiques). Les VLAN ont le même comportement qu'un réseau local physique. Les trafics point-à-point *(unicast)* et de diffusion *(multicast, broadcast)* sont transmis seulement aux interfaces du VLAN concernées. Les machines connectées à deux VLAN différents ne peuvent communiquer entre elles (un routeur est alors nécessaire).

Les commutateurs de la plateforme sont des CISCO Catalyst 2950-12 avec logiciel standard. Ils supportent 64 VLAN. Les VLAN sont identifiés par un numéro compris entre 1 et 1005. Par défaut toutes les interfaces appartiennent au VLAN 1. Deux modes de fonctionnement relatifs aux VLAN sont possibles pour les interfaces :

mode "access" interface appartenant à un seul VLAN et faisant seulement transiter le trafic de celui-ci ;

mode "trunk" interface faisant transiter le trafic de l'agrégat des VLAN (le *trunk*, comprenant par défaut tous les VLAN).

3.2.1 Commandes de base du commutateur

Les principales commandes dont vous aurez besoin pour configurer le commutateur afin de réaliser la topologie de ce lab sont présentées dans la TABLE 2.

TABLE 2 : Commandes du commutateur CISCO

Action	Commandes	Mode
afficher la configuration de tous les VLAN	`show vlan`	privilege
afficher le résumé des VLAN	`show vlan brief`	privilege
créer un VLAN	`vlan <num-vlan>`	global conf.
nommer un VLAN	`name <nom-vlan>`	vlan conf.
configurer un VLAN	`interface vlan <num-vlan>`	global conf.
associer une adresse IP à un VLAN	`ip address <adresse-ip> <masque>`	interf. conf.
supprimer un VLAN	`no vlan <num-vlan>`	global conf.
configurer une interface	`interface fa0/<num-if>`	global conf.
configurer plusieurs interfaces	`interface range fa0/<nif1> - <nif2>, 0/<nif3>`	global conf.
associer une interface à un VLAN en mode "access"	`switchport access vlan <num-vlan>`	interf. conf.
associer une interface à l'agrégat des VLAN (mode "trunk")	`switchport mode trunk`	interf. conf.
limiter l'agrégat en retirant/rajoutant des VLAN (mode "trunk")	`switchport trunk allowed vlan remove <num-vlan>` ou `switchport trunk allowed vlan add <num-vlan>`	interf. conf.
afficher la configuration d'une interface	`show interfaces fa0/<num-if> switchport`	privilege
afficher la configuration de l'interface relatives à l'agrégat (mode "trunk")	`show interfaces fa0/<num-if> trunk`	privilege

3.2.2 Interfaces des commutateurs de la plateforme

Les commutateurs de la plateforme disposent de 12 interfaces Ethernet. Le câblage associé à chacune de ces interfaces est fixé. Voir la TABLE 3 pour la numérotation des interfaces (les équipements reliés à chacune des interfaces sont indiqués pour le commutateur **N**, associé au PC ARI **N**).

1. A partir du PC ARI **N**, connectez-vous sur le commutateur correspondant via telnet (`telnet 10.0.2.N`).
2. Visualiser la configuration du commutateur, en particulier détaillez les paramètres des interfaces et des VLAN (ces quelques VLAN peuvent être déjà créés avec les numéros **N1**, **N2**, **200** et les noms respectifs `VLAN_A`, `VLAN_B`, `VLAN_ADMIN`).

TABLE 3 : Interfaces du CISCO Catalyst 2950-12

Numéro d'interface	Nom IOS	Remarque
1	`fa0/1`	reliée à l'interface `fa0/0` du routeur **N**
2	`fa0/2`	reliée à l'interface `fa0/1` du routeur **N**
3	`fa0/3`	reliée à l'interface `eth1` de la VM "client" **N1**
4	`fa0/4`	reliée à l'interface `eth1` de la VM "sonde" **N2**
5	`fa0/5`	reliée à l'interface `eth1` de la VM "serveur" **N3**
6	`fa0/6`	non utilisée dans ce lab
7	`fa0/7`	non utilisée dans ce lab
8	`fa0/8`	non utilisée dans ce lab
9	`fa0/9`	non utilisée dans ce lab
10	`fa0/10`	non utilisée dans ce lab
11	`fa0/11`	reliée au commutateur central de la baie
12	`fa0/12`	reliée au PC ARI **N**

3.2.3 Recopie d'interface (SPAN)

Un commutateur relaye généralement les trames directement d'un port d'entrée vers un port de sortie. L'interception de trafic à partir d'un autre port n'est pas réalisable comme sur un simple répéteur multiport *(hub)*. La réalisation de captures à partir d'une machine sonde différente du client ou du serveur, c'est-à-dire à partir d'une interface différente de celle du client ou du serveur, nécessite l'utilisation d'une fonction de recopie explicite du trafic vers l'interface de la sonde.

La recopie d'interface, également appelé *port mirroring*, *port monitoring* ou SPAN *(Switched Port Analyzer* chez CISCO) permet de répliquer des trames. Son principe est le suivant : toutes les trames arrivant sur une interface sont dupliquées vers une autre interface spécifiées lors de la configuration. Dans le cadre de la plateforme, nous réalisons la recopie des trames passant par l'une des interfaces reliées aux machines client ou serveur vers l'interface reliée à la machine sonde. La TABLE 4 présente quelques commandes IOS relatives à cette technique.

TABLE 4 : Commandes pour la recopie d'interface (SPAN)

Action	Commandes	Mode
spécifier les interfaces source	`monitor session 1 source interface fa0/<num-if> [, | -]`	global conf.
spécifier l'interface destination	`monitor session 1 destination interface fa0/<num-if>`	global conf.
afficher les recopies d'interfaces	`show monitor`	global conf.
supprimer les recopies d'interfaces	`no monitor session all`	global conf.

Pour mettre en œuvre la recopie d'interface sur la plateforme, une **session SPAN** est utilisée. Celle-ci correspond à l'association d'une interface destination avec plusieurs interfaces sources (les commutateurs plus sophistiqués que le CISCO Catalyst 2950-12 peuvent également utiliser des VLAN sources mais nous ne décrirons dans le présent document que les fonctionnalités relativement limitées des commutateurs de la plateforme).

Une **interface source** *(source port)*, également appelée interface supervisée *(monitored port)*, est une interface commutée dont on souhaite analyser le trafic. Dans une unique session, le trafic reçu (`Rx`), le trafic émis (`Tx`) ou le trafic bidirectionnel (`Both`) peuvent être supervisés. Par défaut, le trafic bidirectionnel des interfaces source est supervisé.

Toute session SPAN locale doit être associée à une **interface destination** *(destination port)*, également appelée interface de supervision *(monitoring port)*. Par défaut, le trafic qui y est recopié est identique à celui des interfaces sources supervisées. Nous n'utiliserons pas d'encapsulation particulière pour le trafic dupliqué *(no encapsulation type header)*.

1. Visualisez la configuration de la recopie d'interfaces du commutateur et justifiez la.

2. Quelles commandes IOS ont permises de mettre en place cette session de recopie d'interfaces ?

3.2.4 Création de 2 VLAN

Avec la première topologie de la plateforme (utilisée lors des labs 2 à 5), nous avons fait transiter tout le trafic entre les machines client et serveur de la plateforme via le commutateur. Pour cela, les interfaces reliant les VM client et serveur au commutateur sont placées au sein du même VLAN (voir la FIGURE 2).

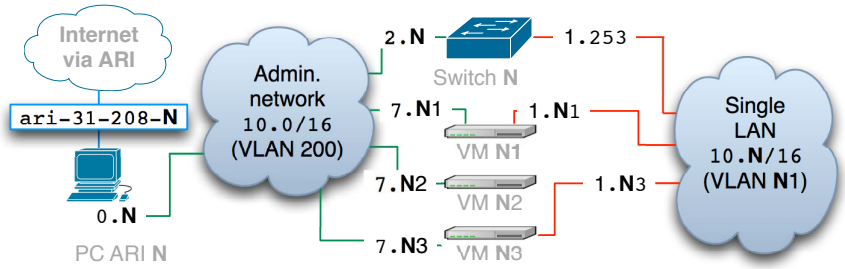


FIGURE 2 : Topologie virtuelle 1 (1 LAN)

1. Quelles commandes IOS ont permis de mettre en place le VLAN **N**1 ?

Dans la première topologie, les interfaces des VM "client" et "serveur" étant sur le même VLAN, il n'y avait pas besoin de routage et donc pas de configuration de routeur. Nous souhaitons mettre en place la **seconde topologie**, utilisant deux VLAN afin de séparer ces VM. Le but de cette modification est d'étudier le trafic inter-réseau, et en particulier l'intégration des routeurs (voir la FIGURE 3).

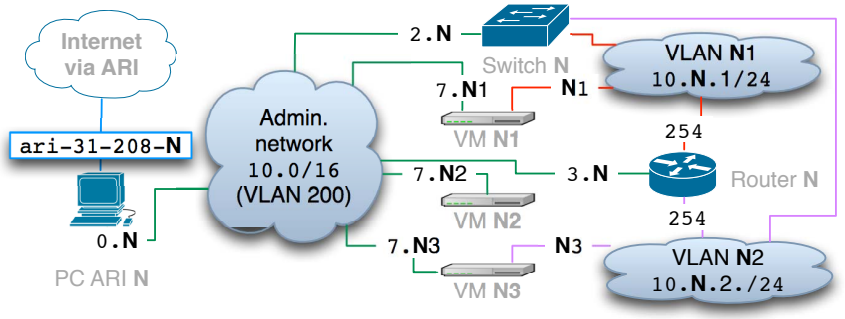


FIGURE 3 : Topologie virtuelle 2 (2 LAN et 1 routeur)

Nous allons conserver le VLAN **N**1 pour le client et en introduire un nouveau, le VLAN **N**2 pour le serveur :

2. Quelles commandes IOS permettent de mettre en place le nouveau VLAN ? Mettez en place cette nouvelle configuration (si besoin).
3. Vérifiez maintenant que les VM "client" et "serveur" ne peuvent plus communiquer directement.
4. Pouvez-vous toujours capturer le trafic correspondant à l'aide de l'outils `wireshark` sur la VM sonde ?
5. Analysez le trafic généré alors que vous testez la connectivité entre les VM "client" et "serveur".

3.3 Configuration du routeur

Dans la topologie 2, pour faire communiquer la VM "client" et la VM "serveur", il nous faut ajouter un routeur car les deux VM sont maintenant dans des réseaux différents. Les routeurs CISCO 2800 de la plateforme disposent de deux interfaces Ethernet. La première est utilisée pour accéder au VLAN d'administration et la seconde est virtualité pour accéder à plusieurs VLAN simultanément (VLAN **N**1 et VLAN **N**2).

3.3.1 Commandes de base du routeur

Les principales commandes dont vous aurez besoin pour configurer le routeur dans le cadre de la topologie 2 sont présentées dans la TABLE 5.

TABLE 5 : Commandes du routeur CISCO

Action	Commandes	Mode
afficher le résumé des interfaces	`show interfaces brief`	privilege
configurer une interface	`interface fa0/<num-if>`	global conf.
configurer une interface virtuelle	`interface fa0/<num-if>.<num-sub>`	global conf.
associer une adresse IPv4 à une interface	`ip address <adresse-ip> <masque>`	interf. conf.
ne pas associer d'adresse IP à une interface	`no ip address`	interf. conf.
supprimer le proxy ARP d'une interface	`no ip proxy-arp`	interf. conf.
sélection du VLAN avec une interface en mode "trunk"	`encapsulation dot1Q <vlan-num>`	interf. conf.
afficher la table de routage	`show ip route`	privilege

1. Quelle configuration doit-on mettre en place au niveau du commutateur pour intégrer le routeur ?
2. Quelle configuration doit-on mettre en place au niveau du routeur ? Quelles commandes IOS permettent de réaliser celle-ci ?
3. Quel protocole de routage faut-il démarrer pour construire la table de routage ? Comment afficher cette dernière ?
4. A partir du PC ARI **N**, connectez-vous sur le routeur correspondant via telnet (`telnet 10.0.3.`**N**).
5. Visualisez et analysez la configuration du routeur.

3.3.2 Configuration avec interfaces virtuelles

Les **interfaces virtuelles** sont des interfaces de réseau non associées directement à un élément physique (par exemple un connecteur RJ45). Les interfaces virtuelles existent seulement au niveau du logiciel.

Les **sous-interfaces** sont associées à des interfaces physiques. Elles ne fonctionnent que lorsque l'interface physique correspondante fonctionne également. Les sous-interfaces sont crées en subdivisant l'interfaces physique en 2 ou plus interfaces virtuelles auxquelles on peut associer des adresses IP uniques. Les sous-interfaces sont identifiées par le préfixe de l'interface physique auxquelles elles sont associées suivit d'un point puis d'un entier unique pour ce préfixe. Par exemple, la première sous-interface de `fa0/1` sera nommée `fa0/1.1`.

La configuration du routeur montre l'utilisation de sous-interfaces (`fa0/1.1` et `fa0/1.2`). Ces interfaces virtuelles sont nécessaires car la topologie 2 (voir la FIGURE 3) utilise 3 interfaces alors que le routeur n'en dispose physiquement que de 2.

Le routeur est directement connecté aux réseaux `10.0/16`, `10.`**N**`.1/24` et `10.`**N**`.2/24`. L'interface (`fa0/1`) est virtualisée en la déclarant fonctionnelle mais sans adresse IP (commande `no ip address`). Les interfaces virtuelles `fa0/1.1` et `fa0/1.2` sont déclarées de manière similaire à une interface physique.

L'interface physique du routeur étant reliée à l'interface en mode "trunk" du commutateur, tous les VLAN y sont disponibles. Il ne reste qu'à associer chacune de ces interfaces virtuelles à un VLAN (commande `encapsulation dot1Q` avec le numéro du VLAN correspondant).

3.3.3 Test de la communication entre les 2 VLAN

Après avoir vérifié les paramètres des interfaces (`fa0/1.1` et `fa0/1.2`), veillez à supprimer la fonctionnalité de *proxy ARP* configurée par défaut sur chaque interface d'expérimentation du routeur. Cette fonctionnalité autorise le routeur à répondre à des requêtes ARP de machine mal configurées. Elle pourrait interférer avec les diagnostics que nous réalisons :

```
routeurN# configure terminal
routeurN(config)# interface fa0/1.1
routeurN(config-if)# no ip proxy-arp
routeurN(config)# interface fa0/1.2
routeurN(config-if)# no ip proxy-arp
routeurN(config)# end
```

Une fois la configuration du routeur adéquat, le routage entre les deux sous-réseaux est opérationnel.

1. Vérifiez la connectivité à partir du routeur (en utilisant commande `ping` vers le client et vers le serveur).
2. Vérifiez la connectivité vers le routeur à partir des VM "client" et "serveur" (en utilisant toujours la commande `ping`).
3. Les VM "client" et "serveur" peuvent-elles à nouveau communiquer entre elles ?

3.3.4 Reconfiguration des VM de la plateforme

La modification des paramètres de réseau implique la modification de la configuration de tous les équipement directement connectés à ces réseaux. C'est le cas pour les VM "client" et "serveur". Les commandes utilisées sous UNIX sont les suivantes : `ifconfig` et `route`.

Voici le début de la description du `man` UNIX sur la commande `ifconfig` :

> `Ifconfig is used to configure the kernel-resident network interfaces. It is used at boot time or when system tuning is needed. If no arguments are given, ifconfig displays the status of the currently active interfaces. Otherwise, it configures an interface.`

Et voici celui de la commande `route` :

> `Route manipulates the kernel?s IP routing tables. Its primary use is to set up static routes to specific hosts or networks via an interface after it has been configured with the ifconfig(8) program. When the add or del options are used, route modifies the routing tables. Without these options, route displays the current contents of the routing tables.`

Lorsqu'elles sont utilisées pour modifier la configuration réseau de la machine, ces commandes nécessitent des privilèges d'administrateur et donc l'utilisation de la commande `su`.

1. Quelle analyse à l'aide de `wireshark` pouvez-vous faire de la situation ?
2. Vérifiez votre diagnostique de la panne à l'aide de la commande `/sbin/ifconfig` sur les VM (sur les interfaces `eth1` sur le client et sur le serveur).
3. Une modification du masque des interfaces des VM concernés est donc nécessaire. Utilisez la commande suivante : `/sbin/ifconfig eth<n> netmask <nouveau-masque>`.
4. Les VM "client" et "serveur" peuvent-elles à nouveau communiquer entre elles ?
5. Une analyse à l'aide de `wireshark` peut-elle vous aider à diagnostiquer la situation ?
6. Une indication de la passerelles pour accéder aux réseaux non directement connectés est nécessaire. Utilisez la commande suivante :
   `/sbin/route add default gw <adresse-passerelle>`.
7. Les VM "client" et "serveur" peuvent-ils enfin communiquer entre eux ?

3.4 Avant de quitter la salle...

A la fin de la séance (**pas tout de suite !** vous avez encore besoin de la configuration réalisée), quelque soit votre état d'avancement dans le présent sujet du lab, **si votre encadrant vous le demande**, n'oubliez pas remettre les machines de la plateforme dans leur état initial (celui dans lequel vous les avez trouvé) :

- supprimer le VLAN **N2** du **commutateur** (`telnet 10.0.2.N`) :

```
swN#show vlan brief
swN#configure terminal
swN(config)#interface range fa0/3, fa0/5
swN(config-if-range)#switchport access vlan N1
swN(config-if-range)#exit
swN(config)#no vlan N2
swN(config)#end
swN#show vlan brief
```

- supprimer les potentielles modifications apportées à la configuration du **routeur** (`telnet 10.0.254.N`).
- reconfigurer **chacune des VM client et serveur**...
  - VM "client" (`telnet 10.0.7.N1`) :

```
client@vmN1:~$ su
root@vmN1:~$ /sbin/ifconfig eth0 netmask 255.255.0.0
root@vmN1:~$ /sbin/route del default
```

  - VM "serveur" (`telnet 10.0.7.N3`) :

```
client@vmN3:~$ su
root@vmN3:~$ /sbin/ifconfig eth2 netmask 255.255.0.0
toot@vmN3:~$ /sbin/route del default
```

4 Messages de contrôle ICMP

4.1 Etude des l'outil `ping`

Voici le début de la description du `man` UNIX sur la commande `ping` :

> `Ping uses the ICMP protocol's mandatory ECHO_REQUEST datagram to elicit an ICMP ECHO_RESPONSE from a host or gateway. ECHO_REQUEST datagrams (''pings'') have an IP and ICMP header, followed by a ''struct timeval'' and then an arbitrary number of ''pad'' bytes used to fill out the packet.`

4.2 Test d'une machine distante

La commande `ping` permet de tester la connectivité vers une machine distante, et — en envoyant plusieurs paquets à la suite — d'effectuer des statistiques sur les caractéristiques du chemin suivi (RTT, taux de perte, variabilité des résultats en fonction de la taille des datagrammes émis...). Voici un exemple d'utilisation :

```
pirogue:~# ping sphinx.lip6.fr
PING sphinx.lip6.fr (132.227.74.253): 56 data bytes
64 bytes from 132.227.74.253: icmp_seq=0 ttl=247 time=5.5 ms
64 bytes from 132.227.74.253: icmp_seq=1 ttl=247 time=9.3 ms
64 bytes from 132.227.74.253: icmp_seq=2 ttl=247 time=8.0 ms
64 bytes from 132.227.74.253: icmp_seq=3 ttl=247 time=6.3 ms
64 bytes from 132.227.74.253: icmp_seq=4 ttl=247 time=4.8 ms
64 bytes from 132.227.74.253: icmp_seq=5 ttl=247 time=7.6 ms
64 bytes from 132.227.74.253: icmp_seq=6 ttl=247 time=5.8 ms
```

```
--- sphinx.lip6.fr ping statistics ---
7 packets transmitted, 7 packets received, 0% packet loss
round-trip min/avg/max = 4.8/6.7/9.3 ms
pirogue:~#
```

1. Pour vous échauffer, analysez **manuellement** (sans `wireshark`) la première trame correspondant à cet échange :

```
0000  00 08 21 59 66 42 00 04  76 21 1b 95 08 00 45 00   ..!YfB.. v!....E.
0010  00 54 64 db 00 00 40 01  df 38 c2 fe a3 b6 84 e3   .TdÛ..@. ß8ÂP£¶.ã
0020  4a fd 08 00 d8 2d 6e 5b  00 00 3f 9f 10 01 00 0d   Jý..Ø-n[ ..?.....
0030  76 c6 08 09 0a 0b 0c 0d  0e 0f 10 11 12 13 14 15   vÆ...... ........
0040  16 17 18 19 1a 1b 1c 1d  1e 1f 20 21 22 23 24 25   ........ .. !"#$%
0050  26 27 28 29 2a 2b 2c 2d  2e 2f 30 31 32 33 34 35   &'()*+,- ./012345
0060  36 37                                             67
```

2. A votre avis, à quoi correspondent les données transportées par le message ICMP analysé précédement ? Utilisez le `man` UNIX pour avoir des informations complémentaires sur `ping`.
3. Réalisez sur la plateforme la capture d'un `ping` entre le client et le serveur à l'aide du logiciel `wireshark` (ou chargez le fichier suivant : `/Infos/lmd/2014/master/ue/ares-2014oct/tme6-pin.dmp.gz`).
   Quel échange observez-vous dans la trace ?
4. Quelle réponse est retournée à la requête analysée précédemment ?
5. Analysez les champs ICMP de plusieurs paquets échangés. Développez leur signification et inférez leur utilité pour la commande `ping`.

4.3 Test d'une machine proche avec enregistrement du chemin

La commande `ping -R` est une variante de l'exécution illustrée précédement où des options de l'entête IPv4 sont introduites. A vous de les découvrir dans l'exemple suivant :

```
pirogue:~# ping -R rap-jussieu.cssi.renater.fr
PING rap-jussieu.cssi.renater.fr (193.51.182.201): 56 data bytes
64 bytes from 193.51.182.201: icmp_seq=0 ttl=252 time=11.2 ms
RR:     pirogue.l2ti.univ-paris13.fr (194.254.163.182)
        192.168.208.252
        paris13-jussieu.cssi.renater.fr (193.51.182.205)
        jussieu-a1-0-65.cssi.renater.fr (193.51.182.202)
        rap-jussieu.cssi.renater.fr (193.51.182.201)
        jussieu-f3-3.cssi.renater.fr (193.51.182.206)
        192.168.208.254
        gw163.univ-paris13.fr (194.254.163.254)
        pirogue.l2ti.univ-paris13.fr (194.254.163.182)

--- rap-jussieu.cssi.renater.fr ping statistics ---
5 packets transmitted, 1 packets received, 80% packet loss
round-trip min/avg/max = 11.2/11.2/11.2 ms
pirogue:~#
```

1. Réalisez sur la plateforme la capture d'un `ping -R` entre le client et le serveur à l'aide du logiciel `wireshark` (ou chargez le fichier suivant : `/Infos/lmd/2014/master/ue/ares-2014oct/tme6-pRp.dmp.gz`).
   Quelles informations nouvelles découvrez-vous dans ces trames ?
2. Analysez les champs ICMP. Développez leur signification et inférez, avec le contenu de l'option IP, leur utilité pour la commande `ping -R`.
3. Pouvez-vous réaliser un schéma des réseaux traversés ?

4.4 Test d'une machine éloignée avec enregistrement du chemin

La commande `ping -R` utilise le champ d'options limité de l'entête IP. Que se passe-t-il dans l'exemple suivant ?

```
pirogue:~# ping -R sphinx.lip6.fr
PING sphinx.lip6.fr (132.227.74.253): 56 data bytes
64 bytes from 132.227.74.253: icmp_seq=0 ttl=247 time=8.8 ms
RR:     pirogue.l2ti.univ-paris13.fr (194.254.163.182)
        192.168.208.252
        paris13-jussieu.cssi.renater.fr (193.51.182.205)
        jussieu-a1-0-65.cssi.renater.fr (193.51.182.202)
        gw-rap.rap.prd.fr (195.221.126.78)
        rap-jussieu.rap.prd.fr (195.221.127.181)
        r-jusren.reseau.jussieu.fr (134.157.254.126)
        r-olympe.lip6.fr (132.227.109.254)
        132.227.74.254
64 bytes from 132.227.74.253: icmp_seq=1 ttl=247 time=3099.5 ms (same route)
64 bytes from 132.227.74.253: icmp_seq=2 ttl=247 time=2099.7 ms (same route)
64 bytes from 132.227.74.253: icmp_seq=3 ttl=247 time=1099.9 ms (same route)
64 bytes from 132.227.74.253: icmp_seq=4 ttl=247 time=100.1 ms  (same route)
64 bytes from 132.227.74.253: icmp_seq=5 ttl=247 time=8.6 ms    (same route)
64 bytes from 132.227.74.253: icmp_seq=6 ttl=247 time=14.0 ms   (same route)

--- sphinx.lip6.fr ping statistics ---
7 packets transmitted, 7 packets received, 0% packet loss
round-trip min/avg/max = 8.6/918.6/3099.5 ms
pirogue:~#
```

1. Chargez le fichier suivant : `/Infos/lmd/2014/master/ue/ares-2014oct/tme6-pRl.dmp.gz`.
   Quelles informations nouvelles découvrez-vous dans ces trames ?
2. Analysez la route enregistrée dans les paquets IP de retour. Quelles différences observez-vous ?
3. Quelles limitations constatez-vous pour l'approche `ping -R` pour déduire la route suivie par les paquets vers un destinataire donné ?

5 Etude de l'outil `traceroute`

Voici le début de la description du `man` UNIX sur la commande `traceroute` :

> The Internet is a large and complex aggregation of network hardware, connected together by gateways. Tracking the route one's packets follow (or finding the miscreant gateway that's discarding your packets) can be difficult. Traceroute utilizes the IP protocol 'time to live' field and attempts to elicit an ICMP TIME_EXCEEDED response from each gateway along the path to some host.

La commande `traceroute` permet de récupérer les adresses des interfaces des machines intermédaires vers une machine distante. Voici un exemple d'utilisation :

```
pirogue:~# traceroute sphinx.lip6.fr
traceroute to sphinx.lip6.fr (132.227.74.253), 30 hops max, 38 byte packets
 1  gw163.univ-paris13.fr (194.254.163.254)  2.579 ms  0.496 ms  0.415 ms
 2  192.168.208.254 (192.168.208.254)  0.640 ms  0.602 ms  0.216 ms
 3  jussieu-f3-3.cssi.renater.fr (193.51.182.206)  1.813 ms  2.100 ms  1.778 ms
 4  rap-jussieu.cssi.renater.fr (193.51.182.201)  1.986 ms  2.167 ms  2.216 ms
 5  cr-jussieu.rap.prd.fr (195.221.126.77)  2.867 ms  2.799 ms  2.642 ms
 6  jussieu-rap.rap.prd.fr (195.221.127.182)  3.021 ms  3.797 ms  2.315 ms
 7  r-scott.reseau.jussieu.fr (134.157.254.10)  3.692 ms  4.402 ms  5.438 ms
 8  olympe-gw.lip6.fr (132.227.109.1)  4.881 ms !N  3.932 ms !N  3.917 ms !N
pirogue:~#
```

1. Réalisez sur la plateforme un `traceroute` entre le client et le serveur (ou chargez le fichier suivant : `/Infos/lmd/2014/master/ue/ares-2014oct/tme6-tcr.dmp.gz`).
   Analysez la première trame `traceroute` envoyée. Quel est son but ?
2. Quel événement génère la trame suivante ? Quel intérêt peut-on retirer de ce phénomène ?
3. A votre avis, de quelle manière la commande `traceroute` génère-t-elle ses réponses ?
4. Pourquoi les numéros de port UDP destination évoluent-ils ?
5. Finalement, de quelle manière la commande `traceroute` termine-t-elle ?