

Exercice 1 (TD) : modèles, couches, fonctions, unités d'information et identifiants

Dans un schéma d'une pile (empilement de couches) représentant un équipement (ordinateur, switch...), placer :

- les noms des couches OSI : Physique, Liaison, Réseau, Transport, Session, Présentation, Application, "l'humain"
- les noms des couches TCP/IP : Ethernet, Internet, TCP/UDP
- les noms des unités d'information transportées : signal, trame, paquet, segment, message
- les fonctions principales des couches :
 - se charge de l'acheminement d'un signal (numérique ou analogique), de sa régénération, de sa conversion numérique/analogique éventuelle, sans se soucier de la nature de l'information qui est transportée ;
 - se charge de l'accès au médium, de découper un flux de bits en trames, de trouver un destinataire local et de lui acheminer une trame ;
 - traite individuellement chaque paquet d'information d'un message, se charge de trouver le chemin le plus court vers un destinataire présent sur

--	--	--	--	--

Exercice 2 (TD) : vrai ou faux

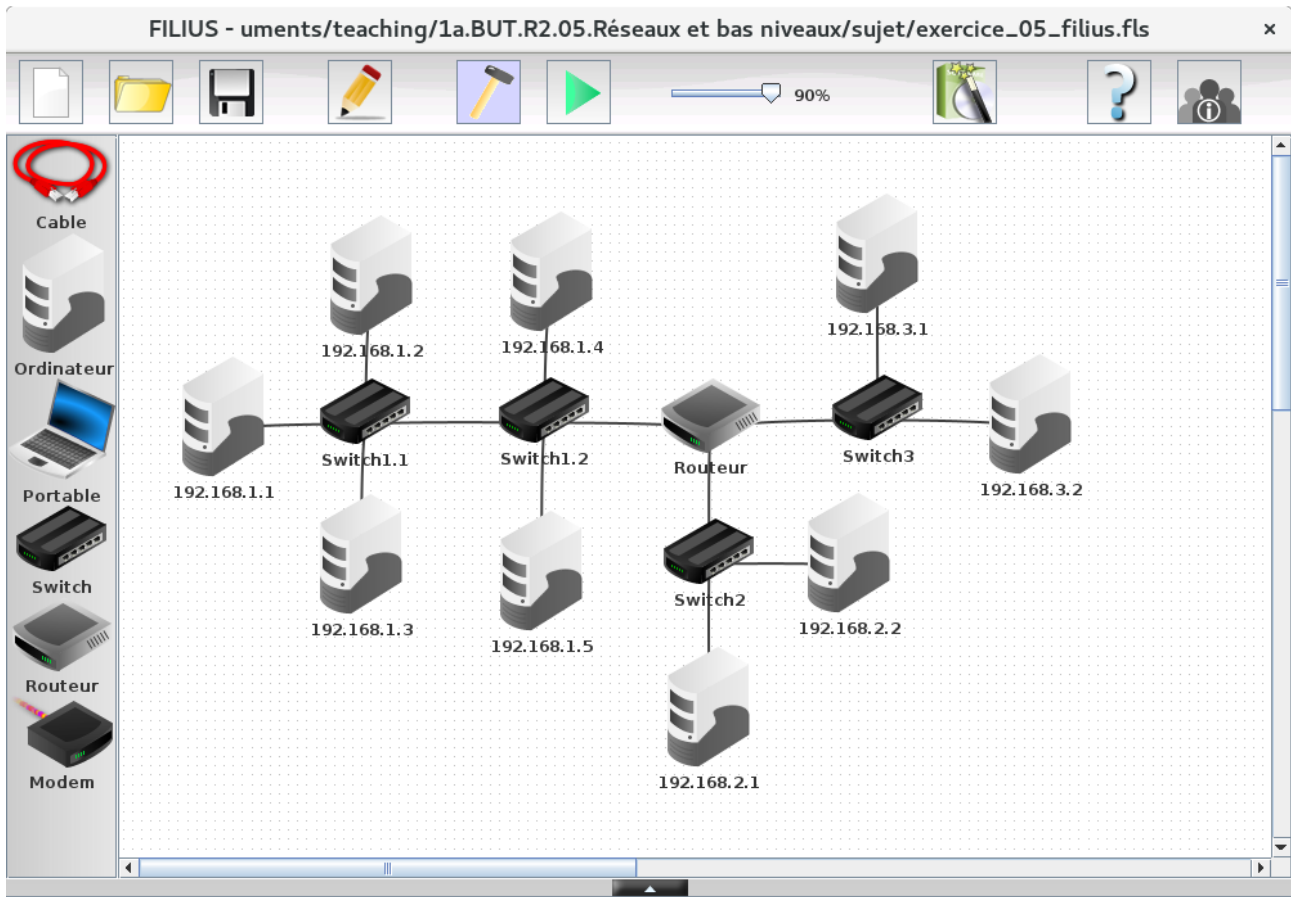
Pour chacune des propositions suivantes, réfléchissez 1 minute sur sa signification, puis votez sur sa véracité et discutez les divergences.

- a) Pour qu'une application cliente puisse communiquer avec une application serveur distante, il faut que l'application cliente dispose de l'adresse IP de l'ordinateur qui héberge le service ainsi que le numéro de port sur lequel écoute le service.
- b) Quand une application cliente communique avec une application serveur distante, la partie cliente n'a pas besoin d'avoir une IP ni un numéro de port.
- c) Une application cliente qui communique avec une application serveur non présente sur le réseau local connaît l'adresse MAC de l'ordinateur qui héberge le service.
- d) Dix navigateurs (application cliente) peuvent tourner sur un même ordinateur en utilisant le même port pour communiquer.
- e) Dix serveurs web peuvent tourner sur le même ordinateur en utilisant le même port pour écouter.

- f) Les informations encapsulées par la couche Transport (e.g. n° port source/destination) ne sont pas utilisées par la couche inférieure adjacente mais par la couche Transport homologue du destinataire.
- g) Les informations encapsulées par la couche Réseau (e.g. @IP source/destination) ne sont pas utilisées par la couche inférieure adjacente mais par la couche Réseau de chaque routeur intermédiaire ainsi que celle de l'ordinateur destinataire.
- h) Le terme "n° de port" utilisé pour désigner les connectiques physiques d'un ordinateur ou d'un équipement d'interconnexion (switch ou routeur) a la même signification que la notion de port présente à la couche Transport.

Exercice 3 (TD) : équipements d'interconnexion, fonctions et OSI

Le schéma ci-dessous présente la topologie d'un réseau vue du dessus.



Nommé ou pas nommé, vous devriez compter 4 types d'équipements dans ce réseau. Lesquels ?

Un _switch_ est un équipement multi-ports qui interconnecte des équipements au niveau liaison (2 OSI), il étend un réseau local. On parle aussi de _domaine

diffusion_ pour désigner l'ensemble des ordinateurs qui peuvent être joints avec seulement leur identifiant MAC. Les domaines de diffusion s'arrêtent dès qu'ils rencontrent un équipement qui implémente une couche OSI 3. Un switch analyse l'en-tête des trames qui lui arrivent. Il redirige (on dit aussi “_commute_”) une trame entrante sur le port de sortie par lequel il sait qu'il peut joindre l'adresse MAC de destination qu'il aura lu dans la trame. On verra ultérieurement comment il apprend quelle @MAC se cache derrière quel port et construit ainsi sa _table de commutation_.

Un _route(u)r_ est un équipement multi-ports qui interconnecte des réseaux locaux (3, OSI) et stoppe/cloisonne l'étendue de domaines de diffusion. Il analyse les en-têtes des paquets qui lui arrivent et les redirige (on dit aussi “_route_”) vers l'interface par laquelle il sait qu'il peut joindre, pour le plus bas des “coûts”, l'adresse IP de destination qu'il aura lu dans le paquet. On ne verra probablement pas dans ce cours comment se construisent les _tables de routage_.

A noter que le _hub_ est un équipement multi-ports qui

étend une infrastructure au niveau physique. Tout ce qui rentre par un port est redirigé systématiquement sur tous les autres ports. La résultante est que les ordinateurs réunis par un hub “entendent” les communications de chacun. De plus, si deux ordinateurs communiquent en même temps leurs trames peuvent entrer en collision suivant le type de câble utilisé. De nos jours, on n'utilise plus ces types d'équipement (excepté dans notre salle réseau pour jouer un peu). Un répéteur étend aussi une infrastructure au niveau physique sur la distance en régénérant un signal. On n'en trouve plus dans les réseaux filaires mais on peut en voir dans les réseaux sans fil.

Indiquez pour chaque équipement sur le schéma ci-dessus le niveau OSI le plus haut pris en charge par chacun (câbles y compris).

Dessinez les contours des domaines de diffusion (i.e. réseaux locaux).

Discutez sur le fait d'inclure totalement ou partiellement les ordinateurs et les routeurs dans les domaines de diffusion. Dans ce contexte, quel est le rôle des cartes réseaux que

Exercice 4 (TP) : paramètres de configuration réseau de votre machine

Les commandes suivantes sont disponibles sur les systèmes linux et peuvent s'exécuter depuis un compte simple utilisateur.

En "simplifiant un peu", une machine a besoin d'une adresse MAC pour exister sur un réseau local et d'une adresse IP pour exister sur une interconnexion de réseaux. La carte réseau est le matériel qui permet à une machine de se connecter à un réseau local, on parle d'interface. Au niveau "soft", ces interfaces sont désignées par des petits noms, qui sont bien utiles quand il y a plusieurs interfaces... La commande ``ip link`` permet d'obtenir des informations de configuration des interfaces de votre machine au niveau liaison, ``ip address`` donne en plus les informations de niveau réseau. Depuis Debian 10, ces commandes remplacent la commande ``ifconfig``.

Exécutez les commandes et apprenez à identifier les noms d'interface, les adresses MAC (séquence de 12 caractères hexadécimaux) et les adresses IP (la version 4 est notée en décimal pointée) associées. L'enseignant pourra vous faire remarquer comment observer si l'interface est active mais

ne s'étendra pas sur les notions de masques ou d'adresses "broadcast".

L'interface ``lo``, qui signifie `_loopback_`, permet de contacter la machine locale sans passer par une interface qui serait accessible depuis l'extérieur... Utile pour tester un serveur qui tourne en local.

Du fait qu'une communication soit initiée par une application, communiquer avec une machine (qu'elle soit présente sur le réseau local ou sur l'interconnexion des réseaux) signifie lui adresser un paquet à son adresse IP. Sur un réseau local, connaître ses voisins signifie connaître les adresses MAC associées à l'adresse IP de ses voisins. On verra plus tard comment obtenir l'adresse IP d'un destinataire ainsi que comment déterminer l'adresse MAC d'un voisin à partir de son adresse IP. La commande ``ip neighbour`` permet de s'informer sur l'état des connaissances de la machine. Exécutez la commande et relevez quelles informations vous apprenez à propos de comment joindre chaque voisin. On appelle cet état le cache arp du nom du protocole qui permet d'identifier l'adresse MAC d'un voisin donné son IP... Depuis Debian 10, cette commande remplace la commande ``arp``.

Le fichier `/etc/services` présente les ports réservés sur lesquels tournent les applications les plus communes. Vous pouvez jeter un oeil par exemple avec `cat /etc/services | less` ou `cat /etc/services | grep 21`. La commande `ss -taupen` permet de connaître l'état des connexions de votre machine avec les autres machines. La commande renseigne les connexions que vous soyez client ou serveur dans les communications. Exécutez la commande en ajoutant incrémentalement des lettres de paramètres (t, tcp ; u, udp ; p, process ; n, numeric ; a, all ; e, ?) et apprenez à reconnaître les services qui tournent sur votre machine, les ports utilisés dans des connexions et l'état des connexions. Depuis Debian 10, cette commande remplace la commande `netstat`.

Exercice 5 (TP) : ping, ICMP, ARP, switch et routeur avec filius