

DEVOPS 01

Réseaux et sécurité

Unité 1

Un peu d'histoire

Une brève histoire d'Internet

Dans les années 1950, les machines ne pouvaient communiquer qu'avec une seule machine à la fois. Les universités ont rapidement exprimé le besoin de pouvoir communiquer en temps réel directement avec plusieurs machines pour échanger des données, plutôt que de passer d'un interlocuteur à un autre successivement. C'est alors qu'ils ont conceptualisé une architecture réseau qui ne serait plus centralisé mais **maillé**.

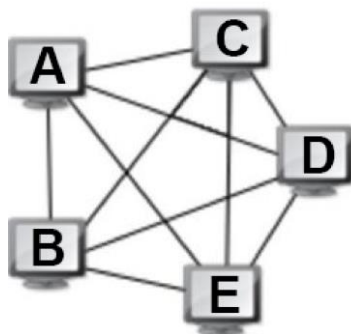
Le WEB : une toile mondiale

A la page 16 du manuel, nous avons une représentation du câble sous-marin à fibre optique qui va relier le Cameroun et le Brésil (6000 km de long). C'est à travers de tel câble que passent les données échangées d'un pays à un autre, d'une ville à une autre.

Internet est similaire à une gigantesque toile d'araignée (en anglais : WEB) où chaque machine est connectée aux autres. On appelle cela un réseau en maille :



Exercices



Voici une représentation d'un réseau en maille.

Si la machine A veut communiquer avec la machine D, elle peut utiliser plusieurs chemins.

Q1 Si la liaison est cassée entre A et D, indiquez quelles sont les chemins possibles que peut emprunter alors la connexion ?

Q2 Quel est donc l'avantage d'un réseau en maille ?

Au début des années 1960, les chercheurs mettent en place un réseau pour l'armée : le réseau Arpanet, ancêtre d'Internet.

Arpanet ne comportait que 4 machines à la fin des années 60 eWt était loin des objectifs initialement fixé à savoir de faire dialoguer des machines provenant de différents réseaux. Cela ne sera réalisable qu'en 1978, avec la création du protocole TCP/IP. De plus en plus de machines intégreront Internet qui continua de grandir doucement au fil des années jusqu'à la véritable révolution de 1990 qui va être le point de départ d'une forte croissance avec la création du langage HTML et du protocole d'échange HTTP. Apparaissent les premiers

navigateurs WEB capables d'afficher des images puis la facilité d'accès à la création de noms de domaine pour tous. Cela va entraîner l' « explosion » d'Internet dans les années 1990-2000.

Mais attention, l'internet de cette époque n'était pas aussi rapide que le nôtre aujourd'hui. Dans les foyers, il fallait s'équiper d'un modem qui occupait la ligne téléphonique pendant que vous surfiez sur le WEB. (Impossible alors de recevoir ou de passer des appels téléphonique).

Il fallait plusieurs heures pour télécharger un fichier de 4 Mo ! (Vitesse entre 2 à 5 ko/s)

Une adresse pour tous !

Lorsque vous souhaitez téléphoner à un ami, vous avez besoin de son numéro de téléphone. Ce numéro est unique et lui est associé. De la même manière, sur un réseau comme Internet, les machines possèdent un numéro d'identification unique que l'on appelle :

adresse IP (Internet Protocole).

Il existe deux versions des adresses IP : **IPv4** et **IPv6**. IPv6 a été conçu pour pallier à un besoin toujours croissant de nouvelles adresses IP.

Les adresses IPv4 sont de la forme : xxxx.xxxx.xxxx.xxxx.

Vous avez un choix de combinaison d'adresses IP possible compris entre : 0.0.0.0 et 255.255.255.255

Ce qui fait environ : 4 milliards d'adresses IPv4.

Ce chiffre semble énorme mais en comparaison avec la population mondiale et le nombre de machines connectés sur Internet, il est au final petit ... au point que l'on commence à manquer d'adresse IP !

C'est pourquoi une nouvelle norme a été créée : IPv6.

Ce qui porte le nombre d'adresse disponible à : 340 milliards de milliards de milliards de milliards soit : 2.6 milliards d'adresse IPv6 possibles par millimètre carré de surface terrestre (océan compris).



Exercices

Internet est un réseau informatique mondial qui rend accessible à ses utilisateurs un certain nombre de services comme la messagerie, la publication (le Web), la communication directe (le chat) et les transferts de fichiers.

Né à la fin des années 60 comme un projet essentiellement militaire, internet (ArpaNet à l'origine) a vite été utilisé pour relier les grandes universités américaines et accélérer l'échange de connaissances et la collaboration scientifique. Au cours des années 90, internet a vu un nouveau tournant avec l'essor du commerce en ligne. Les années 2000 ont marqué le début des réseaux sociaux. En quelques décennies, internet est passé d'un projet expérimental à un réseau omniprésent, devenu nécessaire à la vie économique mondiale, et considéré aujourd'hui à juste titre comme une infrastructure critique pour nos sociétés.

Doc. b Internet court les rues

En vous aidant également de la frise chronologique de la **page 17** et du **doc b** ci-contre répondez aux questions suivantes :

Q3 Quand est né Internet et quel était son premier nom ?

Q4 Quels ont été les premiers objectifs ?

Q5 Qu'est ce qui a permis l'essor d'Internet dans les années 90 ?

Q6 Combien il y a d'utilisateurs sur

Internet en 2009 ?

En 1980, avant Internet, apparaît en France un terminal de connexion qui permet à chaque foyer d'accéder à des services en lignes comme l'annuaire téléphonique, la météo ou la solution des jeux vidéo.


Q7 Quel est le nom de ce terminal ?

Internet et les réseaux physiques

Internet est un immense réseau mondial où toutes les machines sont interconnectées entre elles. Cela est possible grâce à une grande variété d'infrastructures physiques :

- Câbles
- Antennes et relais
- Satellites
- Fibres optiques

A partir de la carte interactive (<https://www.submarinecablemap.com/>) vous pouvez avoir un aperçu de l'ensemble des câbles sous-marins qui relient les pays entre eux sur le réseau Internet. En cliquant sur un câble vous pouvez obtenir des informations précises :




[Submarine Cable List](#)
Circe South
[Email link](#)
RFS: 1999 February
Cable Length: 115 km
Owners: VTLWavenetdf, euNetworks
URL: n.a.

Landing Points
[Cayeux-sur-Mer, France](#)
[Pevensey Bay, United Kingdom](#)

Exemple : le câble reliant Cayeux-sur-Mer (France) à Pevensey Bay (Angleterre) fait 115km et appartient aux sociétés « VTLWAVENETDF » et « euNetworks ».

En 1858, il était possible d'envoyer un télégramme d'Europe vers l'Amérique grâce a un câble unique qui traversait l'océan Atlantique.



Exercices

Q1 Aujourd'hui combien de câbles ont été installés au fond des mers ?

Q2 Sur combien de kilomètres s'étend ce réseau ?

Q3 Quel pourcentage total du trafic internet mondiale supporte ces câbles ?

Q4 Pourquoi est-il nécessaire d'en ajouter des nouveaux régulièrement ?

Mode de transmission	Type de réseau	Débits constatés
Fibre optique domestique	Câble (fibre optique)	300 Mbit/s à 1 Gbit/s
ADSL	Câble (réseau téléphonique)	1 à 70 Mbit/s
Réseaux câblés urbains	Câble (cuivre)	600 Mbit/s
4G	Sans fil	30 Mbit/s
Satellite	Sans fil	20 Mbit/s

Sur votre machine, les fichiers sont stockés sous forme binaire : une série de bits qui peuvent prendre soit la valeur 1 ou 0.

Lorsqu'un fichier est envoyé sur internet, les bits le constituant sont convertis en impulsions électriques qui pourront traverser les câbles en cuivre. (très bon conducteur de courant électrique)

Et le signal électrique reçu est de nouveau reconverti en binaire pour reconstituer le fichier au format numérique.



Exercices

Q5 Dans le cadre d'une transmission par la fibre optique, sous quelle forme de signal sont envoyés les données ?

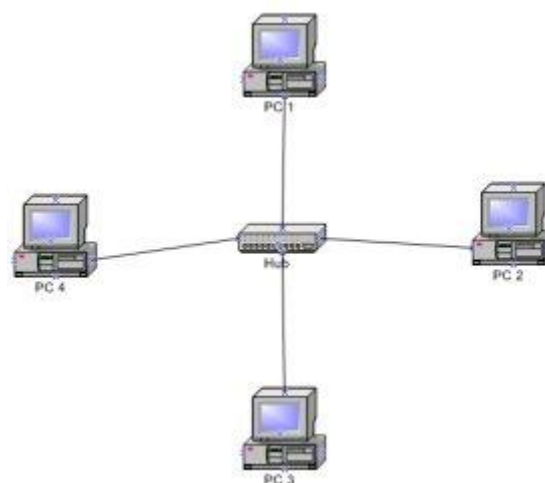
Le signal électrique a tendance à s'atténuer en fonction de la distance qu'il a à parcourir dans le câble. Plus votre câble est long, plus le signal s'atténue et l'on risque de perdre l'information transmise. Avec la fibre optique, il n'y a pas d'atténuation grâce aux propriétés physiques de la lumière.

L'un des principaux intérêts d'appareils comme les ordinateurs, les tablettes, les smartphones, est de pouvoir communiquer entre eux et avec des serveurs qui leur procurent une énorme quantité d'informations.

Le réseau mondial Internet descend du réseau militaire **Arpanet** créé par l'agence **ARPA** (pour Advanced Research Project Agency) née en 1958 aux Etats-Unis. Au début des années 1960, l'idée apparaît de découper une information en paquets indépendants, qui connaissent chacun l'adresse du destinataire et peuvent emprunter des routes différentes. À l'arrivée, l'information est reconstituée à partir des paquets reçus. Si un paquet s'est perdu, lui seul est renvoyé.

Le **protocole IP** a été conçu en 1968 : chaque ordinateur a une adresse numérique codée sur 4 octets. Ce protocole permet la communication sur un réseau distribué qui est constitué d'un maillage d'ordinateurs sans aucune hiérarchie entre eux.

A cette époque, le réseau téléphonique par exemple avait **une architecture en étoile** : un central était relié à un ensemble de téléphones et un autre central lui-même relié à un ensemble de téléphones, etc. Si le central auquel était relié un téléphone tombait en panne, le téléphone était coupé. Il n'y avait qu'un seul chemin possible pour relier deux téléphones quelconques. Le système de maillage offre par contre une multitude de chemins entre deux appareils.



En 1969, le réseau Arpanet relie les universités américaines et le premier courrier

électronique est envoyé en 1971.

Le protocole TCP/IP est élaboré en 1973-1974 par Vinlon Cerf et Bob Kahn.

Transmission de données

Une communication entre deux appareils peut s'établir à l'aide de protocoles. Chaque protocole est un ensemble de règles bien définies. Les plus connus sont :

- **Transfert Control Protocol (TCP),**
- **Internet Protocol (IP),**
- **Hypertext Transfer Protocol (HTTP).**

Ce sont des protocoles réseaux qui régissent Internet et le Web.

Plusieurs modèles existent pour schématiser les différents protocoles mis en œuvre dans une transmission de données. Ils sont présentés sous la forme de couches superposées.

On parle ainsi de modèle **TCP/IP**, les deux protocoles étant au cœur du modèle utilisé dans les réseaux de communication.

Le modèle TCP/IP propose quatre couches, qu'on présente en général de haut en bas.

- **La couche application** avec le protocole HTTP (ou HTTPS). On trouve aussi à ce niveau de nombreux protocoles : DHCP, DNS, FTP, ..., SMTP, Telnet, TLS/SSL, etc.
- **La couche transport** avec le protocole TCP. Le protocole UDP est aussi à ce niveau.
- **La couche internet** avec le protocole IP (IPv4 et IPv6).
- **La couche accès réseau** avec les protocoles Ethernet, Bluetooth, Wi-fi mais aussi le protocole ARP, Address Resolution Protocol, utilisé pour connaître une adresse physique (MAC) à partir d'une adresse IP. On parle de sous couche MAC (Media Access Control).

Le modèle OSI, présente aussi un modèle en couches. Pour simplifier disons que la couche application du modèle TCP/IP regroupe les trois couches supérieures du modèle OSI (**application, présentation et session**), les couches transport et internet correspondent aux couches transport et réseau, et la couche accès réseau du modèle TCP/IP regroupe deux couches, la couche lien (ou liaison de données) et la couche physique du modèle OSI.

La couche physique assure la transmission par câble, fibre, ou onde.

On appelle modèle TCP/IP, un modèle centré sur les protocoles TCP et IP. Le protocole IP implémente officiellement la couche internet (les réseaux et en particulier l'Internet), le protocole TCP se partage officiellement l'implémentation de la couche transport avec UDP.

Les données de chaque couche sont encapsulées par la couche de dessous chez l'émetteur et désencapsulées par la couche de dessus chez le récepteur.

Par exemple, pour demander une simple page Web, un navigateur prépare une requête http (un message de demande).

Cette requête est mise en forme dans **un paquet** ou **plusieurs paquets** par le protocole **TCP**.

Ces paquets sont encapsulés dans des paquets munis des adresses IP pour pouvoir circuler entre différents nœuds du réseau. Ces paquets IP sont à leur tour encapsulés dans des trames, Ethernet ou Wifi, pour ce qui nous concerne.

Ces trames ajoutent aux paquets les adresses MAC (Médium Access Control) qui sont les adresses physiques des appareils.

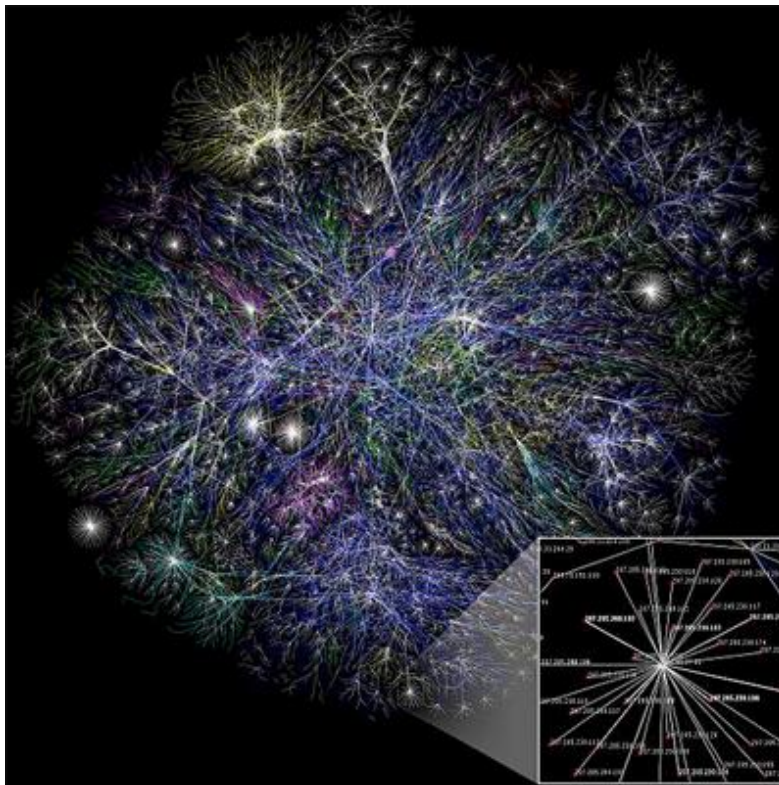
Le processus inverse est ensuite réalisé chez le récepteur jusqu'à récupérer le contenu du message de demande **HTTP** Il répond suivant le même principe en envoyant le code de la page Web.

Selon « Les Echos », sur les 7,7 milliards d'humains qui peuple la Terre :

- 5,1 milliards possèdent un téléphone mobile.
- 4,4 milliards utilisent Internet (soit 57%).
- La population mondiale s'est accrue de 1.1% alors que le nombre d'Internaute a progressé de 9.1% par rapport à 2018.
- 45% de l'humanité est présent sur les réseaux sociaux.

Internet est devenu le principal moyen de communication entre les hommes et les machines.

Sur l'illustration ci-dessous vous avez un aperçu de l'ensemble des connexions entre les machines dans le monde.



Source : Wikipédia

Internet permet donc à chacun de communiquer avec n'importe qui, n'importe où et n'importe quand. Pourtant quand on tente d'imaginer la situation cela nous semble difficile : communiquer dans une salle avec 10 personnes, tout le monde, en même temps et n'importe quand, cela est pénible. Imaginons la même chose avec 10 000 personnes ... c'est impossible. A moins d'établir un ensemble complexe de règles permettant de réguler la parole de chacun, c'est ce que l'on appelle un **protocole**.

Lorsque deux personnes se parlent, elles suivent un protocole, un code, qui permet d'initier la conversation.



Exercices

Q1 A partir de ce document déterminer les différentes étapes du protocole d'échange entre deux individus. (Pour demander son chemin, par exemple)

Pour compléter, voyons en détail les moyens nécessaires pour communiquer :

- La parole
- Le téléphone
- Le courrier

Pour chacun de ses moyens vous voyez qu'il faut :

- Un émetteur, celui qui envoie le message.
- Un récepteur, celui reçoit le message.
- Un support, sur lequel le message est transmis.
 - L'air pour la parole.
 - Le câble téléphonique pour le téléphone.

L'exemple de la lettre permet de nous faire comprendre la notion d'encapsulation de l'information : la lettre est pliée, mise dans une enveloppe timbrée avec l'adresse du destinataire. Puis un intermédiaire, le service de La Poste, est chargé de la livrer à la bonne adresse.

De même, lorsque deux objets connectés veulent communiquer sur **Internet**, ils doivent obéir à un ensemble de règles définies dans le protocole appelé : **TCP/IP**. Sur le réseau, chaque machine doit alors posséder une adresse pour être capable de recevoir ou envoyer un message : C'est son adresse IP.



Exercices

Q2 Faites des recherches sur l'adresse IP et inventez une adresse IP qui respecte la convention explicitée dans le texte.

Q3 De combien d'octet est composé une adresse IP ?

Q4 De combien de bit est composé un octet ? en déduire le nombre de bit dans une adresse IP.

Q5 Sur votre ordinateur, entrez dans l' « invite commande » et taper la commande « *ipconfig* ». Décrire ce que vous voyez.

L'adresse IP est toujours accompagnée de son **MASQUE DE SOUS-RESEAU** qui permet de connaître quelle partie de l'adresse IP est associée à l'adresse du réseau et à l'adresse de la machine.

Exemple :

Adresse IP :	192.168.0.32
Masque :	255.255.255.0
Le masque ici est composé de : <ul style="list-style-type: none">• 3 premiers octets à la valeur 255.• Le dernier octet à la valeur 0. Donc les 3 premiers octets de l'adresse IP correspondent à la partie réseau de l'adresse IP. Le dernier octet de l'adresse IP correspond à l'adresse de la machine sur ce réseau.	



Exercices

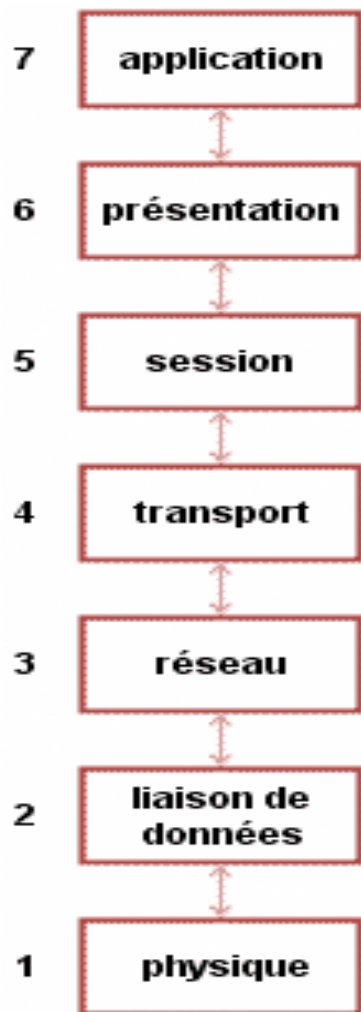
Q6 Déterminer le rôle du protocole TCP (Transmission Control Protocol) lors des échanges de paquets sur Internet.

Q7 Le protocole TCP est-il fiable ? expliquez.

Q8 Google possède des serveurs à l'adresse IP : 8.8.8.8. Testez si votre machine communique correctement avec cette adresse IP, en tapant « ping 8.8.8.8 » dans l' « invite de commande ».

On appelle « **paquet** » les unités élémentaires de l'information qui circule dans un réseau. Il s'agit d'une suite d'octets suffisamment courte (1500 maximum) pour pouvoir être communiquée sous forme numérique et sans erreur sur un câble de communication ou tout autre type de liaison numérique.

Le protocole TCP/IP est un modèle en **couches**



Quand une Machine 1 envoie un paquet sur Internet, celui-ci est traité par plusieurs couches avant d'arriver à destination. (À la manière d'un colis postal, qui passe par le centre postal -> centre de tri de la ville -> plateforme de distribution -> facteur -> destinataire).

1. La couche application
2. La couche transport
3. La couche Internet
4. La couche accès réseau.

Notions complémentaires :

Un **serveur** est une machine qui **offre un service**. Par exemple, un serveur WEB est tout simplement un ordinateur qui héberge des pages WEB et qui permet à d'autres **machines distantes**, ou des **clients**, d'y consulter le contenu.

Les demandes émises par un ordinateur client à un serveur sont appelées : des **requêtes**.

Récupération de paquets

Pour pouvoir communiquer, un protocole commun doit être utilisé entre des parties qui sont à tour de rôle émettrice et réceptrice. Par exemple au niveau du protocole TCP, un client envoie une demande de connexion à un serveur qui accepte la connexion et répond, "c'est bon, vous pouvez vous connecter". Le client envoie alors à son tour une confirmation, "puisque vous m'autorisez, je me connecte".

L'information transmise est découpée en paquets. Et même si la fiabilité des réseaux a beaucoup progressé, il peut arriver qu'un paquet se perde ou soit endommagé pendant la transmission. Parfois ce n'est pas très grave, par exemple dans le cas d'un message audio entre deux personnes ou d'une image dans une vidéo. Dans d'autres cas, chaque paquet est nécessaire à l'arrivée pour reconstituer l'ensemble de l'information correctement. Si nous pouvions savoir avec certitude que par exemple sur cinq paquets, un paquet est sûr d'arriver, il suffirait d'envoyer cinq fois chaque message. Mais cette connaissance est impossible à établir de manière certaine. Des protocoles doivent donc être mis en œuvre pour savoir si un paquet s'est perdu et le récupérer si c'est le cas.

Supposons que les messages sont envoyés d'un émetteur A à un récepteur B. Un premier message est envoyé par A, B répond alors qu'il a bien reçu un message de A. Il s'agit d'un système d'acquiescement. Si A ne reçoit pas de réponse, c'est que soit son message, soit la réponse, s'est perdu. De même si B ne reçoit plus rien, c'est que le second message de A ou sa réponse s'est perdu. On peut alors imaginer que A renvoie à intervalles réguliers son message tant qu'il n'a pas reçu de réponse et B envoie des réponses lui aussi à intervalles réguliers. Le problème est alors de décider d'un temps d'attente. Mais quelque soit cet

intervalle, il pourra arriver que A reçoivent un acquittement d'un message n alors qu'il a déjà envoyé un message $n+1$. Une solution à ce problème est de numéroté les paquets et la manière la plus économique est d'utiliser un bit.

Le protocole du bit alterné fonctionne ainsi. Il se situe au niveau de la couche accès réseau du modèle TCP/IP ou de la couche lien du modèle OSI. Ce protocole permet de retransmettre des paquets qui ont été perdus ou corrompus. Nous allons préciser le système de comptage des paquets.

Les messages sont envoyés d'un émetteur A à un récepteur B. Chaque message est composé de données et d'un bit. Soit 0, soit 1. B renvoie à A un message qui est un accusé de réception comportant le même bit que le message reçu, 0 ou 1.

Quand A envoie un message, il le renvoie à intervalles réguliers avec le même bit, jusqu'à ce qu'il reçoive l'accusé de B qui contient le même bit. Lorsque c'est fait il envoie le message suivant avec un bit différent.

Quand B reçoit un message qui n'est pas corrompu, il envoie à intervalles réguliers l'accusé de réception avec le même bit que celui du message reçu, jusqu'à ce qu'il reçoive un message de A qui contient un bit différent. A partir de ce moment, il envoie l'accusé avec le nouveau bit. Et ainsi de suite.

À cause des délais de transmission, A peut continuer à recevoir des accusés avec le bit 0 alors qu'il a déjà commencé à envoyer un message avec le bit 1. Ces accusés sont alors ignorés.

Pour démarrer le protocole, A peut commencer par envoyer des messages quelconques avec un bit 1. Le premier message avec un bit 0 signifie alors le début de la transmission.

Des problèmes subsistent quand même. Un message n est envoyé plusieurs fois jusqu'à la réception d'un acquittement. L'un de ces messages n s'est un peu perdu et arrive après l'envoi et l'acquittement d'un message $n+1$. Le récepteur va croire qu'il s'agit d'un message $n+2$ et envoyer un acquittement. Comment l'émetteur va gérer cette situation ? Il est important que le système retrouve son état normal le plus rapidement possible même si une mauvaise donnée a été acceptée par le récepteur.

Protocoles TCP et IP

La norme TCP/IP permet à tout appareil connecté de dialoguer sur tous les réseaux internet. Cette norme de communication est fiable.

Deux machines quelconques, aussi éloignées soient-elles, peuvent dialoguer entre elles. Mais cette fiabilité entraîne en contrepartie une faiblesse au niveau de la sécurité.

Le principe est similaire à celui d'une expédition de colis par la poste. Si je dois envoyer un meuble en kit, qui est entièrement démonté, je peux décider de l'envoyer pièce par pièce en empaquetant chaque pièce soigneusement. Ensuite je porte tous les paquets à la poste de mon quartier. Celle-ci va les transporter jusqu'à la poste centrale qui dirigera certains paquets vers l'aéroport pour un transport aérien, d'autres vers une gare pour un transport ferroviaire. On peut imaginer toutes les possibilités jusqu'au destinataire final. Comme la notice de montage est jointe à chaque paquet, celui-ci pourra vérifier s'il ne manque pas un colis et éventuellement faire une réclamation.

Sur le réseau internet, si j'envoie un message, celui-ci est décomposé en paquets qui contiennent des blocs de données, un ou plusieurs paquets suivant la taille du message **C'est le rôle du protocole TCP.**

Les différents paquets vont transiter par différents points, des routeurs, jusqu'à la destination finale. Chaque paquet peut avoir une route différente. C'est le rôle du protocole IP de les faire arriver au bon destinataire qui pourra savoir d'où viennent ces paquets. À l'arrivée, l'ensemble des paquets permet de reconstituer le message envoyé, c'est à nouveau le protocole TCP qui est mis à l'œuvre.

Pour simplifier, le principe du protocole IP est donc d'ajouter aux paquets de données l'adresse IP de l'expéditeur et celle du destinataire. Cela permet au destinataire d'envoyer les accusés de réception et ainsi à l'émetteur de renvoyer éventuellement des paquets qui se seraient perdus.

Le protocole HTTP est utilisé au-dessus. Pour sécuriser le transfert, on peut remplacer http par HTTPS. Celui-ci crée sur la liaison TCP/IP une sorte de tunnel virtuel qui crypte et enrobe les données.

Adresse IP

L'adresse IP est une adresse numérique permettant d'identifier les appareils connectés à un réseau. Il en existe actuellement deux versions, **IPv4** et **IPv6**.

Précisons en quoi consiste une adresse IPv4 (version 4).

Nous avons 4 nombres entiers séparés par des points, chacun compris entre 0 et 255 inclus.

Chaque nombre représente un octet (**8 bits**).

Certaines adresses ou plages d'adresses sont réservées et ne peuvent être utilisées. Donc moins de

$2^{32} = 4294\ 967\ 290$ adresses peuvent être attribués aux interfaces des hôtes IPv4 (le matériel informatique connecté à un réseau utilisant l'Internet Protocol).

Chaque adresse IPv4 publique, utilisable sur Internet est unique dans le monde. Depuis plusieurs années, ce nombre d'adresses disponibles est devenu insuffisant pour connecter tous les appareils.

Les adresses privées sont utilisables sur un réseau local. Donc la question du nombre d'adresses permettant de satisfaire les besoins ne se pose pas dans les mêmes termes.

Afin d'augmenter le nombre d'adresses publiques disponibles, une nouvelle version a été mise en œuvre, avec les adresses IPv6 (version 6). Le principe reste le même mais nous avons 8 nombres entiers séparés par des deux-points, chaque nombre représentant deux octets. L'adresse est donc codée sur un total de **16** octets.

Les deux versions d'adresses sont utilisées actuellement sur Internet.

Le plus souvent on demande une page Web à partir d'un raccourci ou d'un lien, parfois en écrivant le nom comme python.org ou linuxfr.org.

Voyons comment obtenir l'adresse IP d'un site ou le nom d'hôte à partir d'une adresse IP et quelques informations supplémentaires avec Python. Pour cela, nous utilisons le module [socket](#).

Dans le contexte matériel, le mot socket signifie *prise* ou *connecteur*, par exemple pour brancher un processeur sur une carte mère. Ici il s'agit d'un connecteur réseau, plus précisément d'une interface de connexion qui se situe sous la couche TCP et inclut la couche réseau IP.

```
import socket

print(socket.gethostbyname("python.org"))
print(socket.getaddrinfo("python.org",80,proto=socket.IPPROTO_TCP))
print(socket.getaddrinfo("example.org",80,proto=socket.IPPROTO_TCP))
print(socket.gethostbyaddr("185.75.143.24"))
```

On obtient dans l'ordre l'adresse IPv4 de python.org, puis la famille d'adresses de python.org, <AddressFamily.AF_INET: 2>, avec le 2 pour le type IPv4, ensuite, c'est la famille d'adresses de exemple.org avec la famille IPv6 (type 23) puis IPv4 (type 2), enfin l'hôte ayant pour adresse 185.75.143.24, il s'agit de education.gouv.fr, (on peut deviner le sens de MEN-WEBEDU).

```
45.55.99.72 # adresse IPv4 de python.org

[(<AddressFamily.AF_INET: 2>, 0, 6, '', ('45.55.99.72', 80))]

[(<AddressFamily.AF_INET6: 23>, 0, 6, '',
  ('2606:2800:220:1:248:1893:25c8:1946', 80, 0, 0)),
  (<AddressFamily.AF_INET: 2>, 0, 6, '', ('93.184.216.34', 80))]

('MEN-WEBEDU-PROXY01.dedie.ate.info', [], ['185.75.143.24'])
```

Une adresse est constituée de deux parties permettant d'identifier le réseau et l'hôte. On utilise pour cela un **masque**. Si on donne une adresse 192.168.1.24 et le masque 255.255.255.0, le réseau est alors identifié par 192.168.1.0 et l'hôte par 0.0.0.24.

Il existe plusieurs classes de réseaux d'adresses IPv4.

Les adresses des réseaux de classe A ont un premier nombre compris entre 1 et 126. (Le premier bit du premier octet est un zéro). Un réseau x peut donc contenir toutes les adresses de x.0.0.0 à x.255.255.255, soit 16777216 adresses. On dit que le masque de réseau est 255.0.0.0.

Les adresses des réseaux de classes B ont un premier nombre compris entre 128 et 191, (les deux premiers bits du premier octet sont 10), et le masque de réseau est 255.255.0.0. Un réseau x.y peut donc contenir toutes les adresses de x.y.0.0 à x.y.255.255, soit 65536 adresses.

Les adresses des réseaux de classes C ont un premier nombre compris entre 192 et 223, (les trois premiers bits du premier octet sont 110) et le masque de réseau est 255.255.255.0.

Un réseau x.y.z. peut donc contenir toutes les adresses de x.y.z.0 à x.y.z.255, soit 256 adresses.

Les adresses avec un premier nombre compris entre 224 et 255, (les trois premiers bits du premier octet sont 111), sont réservées à un autre usage (le multicast par exemple, qui n'utilise pas le protocole TCP, et consiste à diffuser vers un groupe de récepteurs).

Un particulier qui possède une connexion à Internet par l'intermédiaire d'une box est sur un réseau de classe C. Par exemple, la box a pour adresse 192.168.1.1, le masque de réseau est 255.255.255.0 et tous les appareils connectés à la box ont une adresse de la forme 192.168.1.X.

Les classes sont de moins en moins utilisées. Actuellement les réseaux sont subdivisés en sous-réseaux et les notations utilisées permettent d'avoir un découpage plus fin. Par exemple, la notation 192.168.1.18/27 indique l'adresse IP 192.168.1.18 et précise que les 27 premiers bits constituent l'adresse du sous-réseau, les 5 derniers l'adresse de l'hôte. Le masque s'écrit donc 11111111.11111111.11111111.11100000 soit 255.255.255.224 en notation décimale.

L'adresse du sous-réseau est 192.168.1.0, l'adresse de broadcast est 192.168.1.31, c'est celle qui permet d'envoyer des informations à tous les postes du réseau. Il peut donc y avoir 30 machines hôtes d'adresses 192.168.1.1 à 192.168.1.30. L'adresse du sous réseau suivant est 192.168.1.32.

Le Domain Name System, ou DNS est le service informatique distribué utilisé pour traduire les noms de domaines Internet en adresse IP. Par exemple, le nom "example.org" est traduit en 93.184.216.34, l'adresse IP correspondante.

On peut savoir à qui est assignée une adresse IP sur le site <https://www.whois.com/whois/>. Il suffit d'entrer l'adresse 216.58.213.174 et on constate que cette adresse appartient à l'organisation Google qui possède la plage d'adresse de 216.58.192.0 à 216.58.223.255.

Whois IP 216.58.213.174

Updated 1 second ago

```
#
# ARIN WHOIS data and services are subject to the Terms of Use
# available at: https://www.arin.net/resources/registry/whois/tou/
#
# If you see inaccuracies in the results, please report at
# https://www.arin.net/resources/registry/whois/inaccuracy_reporting/
#
# Copyright 1997-2019, American Registry for Internet Numbers, Ltd.
#
```

```
NetRange: 216.58.192.0 - 216.58.223.255
CIDR: 216.58.192.0/19
NetName: GOOGLE
NetHandle: NET-216-58-192-0-1
Parent: NET216 (NET-216-0-0-0)
NetType: Direct Allocation
OriginAS: AS15169
Organization: Google LLC (GOGL)
RegDate: 2012-01-27
Updated: 2012-01-27
Ref: https://rdap.arin.net/registry/ip/216.58.192.0
```

```
OrgName: Google LLC
OrgId: GOGL
Address: 1600 Amphitheatre Parkway
City: Mountain View
StateProv: CA
PostalCode: 94043
Country: US
RegDate: 2000-03-30
Updated: 2019-10-31
```

Avec le protocole IPv4, c'est en théorie au maximum $2^32 = 4294967296$ adresses qui peuvent être attribuées simultanément. En pratique, certaines ne sont pas disponibles mais dans tous les cas ce nombre n'est plus suffisant. En 2018, on compte plus de quatre milliards d'internautes dont plus de trois milliards connectés sur des réseaux sociaux.

Les adresses IPv6 sont donc de plus en plus souvent utilisées. Elles utilisent 16 octets, ce qui permet 2^{128} adresses. Pour la notation, les octets sont regroupés en 8 groupes de 2 octets séparés par le signe deux-points. Par exemple : 2a00 : 1450 ; 4007 : 80a : :200e qui est une abréviation de l'adresse 2a00 :1450 :4007 : 80a : 0000 : 0000 : 0000 : 200e

Nous avons vu que sur Internet les machines communiquent entre elles grâce à leur adresse IP. Cependant chaque jour nous surfons sur Internet avec notre navigateur d'un site WEB à un autre, en ne saisissant uniquement que des adresses composées de caractères alphanumérique (URL). Et cela est bien pratique ! Imaginons un instant si nous devions retenir les adresses IP de nos sites WEB préférés, cela deviendrait vite ingérable. C'est donc pour nous simplifier la vie qu'ont été créés les serveurs de nom de domaine ou « **Domain Name Server** » ou DNS.

Et ne soyez pas surpris, en effet, derrière chaque adresse de site Internet, se cache une adresse IP.

Pour vous en rendre compte, tapez la commande « ping www.google.fr » dans l'invite de commandes ».

```
C:\Users\Baptiste>ping www.google.fr

Envoi d'une requête 'ping' sur www.google.fr [216.58.206.227] avec 32 octets de données :
Réponse de 216.58.206.227 : octets=32 temps=232 ms TTL=55
Réponse de 216.58.206.227 : octets=32 temps=64 ms TTL=55
Réponse de 216.58.206.227 : octets=32 temps=72 ms TTL=55
Réponse de 216.58.206.227 : octets=32 temps=43 ms TTL=55

Statistiques Ping pour 216.58.206.227:
    Paquets : envoyés = 4, reçus = 4, perdus = 0 (perte 0%),
    Durée approximative des boucles en millisecondes :
        Minimum = 43ms, Maximum = 232ms, Moyenne = 102ms
```

Vous vous rendez compte que l'on communique ici avec l'adresse : **216.58.206.227** alors que nous avons saisi **www.google.fr**

Comment fonctionne un serveur DNS ou appelé plus simplement : un DNS.

Le principe de fonctionnement est très simple. Lorsque vous voulez accéder à un site web quelconque, le DNS se charge de **convertir** le nom du **site web demandé** en une **adresse IP**. Ce processus de conversion se nomme : **la résolution**.

Principe de nommage des adresses Internet

Une adresse Internet est composée de plusieurs parties séparées par un point, organisée en arborescence.

Exemple : **www.google.fr**

En lisant de la droite vers la gauche nous retrouvons l'extension (ou **Top Level Domain**) : **FR**. Il existe des TLD nationaux (fr, it, de, es, etc.) et des TLD génériques (com, org, net, biz, etc.).

La deuxième partie est déterminée par celui ou celle qui enregistre un nom de domaine, il y a donc une infinité de possibilités.

GOOGLE est un sous-domaine de **FR**

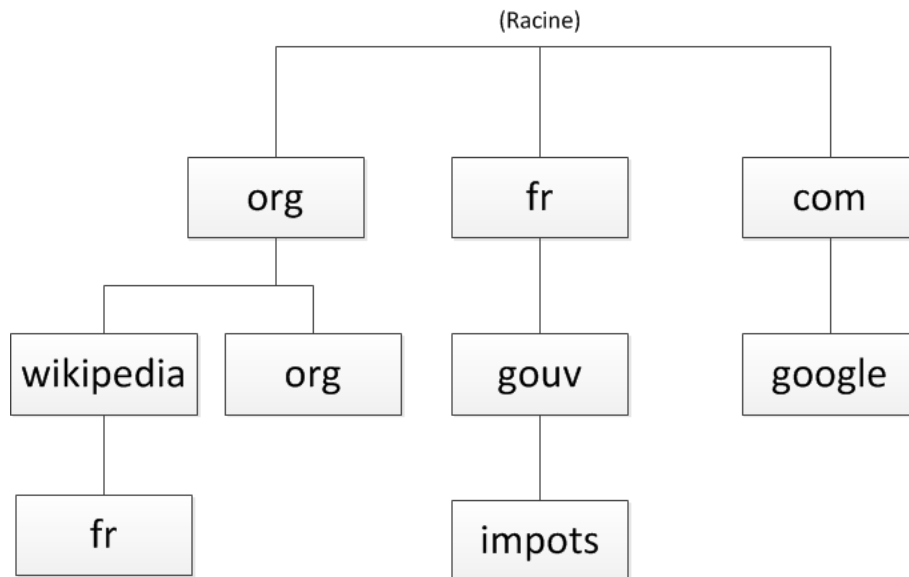
Le domaine **FR** englobe tous les sous-domaines finissant par **FR**.

Concernant la troisième partie, le **WWW**, vous l'aurez deviné seul, il s'agit d'un sous-domaine de google.fr.

Quand vous achetez un nom de domaine, vous pouvez bien entendu ensuite définir autant de sous-domaine que vous le souhaitez.

Exemple : **www.monsite.fr** peut posséder un forum accessible à l'adresse : **forum.monsite.fr**, puis un espace de tchat : **chat.monsite.fr** et une version pour mobile du site web : **mobile.monsite.fr**

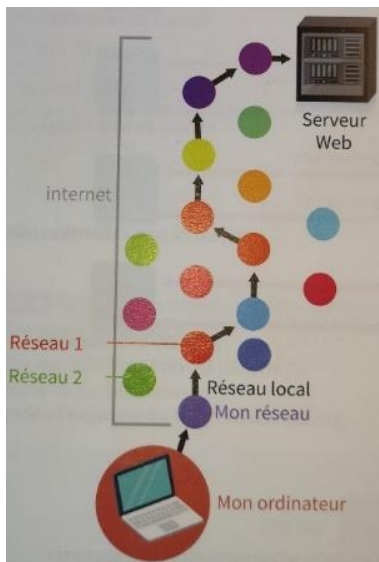
On peut donc organiser l'ensemble des domaines et des sous-domaines d'Internet sous la forme d'un arbre :



Exercices

- Q1** Dans votre navigateur saisir l'URL suivante : 195.254.146.9. Que se passe-t-il ?
- Q2** Allez sur le site <https://whoer.net/fr/checkwhois> et saisissez l'adresse (URL) de votre site préféré. Quelles informations pouvez-vous obtenir ?
- Q3** Chaque fois que saisissez une adresse Internet dans le navigateur, décrivez les étapes qui aboutiront à l'apparition de votre page web.
- Q4** Quelle(s) information(s) peut apporter l'extension d'un nom de domaine ?
- Q5** Donnez quelques sous-domaines du domaine google.fr

Depuis le début nous parlons de « Internet ». Savez-vous que cela est la contraction de l'anglais : « Inter-networks », qui veut dire : « entre réseaux ». Internet est donc l'interconnexion des réseaux du monde entier.



Regardez le schéma ci-contre, chaque bulle de couleur représente un réseau quelconque.

Quand vous êtes connecté à Internet, votre réseau local (votre ordinateur, celui des membres de votre famille et même vos smartphones), par l'intermédiaire de votre Box, l'est également.

Mais avant de comprendre comment communiquer les réseaux entre eux, voyons d'abord ce qu'il se passe quand 2 machines d'un même réseau veulent communiquer.

Le protocole Ethernet.

L'adresse MAC

Nous avons parlé précédemment du protocole TCP/IP qui permet à deux machines de communiquer sur Internet grâce à leur adresse IP.

Il existe un autre protocole qui intervient quand deux machines appartenant à un même réseau veulent communiquer : **le protocole Ethernet.**

Dans un même réseau, une machine connectée peut communiquer avec une ou plusieurs autres machines. Mais pour cibler une machine précisément il faut être capable de la localiser sur ce réseau grâce à une adresse, unique, qui permet de l'identifier : **l'adresse MAC**. Une adresse MAC est codée sur 6 octets.

Rappelons qu'un octet est composé de 8 bits, alors l'adresse MAC est codée sur :

$6 * 8 = 48$. (En informatique le signe multiplier est noté « * »)

Un bit peut prendre 2 valeurs : 0 ou 1.

Pour définir le nombre de combinaison possible pour créer une adresse MAC, nous utilisons donc la formule mathématique suivante : $2^{48} = 281474976710656$.

Vous voyez donc qu'il existe une quantité considérable d'adresse MAC disponible dans le monde, la rendant par conséquent unique.

Votre machine se connecte au réseau grâce à un matériel nommé : **la carte réseau**. Se sont donc les constructeurs des cartes réseaux qui attribuent une adresse MAC à celle-ci. Ils tiennent un registre des adresses MAC déjà utilisées pour éviter les doublons.



Exercices

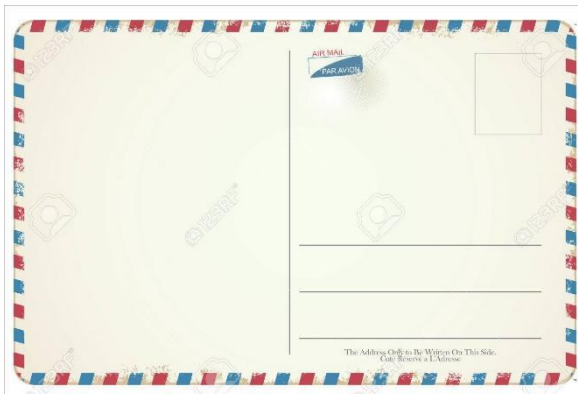
Q1 Dans l'invite de commande Windows (Exécuter et « CMD »), tapez la commande : « ipconfig -all ». Cherchez l'adresse MAC de votre carte réseau. Elle doit ressembler à quelque chose comme cela : 02-4A-2B-88-3F-46, une suite de chiffres et lettres.

Q2 Quel est l'autre nom de l'adresse MAC ? (S'aider de l'intitulé de la ligne où vous avez trouvé votre adresse).

La trame Ethernet

Ethernet est le protocole le plus utilisé car il permet aux machines d'un même réseau de communiquer facilement.

Comment cela fonctionne ?



Prenons un exemple : pour envoyer une carte postale à votre ami(e) il faut que la carte contienne :

Votre *message*, l'*adresse de votre ami(e)* et il faut *aussi signer la carte* pour que l'on sache qui a envoyé la carte.

Sur la carte il faut disposer toutes ses informations dans un **certain ordre** pour que le facteur puisse facilement **trouver les informations**, et que le destinataire puisse

savoir qui a envoyé la carte et puisse lire son contenu.

Le **protocole Ethernet** fonctionne avec cette même logique et le message que vous voulez envoyer à une autre machine du même réseau doit être encapsulé dans une **trame**, on peut dire que c'est notre « carte postale ».

La trame doit contenir 3 éléments primordiaux :

- L'adresse de l'émetteur.
- L'adresse du destinataire.
- Le contenu du message.

Il y a bien entendu d'autres éléments qui constituent la trame Ethernet :

- Le nom du protocole de la couche 3. (Voir **Unité 3**)
- Le CRC (un code de détection d'erreur de transmission).

Le HUB et SWITCH

Pour la suite de ce cours nous allons utiliser un logiciel de simulation réseau.

Regardez la vidéo suivante qui vous explique le fonctionnement du logiciel :
« **Simulateur réseau** ».



Visionnez la vidéo ci-dessous qui vous explique le fonctionnement du logiciel :
« **Simulateur réseau** ».

<https://www.youtube.com/watch?v=Mz26XkJmy1Y>



Téléchargez le logiciel :

<http://fr.lagache.free.fr/netsim/telechargement.php?lang=fr>



Exercices

A partir de la vidéo répondez aux questions suivantes :

Q3 Quel est le sens de communication d'un câble droit ?

Q4 Quel est le sens de communication d'un câble croisé ?

Q5 A partir du logiciel de simulation, reliez 2 stations avec un câble droit puis faites la même manipulation avec un câble croisé. Que conclure ?

Q6 Envoyez une trame de la station 1 à la station 4 comme montré dans la vidéo.

Ajoutez un HUB et reliez 4 stations sur celui-ci.

Envoyez une trame Ethernet entre la *station 1* et la *station 4*.

Q7 A qui est transmise la trame ?

Q8 Par quoi le HUB se fait remplacer aujourd'hui ?

Q9 Remplacez le HUB par un SWITCH et envoyez une trame de la station 1 à la station 4. A qui est transmise la trame ?

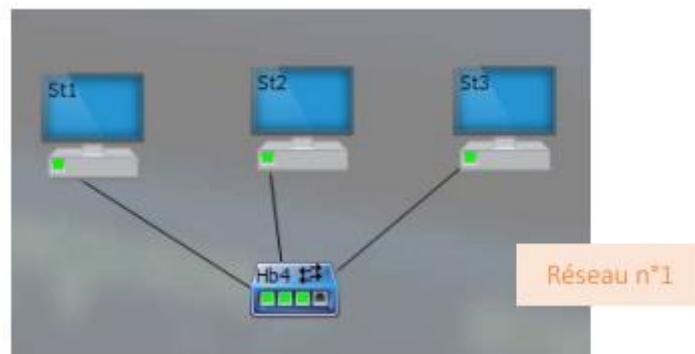


Travaux pratiques

Communiquer sur un réseau IP

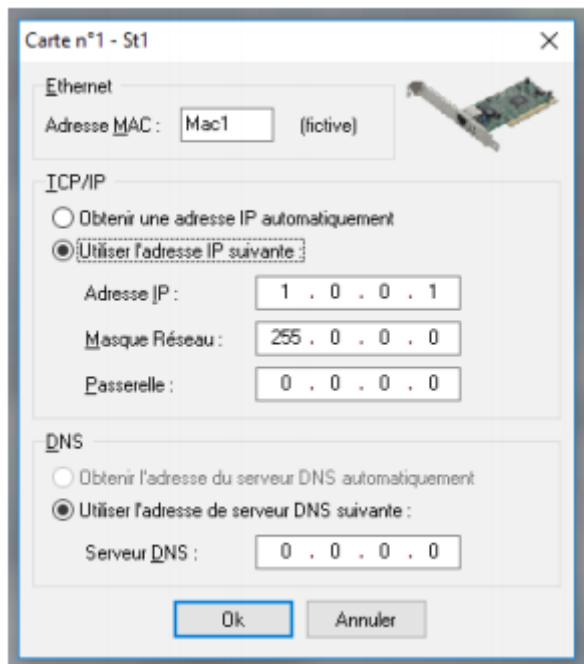
1. Un réseau IP

1.1 Dans un logiciel de simulateur de réseau, réaliser le réseau présenté ci-dessous. Il s'agira du **réseau N° 1**



1.2 On cherche à communiquer en émettant des « paquets IP ». Pour cela, il faut affecter une adresse IP à chaque machine. Pour notre exemple, la structure de l'adresse IP sera la suivante : A.B.C.D. avec : A = numéro du réseau et B.C.D. = numéro de la machine. Ainsi, l'adresse du premier poste sera :





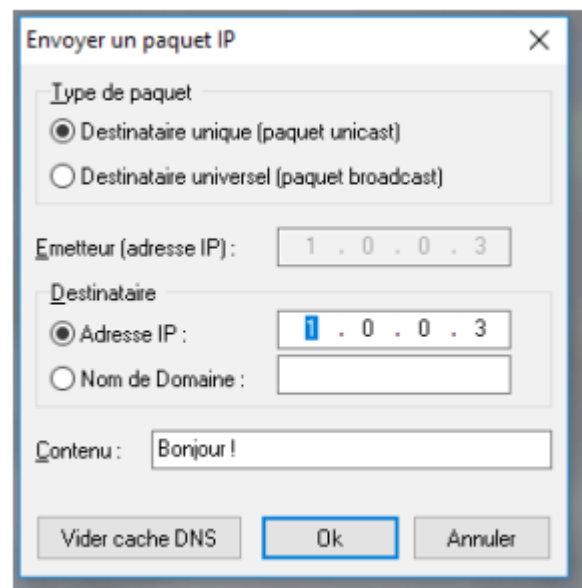
← Cliquer dans les options sur « **Légendes** » puis « **Toutes les IP** ». Double-cliquer ensuite sur **la station 1**, puis dans « **Cartes réseau** », cliquer sur modifier et rentrer les paramètres comme présenté ci-contre.

1.3 Faire de même pour la station 2 en y entrant l'adresse IP 1.0.0.2 et pour la station 3 avec l'adresse IP 1.0.0.3.



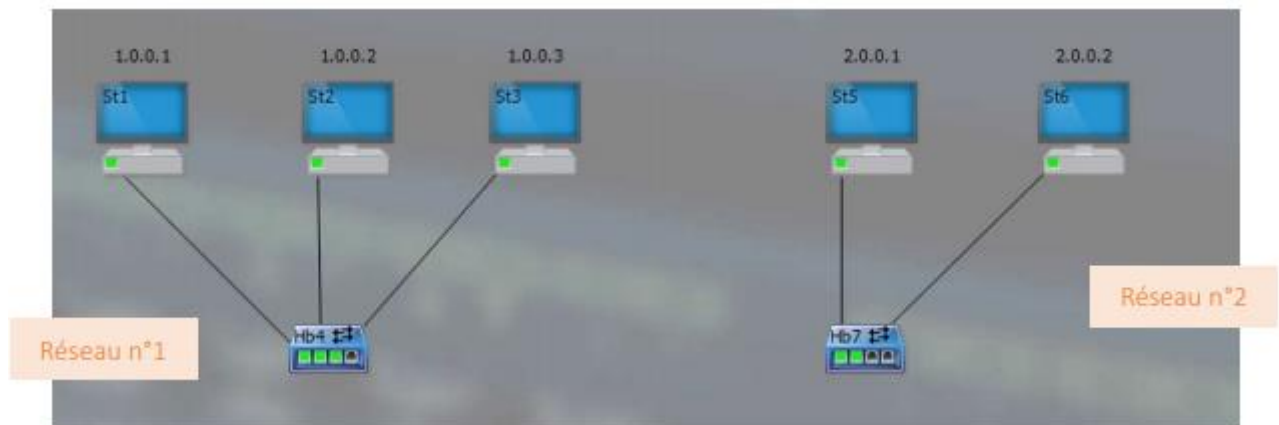
1.4 Depuis la station 1, cliquer sur le bouton droit pour « envoyer un paquet IP » à un destinataire unique, qui sera sélectionné en cliquant sur sa station (ici la station 3).

1.5 Noter les observations : y a-t-il des différences avec le hub du réseau local ?



2. Deux réseaux IP

2.1. Rajouter un deuxième réseau IP indépendant, dont les machines auront les adresses IP suivantes : 2.0.0.1 et 2.0.0.2



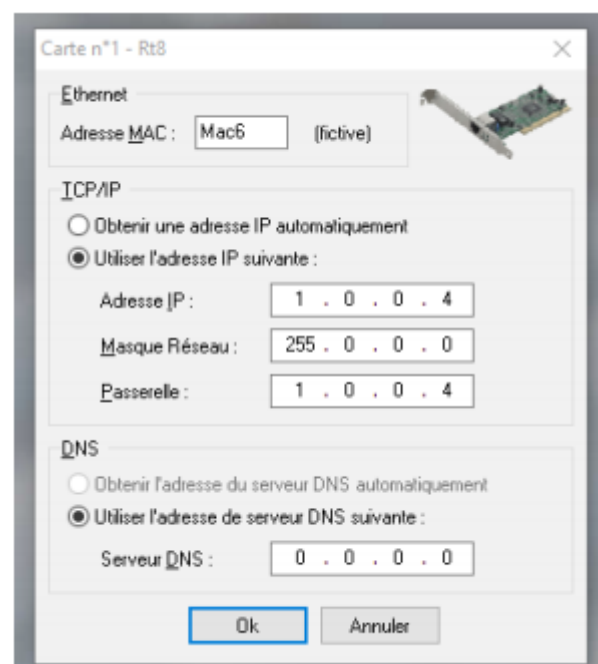
2.2. Relier les deux hubs par un câble croisé, puis essayer d'envoyer un paquet IP de la station 1 vers la station 6. Que constate-t-on ?

2.3. On constate donc qu'on ne peut pas relier les deux réseaux de cette manière. Il va falloir demander à une machine de transmettre le message d'un réseau à l'autre : cette machine est un routeur. Rajouter un routeur qui a pour but de relier les deux hubs.

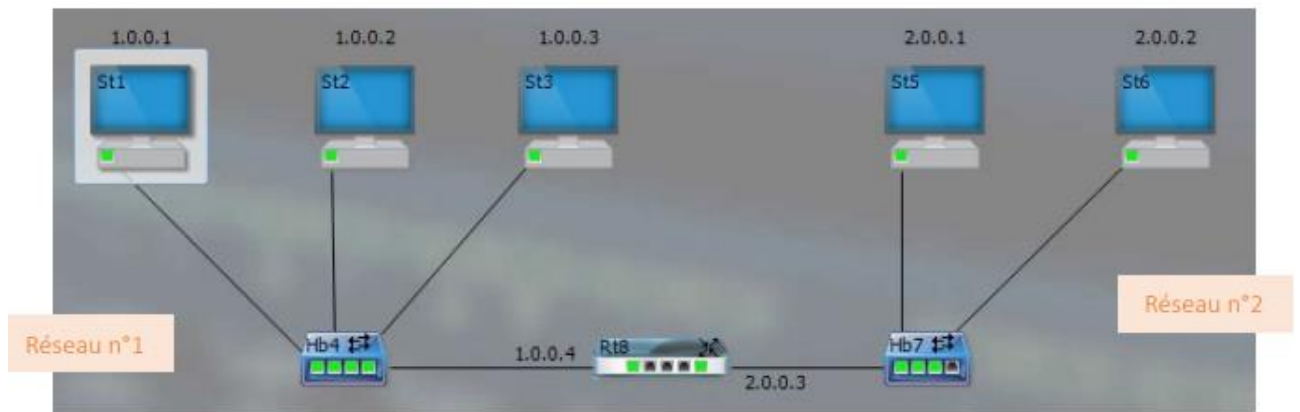
2.4. Ce routeur fait partie des réseaux 1 et 2 et leur permet donc de communiquer entre eux. Mais pour cela on doit paramétrer correctement les machines.

► Paramétrage du routeur

- Double-cliquer sur la carte du routeur reliée au réseau 1. Affecter à cette carte l'adresse 1.0.0.4 et la passerelle 1.0.0.4 comme présenté ci-contre.
- De la même manière, programmer la carte du routeur reliée au réseau 2, avec l'adresse 2.0.0.3 et la passerelle 2.0.0.




Le routeur est désormais convenablement paramétré



Comme son nom l'indique, le routeur va permettre de faire passer les informations d'un réseau à l'autre grâce à la passerelle. Il faut donc indiquer à chaque machine des différents réseaux « qui » est la passerelle.

Paramétrer maintenant les informations du poste St1 comme présenté ci-contre.

 On informe le poste 1 que s'il veut donner des informations à un autre réseau, il devra passer par le poste 1.0.0.4 qui sert de passerelle : on indique donc ici la route à suivre pour le paquet d'informations.

On comprend mieux pourquoi l'appareil qui permet ce transfert est un **routeur**.

Paramétrer également la passerelle 1.0.0.4 pour les postes St2 et St3

Paramétrer enfin convenablement les postes St5 et St6.

Quelle passerelle doit-on utiliser pour ces postes ?

2.5. Vérifier le fonctionnement du réseau : envoyer un paquet IP du poste St1 vers le poste St6. Noter les observations.

2.6. Le message est bien arrivé à destination, mais il a quand même été transmis à toutes les stations. Comment peut-on améliorer la transmission pour ne délivrer le message qu'au poste St6 ? Vérifier en transmettant à nouveau le paquet IP.

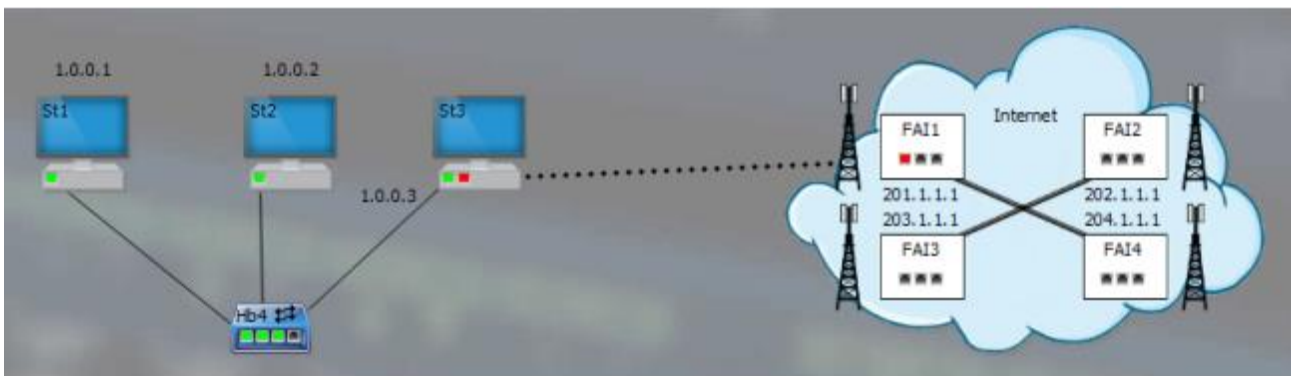
3. Réseau local et internet

3.1. Revenir sur le premier réseau (supprimer les autres éléments) et ajouter internet ainsi :

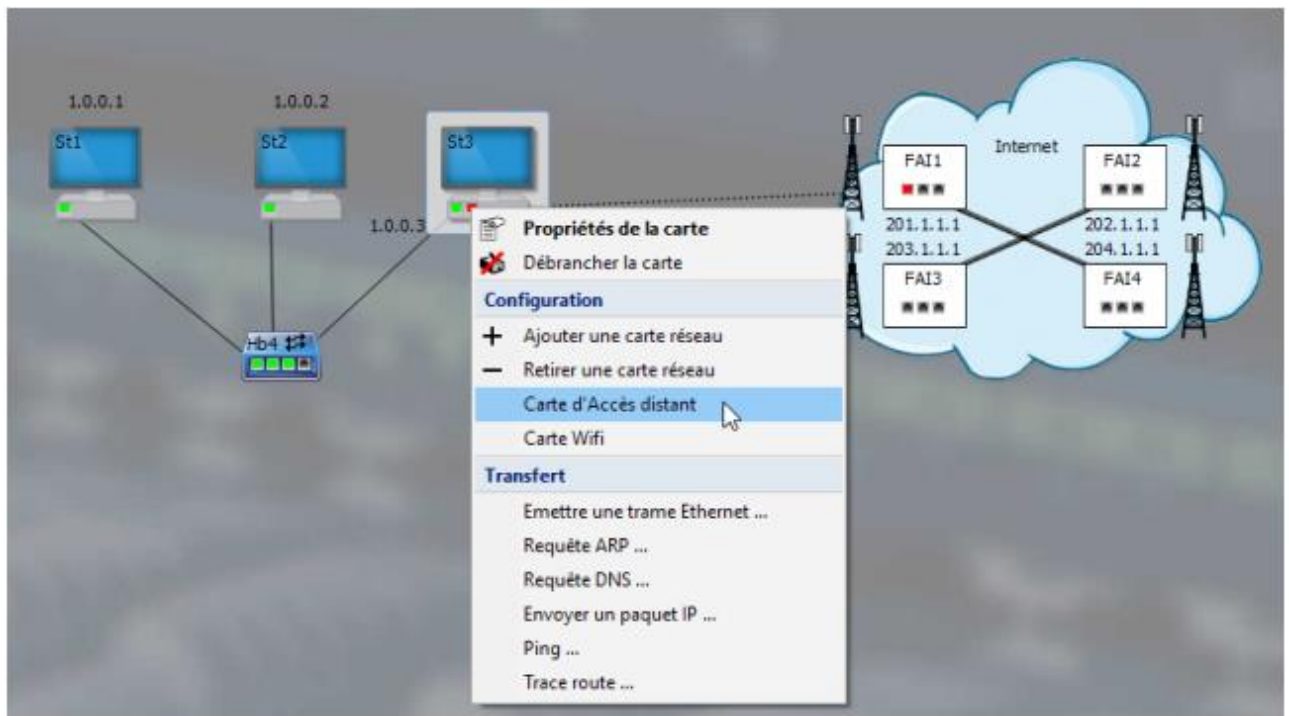


Il est possible, si on le souhaite, de changer les noms des FAI (Fournisseurs d'Accès à Internet) comme Orange, Free, SFR, Bouygues Telecom, etc.

3.2. Cette fois-ci c'est le poste St3 qui va servir au routage. Pour cela, il faut lui ajouter une carte réseau (bouton droit → ajouter une carte réseau) et la relier à un des FAI à l'aide d'un câble télécom :

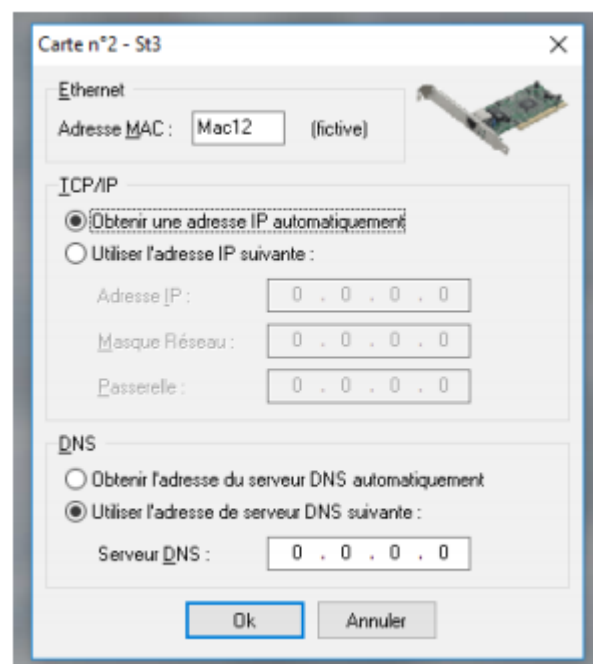


3.3. Pour se relier à internet, on a besoin d'une carte à accès distant. Faire un clic droit sur la carte réseau que l'on vient d'ajouter, puis sélectionner « Carte d'accès distant ».



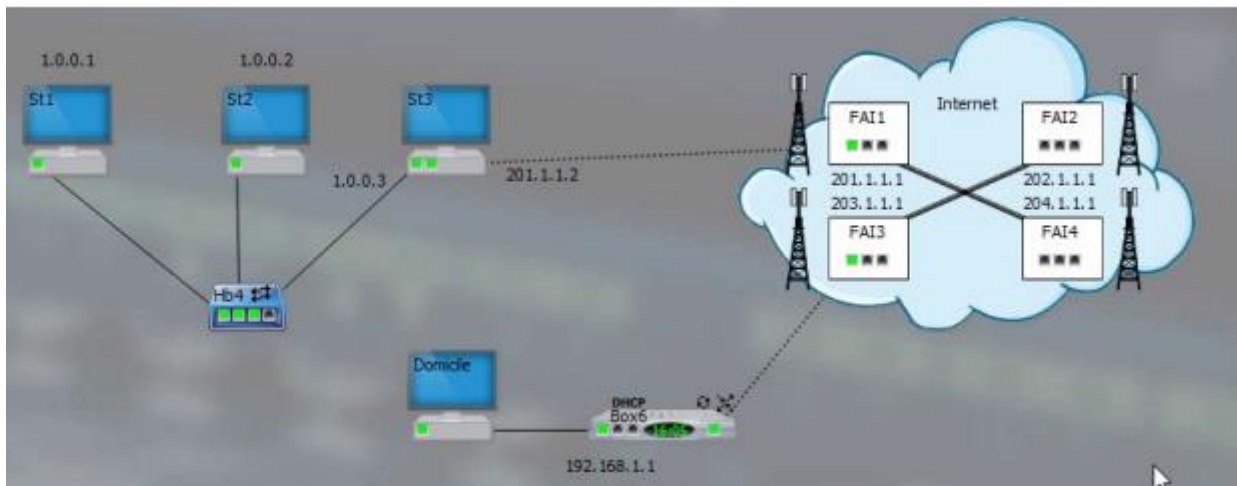
3.4. Il faut maintenant paramétrer la carte mais ce paramétrage est compliqué : c'est le FAI lui-même qui va paramétrer cette carte automatiquement. Pour cela, double-cliquer sur cette carte pour la configurer, comme présenté ci-contre

3.5. Dès que l'on clique sur « Ok », la carte dialogue avec le FAI pour se paramétrer correctement. Une adresse IP est alors attribuée à la carte à l'issue du dialogue : le réseau est relié à internet.



3.6. Pour relier un client à domicile à internet, rajouter une box (reliée à internet par un câble télécom) et un poste à domicile relié à la box.

3.7. Il faut maintenant obtenir une adresse IP convenable pour le poste à domicile. C'est très simple, il faut faire « comme à la maison » : éteindre et rallumer la box. Un dialogue s'instaure alors et des adresses IP sont attribuées à la box et au poste à domicile.



3.8. Il est temps d'envoyer un « Bonjour » du domicile au poste St1 : envoyer un paquet IP à l'adresse du poste St1. Cela bloque au niveau du fournisseur : il ne connaît pas le poste St1 ! En effet, notre poste St3 est bien connecté au réseau 1 et à internet, mais il faut encore qu'il assure le routage. Pour cela, il faut faire un clic droit sur le poste, entrer dans « Fonctionnalités, routeur ». Réessayer l'envoi.

3.9. Les adresses privées ne sont pas routables. Faire la démonstration en essayant d'envoyer un paquet IP d'un des postes du réseau 1 au poste domicile.



Le corrigé du TP « Communiquer sur un réseau IP » se situe à cette adresse :

<https://www.youtube.com/watch?v=K3lqACC2uJw>

Ou sur la chaine YouTube : Cours Bauer

Mode non connecté et mode connecté

En mode **non connecté**, les données envoyées par la machine source sont découpées en paquets avant leur envoi. Ces paquets (les datagrammes) sont alors acheminés dans le réseau indépendamment les uns des autres.

Dans le cas d'une transmission de données en mode non connecté, aucun contrôle sur le flux d'information n'est effectué. En effet, les données sont émises sans évaluation préalable du trafic ou de la qualité du transfert. C'est le cas pour les transferts de données sur Internet, basées sur des services de niveau réseau sans connexion et non fiable.

Certaines applications ne requièrent pas d'établir une connexion avant le début d'un échange : elles fonctionnent en mode non connecté : l'émetteur envoie les données sur le support de transmission et c'est ce dernier qui est en charge de les remettre au destinataire.

Remarquons que l'émetteur ne dispose, lorsqu'il soumet le message au réseau, d'aucune information concernant :

- L'état du destinataire, qui peut par exemple ne pas être disponible à cet instant.
- Le temps nécessaire jusqu'à la réception du datagramme.

Exemples : le mail (un utilisateur qui envoie un message ne vérifie pas la validité de l'adresse du destinataire), l'achat à distance.

La réalisation d'une communication **en mode connecté** nécessite une phase d'établissement d'une connexion préalablement à l'envoi des données : un circuit virtuel est mis en place. Tous les paquets à véhiculer de la source au destinataire transiteront de manière identique par ce chemin.

De même, l'acquittement de chacun des paquets reçus par les deux extrémités est transmis via le circuit virtuel établi, ce qui permet d'offrir un service fiable sans procédé technique supplémentaire.

Certaines applications requièrent d'établir la connexion avant le début de l'échange : elles fonctionnent en mode connecté.

Un tel échange est caractérisé par 3 phases bien distinctes :

1. La connexion
2. L'échange, de durée variable
3. La déconnexion qui termine le dialogue

Exemples : la prise en main à distance (effectuer des tâches sur un poste de travail à partir d'un autre poste en réseau), la communication téléphonique.

En mode sans connexion, le routage est primordial pour l'acheminement de chaque datagramme.

En mode connecté, il est indispensable pour mettre en place le circuit virtuel.

Principe de routage

Un algorithme de routage a pour rôle d'acheminer un datagramme à travers un réseau. Une telle fonction ne peut donc pas être centralisée, mais doit être présente dans chaque nœud du maillage. Elle doit, pour chaque paquet parvenant au nœud sur l'un de ses ports, choisir sur quel port de sortie l'orienter.

De manière évidente, un algorithme de routage doit être :

- **Déterministe** : face à une situation donnée, une solution unique doit être fournie : aucun choix n'est laissé à l'utilisateur ou au hasard.
- **Rapide** : en toutes circonstances, il doit être en mesure de définir rapidement une route.
- **Equitable** entre les utilisateurs dont le nombre peut être très important.
- **Robuste** : il doit fonctionner en toutes circonstances, même dégradées.
- **Optimisé** : il doit proposer le meilleur chemin possible (en temps, en distance, en encombrement.)

Les algorithmes de routages peuvent être divisés en deux familles principales :

- Les **algorithmes non adaptifs** utilisent un ensemble de routes statiques mises en place par une étude préliminaire. Ils ne tiennent pas en compte de l'état des lignes de transmission au moment de l'envoi d'un datagramme.
- Les **algorithmes adaptatifs** précèdent tout envoi de données d'une étude du contexte. Ces algorithmes se basent sur l'observation directe du maillage du réseau ou du trafic sur les lignes à un instant donné. On parle ici de routage dynamique.

Les techniques mises en œuvre sont plus complexes mais sont justifiées par les performances obtenues.

Types d'algorithmes

Nous présentons dans cette partie quelques-uns des algorithmes de routage disponibles. Certains d'entre eux sont implémentés dans des infrastructures de réseaux (routage par inondation, routage à vecteur de distance, routage hiérarchique) alors que d'autres nécessitent des adaptations pour être utilisables.

Routage par inondation

Le routage par inondation(flooding) est la technique utilisée en mode diffusion. Lorsqu'un datagramme est reçu par un routeur sur l'un de ses ports, il est réémis sur tous les autres ports.

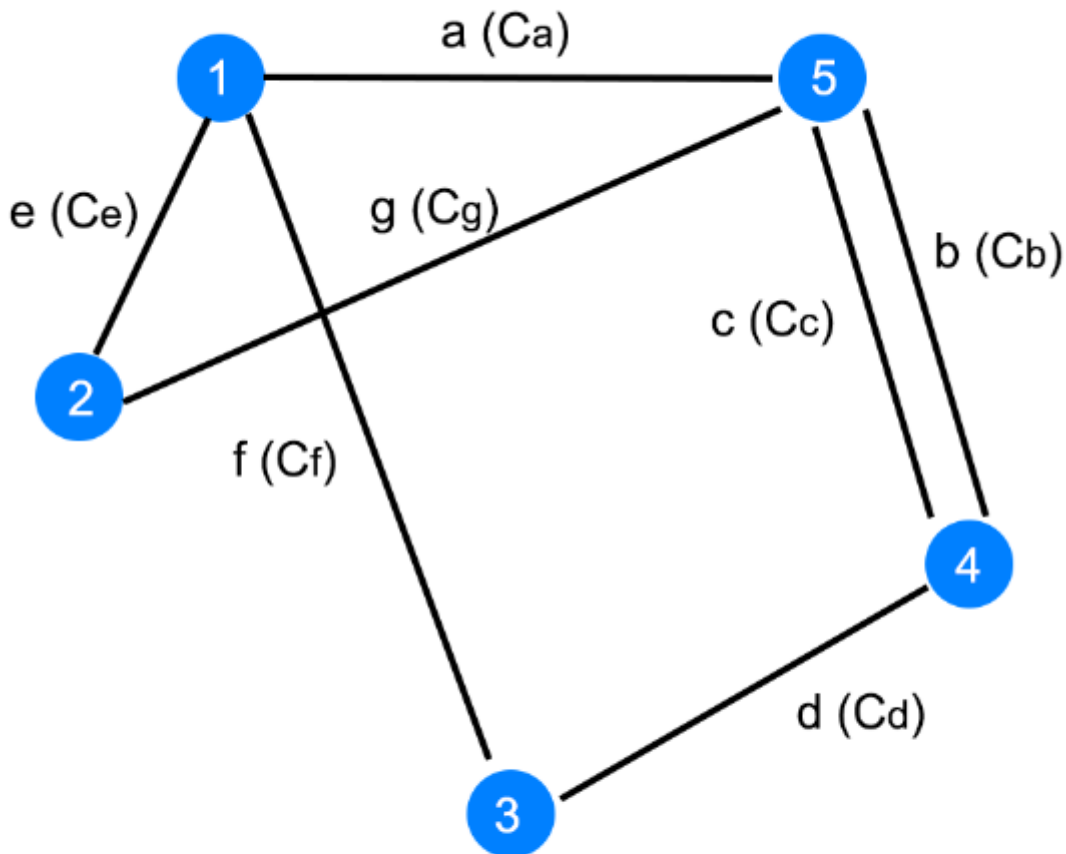
De manière évidente, cette méthode engendre un trafic très important sur la totalité des lignes de transmission. Elle ne convient donc pas à des réseaux de taille élevée ou possédant un grand nombre de nœuds.

Routage du plus court chemin

Un réseau maillé peut être représenté par un graphe $G = (X,E)$ dont l'ensemble des sommets X regroupe les routeurs et l'ensemble des arêtes E contient des lignes de transmission. Un chemin entre deux routeurs correspond alors à une chaîne de G , c'est-à-dire une suite alternée de sommets et d'arêtes.

Il est possible d'associer un coût ($C_{\text{arête}}$) chaque arête : le réseau peut ainsi être assimilé à un graphe valué.

La recherche du plus court chemin consiste alors à trouver la chaîne dont la somme des coûts des arêtes est minimale (ce coût minimal pourra correspondre, en fonction du critère important dans une situation précise, au nombre de routeurs traversés, à la distance géographique ou au trafic sur un chemin ...).



$X = \{ 1, 2, 3, 4, 5 \}$

$E = \{ a, b, c, d, e, f, g \}$

Le protocole OSPF est basé sur le coût de chaque lien, utilisé pour calculer le coût global du chemin (algorithme de Dijkstra).

Routage à vecteur de distance

Créé initialement pour les réseaux locaux Netware de Novell, puis utilisé par Internet le routage à vecteur de distance est l'un des premiers algorithmes dynamiques.

Chaque élément actif possède en mémoire une table de routage qui lui est propre. Cette structure lui indique, pour chacune des destinations connues, le port de sortie à utiliser, ainsi qu'un port par défaut pour les destinations inconnues.

Des communications inter-routeurs permettent de mettre à jour régulièrement la table de routage de chaque routeur à partir des connaissances de ses voisins.

Pour le routage dans Internet, cette technique a atteint ses limites car les tables des routeurs peuvent contenir de très nombreuses entrées, et donc entraîner des pertes de temps trop importantes pour parcourir ces tables lors de la recherche d'un destinataire.

Routage hiérarchique

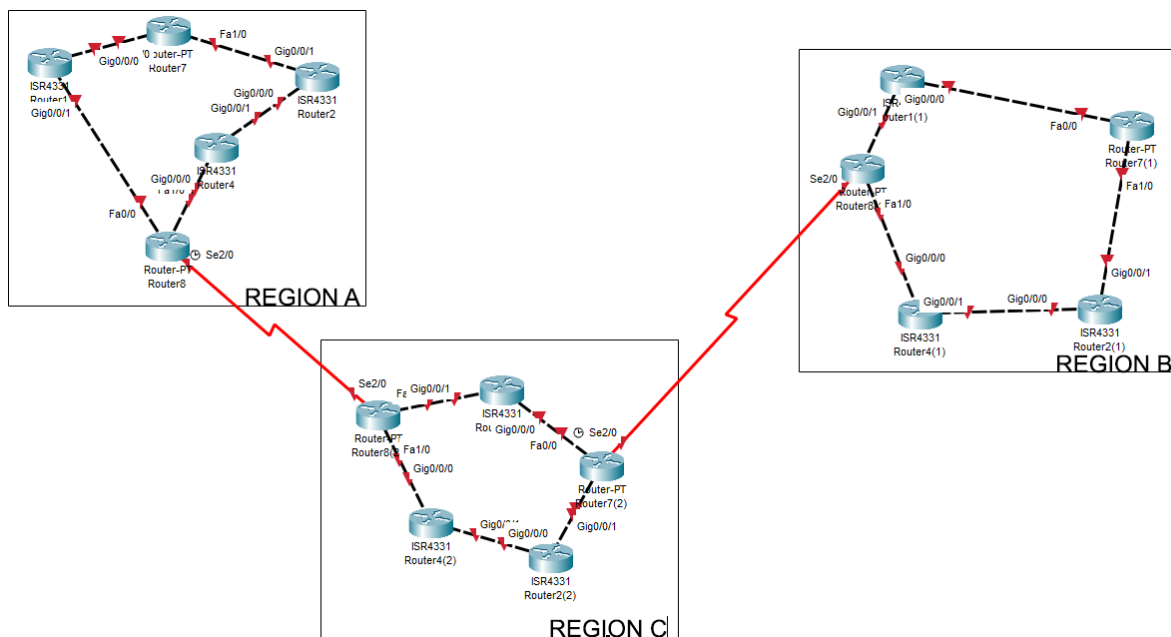
Le routage hiérarchique est basé sur la technique de routage à vecteur de distance, mais une réflexion sur la structure des tables de routage a été menée, dans le but de limiter le nombre d'entrées à consulter lors de la recherche d'un destinataire.

La solution consiste à diviser le réseau en plusieurs zones géographiques appelées régions. Chaque routeur va alors posséder dans sa table trois types de données :

- Les ports de sortie à emprunter pour accéder à chaque destinataire situé dans sa région.
- Les ports de sortie permettant d'accéder à chacune des autres régions du réseau.
- Un port de sortie à utiliser par défaut pour une adresse de destinataire inconnu.

Une telle technique peut être améliorée en mettant en place plusieurs découpages hiérarchiques successifs, c'est-à-dire en divisant chaque zone en plusieurs sous-zones et ceci répété plusieurs fois.

Le routage à vecteur de distance sur lequel est basé l'algorithme de routage RIP utilisé sur Internet intègre cette notion de hiérarchie.



Routage dans les réseaux sans fil

Dans un routage sans fil, une difficulté supplémentaire apparaît pour effectuer le routage : les ordinateurs mobiles se déplacent géographiquement à travers le réseau et sont associés pour une durée limitée à un point d'accès sans fil (on appelle cellule la zone géographique correspondant au point d'accès sans fil).

Le routage adapté à ces changements de cellules nécessite un algorithme particulier. Nous n'en présentons ici qu'un fonctionnement simplifié :

1. Initialement, chaque mobile est associé à un point d'accès de rattachement (sa station de base de rattachement).
2. Lorsqu'un mobile arrive dans une nouvelle cellule, il demande sa connexion à la station de base correspondante. Cette phase de connexion nécessite un échange avec sa cellule de rattachement.
3. La station de base de rattachement enregistre la localisation du mobile dans une table de routage. De même, celle de la nouvelle cellule enregistre l'adresse du mobile dans une table (possédant autant d'entrées qu'il y a de mobiles dans la cellule à cet instant).
4. La communication peut se réaliser : les données destinées au mobile sont envoyées à sa station de base de rattachement, qui les transmet à son tour à la station de base de la cellule où se trouve le mobile à cet instant. Celle-ci, connaissant l'adresse du mobile grâce à sa table, transmet les données au mobile.

Une fois les datagrammes créés, ils sont transmis sur le réseau par la machine émettrice : le protocole IP est alors en charge de les acheminer à travers le maillage du réseau (très complexe lorsque le réseau est étendu comme Internet).

Le protocole IP dispose d'une méthode spécifique de routage : le protocole RIP (Routing Information Protocol).

RIP est un algorithme de routage par sauts successifs (Next-Hop Routing) : cette méthode spécifie qu'un routeur ne connaît pas le chemin que va emprunter un datagramme, mais seulement le routeur suivant à qui le datagramme va être transmis.

Le principe consiste à intégrer à chaque routeur une table de routage proposant le routeur suivant pour chaque destinataire (quelque soit sa nature : machine, réseau, adresse inconnue).

La structure d'une table de routage RIP est simple, comptant 4 champs pour définir une route :

- La **destination** du datagramme est une adresse IP (d'un hôte, d'un réseau ou d'un routeur de sortie par défaut).
- **Le routeur de saut suivant** (passerelle) qui permettra au datagramme d'accéder à un autre réseau (cette adresse est le routeur lui-même si le destinataire est situé sur un réseau directement accessible via une de ses interfaces).
- L'adresse de **l'interface** du routeur à utiliser pour pouvoir accéder au routeur de saut suivant.
- La valeur du **vecteur de distance**, qui correspond au nombre de sauts à effectuer avant d'atteindre le réseau de la machine destinataire du datagramme. On appelle aussi cette valeur la **métrique** de la route.

Adresse de destination	Passerelle	Interface	Vecteur de distance
Adresse 1	Routeur suivant 1	Interface 1	2
Adresse 2	Routeur suivant 2	Interface 1	1
....	1
Autres	Routeur suivant 3	Interface 5	1

L'algorithme utilisé par RIP est relativement simple : il consiste à rechercher dans la table de routage la meilleure route vers le destinataire voulu :

1. Lorsque ce destinataire est connu du routeur, le datagramme lui est transmis directement (si le routeur est directement connecté au réseau de destination) ou au routeur suivant à utiliser.
2. Si le réseau du destinataire est connu du routeur, le datagramme est transmis au routeur suivant à utiliser.

3. Si l'adresse est inconnue, le datagramme est transmis au routeur par défaut (pour la route par défaut, le vecteur de distance est toujours de 1).

Le programme en Python suivant correspond au traitement RIP d'un datagramme reçu par un routeur, pour le réémettre sur la route vers son destinataire :

```
table_de_routage = [
    ["192.200.7.18", "192.200.7.253", 3],
    ["192.200.7.0", "192.200.7.253", 3],
    ["192.190.8.0", "192.200.7.254", 7],
    ["default", "192.200.7.254", 1]
]

trouve = False

adresse_destination = input("Quelle est l'adresse IP Destinataire ? ")

#Nous parcourons la table de routage
for adresse, routeur_saut_suivant, vecteur_de_saut in table_de_routage:
    #Si nous trouvons un chemin vers le destinataire
    if adresse == adresse_destination:
        trouve = True
        #Nous transmettons le datagramme au routeur de saut suivant
        # ou au destinataire s'il est directement accessible
        # envoyer(datagramme, routeur_saut_suivant)
        print("Le datagramme est envoyé à ", routeur_saut_suivant)
    #Si nous n'avons pas trouvé de chemin vers le destinataire
    if trouve == False:
        #Nous parcourons à nouveau la table de routage
        for adresse, routeur_saut_suivant, vecteur_de_saut in table_de_routage:
            #Si nous trouvons un chemin vers le réseau du destinataire
            if adresse == adresse_reseau_destinataire:
                trouve = True
                #Nous transmettons le datagramme au routeur de saut suivant
                # envoyer(datagramme, routeur_saut_suivant)
                print("Le datagramme est envoyé à ", routeur_saut_suivant)
        if trouve == False:
            #Nous parcourons à nouveau la table de routage
            for adresse, routeur_saut_suivant, vecteur_de_saut in table_de_routage:
                if adresse == "default" :
                    trouve = True
                    print("Le datagramme est envoyé à ", routeur_saut_suivant)
        if trouve == False:
            print ("Impossible de transmettre le datagramme")
```

Les routeurs s'échangent les informations contenues dans leurs tables au moyen de messages particuliers appelés messages RIP : à intervalles de temps réguliers (30 secondes), chaque routeur émet un message RIP à destination de ses voisins directs. Un message RIP contient la liste des réseaux connus du routeur émetteurs.

Le protocole TCP est un protocole complémentaire au protocole IP. TCP est associé à IP pour améliorer la qualité de service en mettant en place une transmission fiable en mode connecté.

TCP agit à plusieurs niveaux :

- **Ouverture et fermeture** de la connexion
- Découpage des données reçues des applications en paquets appropriés à la constitution des datagrammes IP (au maximum 65 536 octets) et réassemblage à l'arrivée si nécessaire.
- **Contrôle de la qualité du service** pour conserver un service fiable en mode connecté.
- **Gestion des problèmes de transmission** et reprise en cas d'interruption.

Une fois découpées en paquets, les données sont transmises sur le réseau par le protocole IP, dans les datagrammes IP. Ces datagrammes sont traités individuellement. A la réception ces paquets sont réassemblés puis mis à la disposition des applications.

Si un émetteur envoie plusieurs trames à un même destinataire, toutes n'emprunteront pas forcément le même chemin.

Le support physique n'est pas parfait et des problèmes peuvent survenir au cours de la transmission d'un datagramme entre deux machines. Des méthodes doivent être mises en place pour rendre ces problèmes d'erreurs transparents aux applications.

Deux types de problèmes peuvent se poser au cours d'un échange :

- La détection, à la réception **d'une erreur de transmission** dans le datagramme, doit entraîner un traitement spécifique : correction de l'erreur si une méthode correctrice est disponible ou envoi d'une demande de retransmission dans le cas contraire.
- La perte d'un datagramme doit être détectée et gérée de façon à reconstituer le message émis initialement.

Une méthode de **détection d'erreurs** doit permettre de constater qu'une erreur est apparue dans le datagramme. Elle ne fournit aucun détail sur le nombre d'erreurs, leur localisation, leurs conséquences sur les données... Son seul but est de signaler que le datagramme reçu est différent de celui envoyé, et donc de demander à l'émetteur une retransmission du datagramme endommagé.

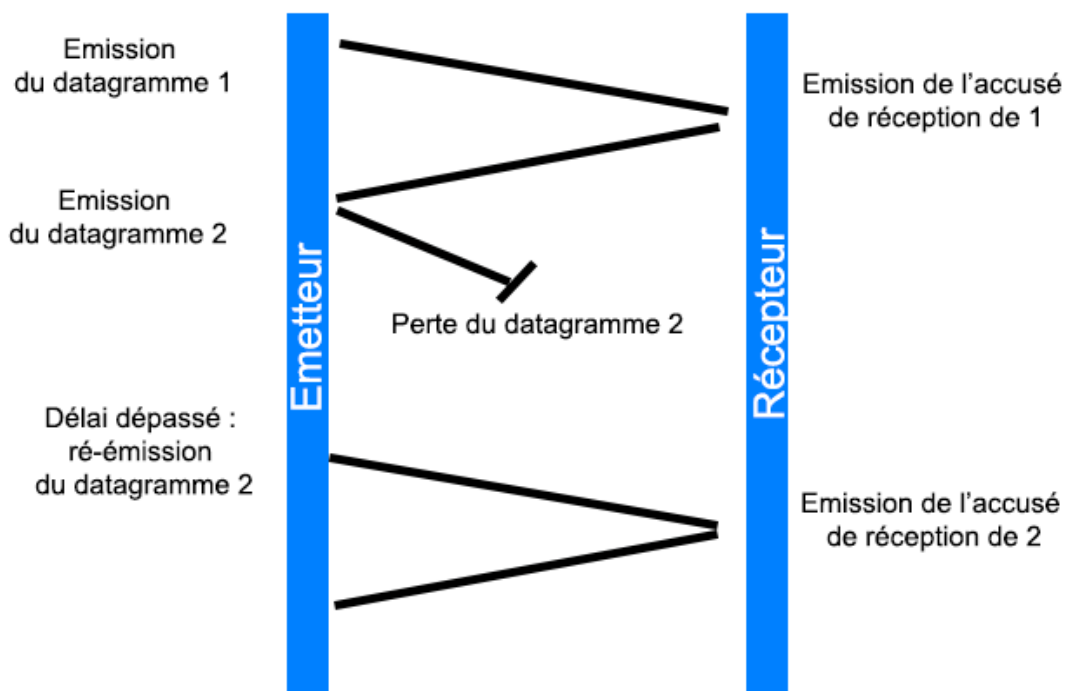
Une méthode de **correction d'erreurs** est beaucoup plus complexe qu'une méthode simplement détectrice. Elle doit transmettre en plus des données tout ce qui est nécessaire à les reconstituer en cas de constat d'erreur à l'arrivée. La complexité de ce type de méthode et la perte de temps importante générée en font des outils peu utilisés dans les réseaux actuels.

La méthode la plus utilisée pour gérer **les pertes de datagramme** dans les échanges entre deux machines est basée sur l'utilisation d'accusés de réception (acquittements).

Le principe de base consiste à faire envoyer un datagramme spécifique par le récepteur, accusant réception du datagramme qu'il vient de recevoir :

- Lorsque l'émetteur reçoit l'accusé de réception du datagramme qu'il a envoyé, il peut émettre le datagramme suivant.
- Si l'émetteur ne reçoit pas d'accusé de réception dans un délai de temps défini, il renvoie ce datagramme.

Concrètement, l'accquittement du datagramme reçu est réalisé par le champ *Numéro d'accquittement* dans un datagramme de réponse retourné à l'émetteur.



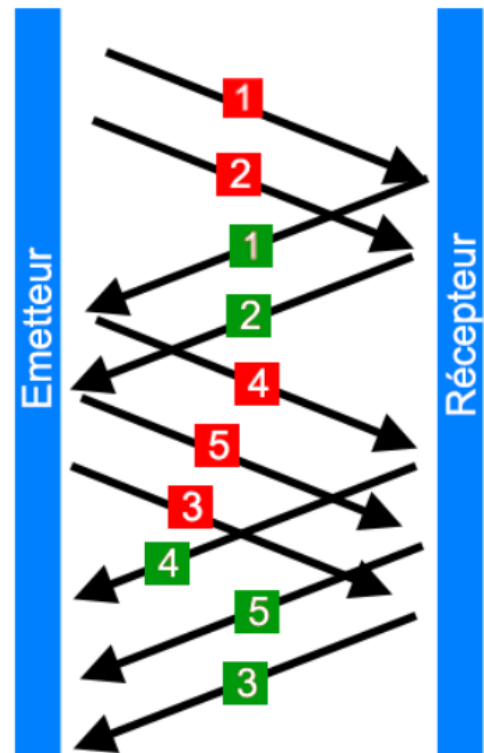
Pour améliorer cette méthode, TCP propose un mécanisme de fenêtre : un nombre défini de datagrammes sont envoyés en continu par l'émetteur sans attendre un accusé de réception. Ce nombre de datagrammes envoyés est ce que l'on appelle la taille de la fenêtre TCP.

Exemple avec une taille de fenêtre définie à 3 : ■ ■ ■

1 2 3 4 5 6 7

1 2 3 4 5 6 7
1 2 3 4 5 6 7

1 2 3 4 5 6 7
1 2 3 4 5 6 7
1 2 3 4 5 6 7



Notons qu'en mode non connecté, c'est le protocole UDP (User Data Protocol) qui est associé à IP pour améliorer la qualité de service.

Les besoins

Pour mettre en place une communication sécurisée, plusieurs techniques sont complémentaires :

- L'authentification des extrémités : cette phase, prérequis indispensable à toute communication, est basée sur un système de clés de sécurité (symétrique ou asymétrique).
- La confidentialité : assurée par un algorithme de chiffrement, qui a pour objectif que le contenu de la communication ne soit pas lisible par un tiers. De même que pour l'authentification, le chiffrement est basé sur un système de clés (symétriques ou asymétriques).
- L'intégrité des données : les deux extrémités de la communication ont la garantie que les données ne sont pas modifiées entre eux.
- La gestion d'autorisations : il est possible d'appliquer des droits aux utilisateurs, selon la stratégie de sécurité définie pour chaque nature de communication.

Principes de chiffrement

Le principe de base de la sécurisation d'une communication consiste à modifier la donnée qui doit être transmise (le chiffrement), de façon à ce que toute personne qui l'intercepterait ne pourrait pas en comprendre le sens. Seul le destinataire va pouvoir retrouver la donnée initiale (le déchiffrement) et la lire.

Les méthodes de chiffrement sont basées sur l'utilisation de clés (des chaînes de caractères numériques) qui vont, par l'application d'algorithmes spécialisés, permettre de chiffrer ou déchiffrer des messages.

Le chiffrement d'un message consiste à le modifier pour le rendre illisible par une personne qui n'y est pas autorisée.

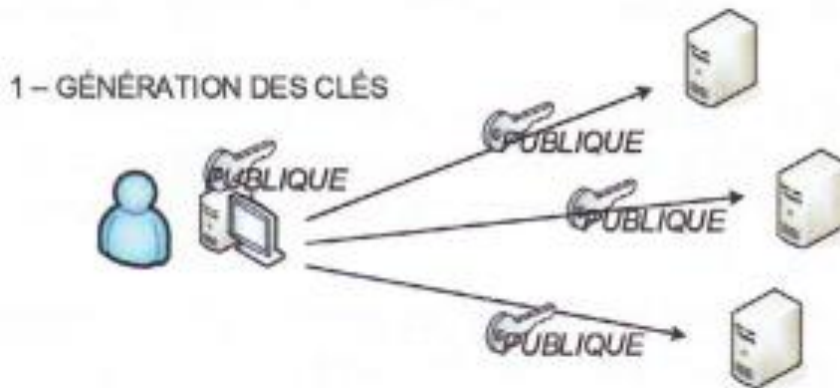
Le chiffrement (puis le déchiffrement) est effectué par l'application au message initial d'une fonction mathématique, basée sur une donnée convenue entre les deux extrémités appelée clé de chiffrement.

Dans ce domaine, deux principales techniques sont disponibles : le chiffrement symétrique ou asymétrique. Les protocoles actuels utilisent selon leur nature l'une ou l'autre ou la combinaison des deux méthodes (HTTPS, le protocole de base du WEB par exemple).

Chiffrement symétrique

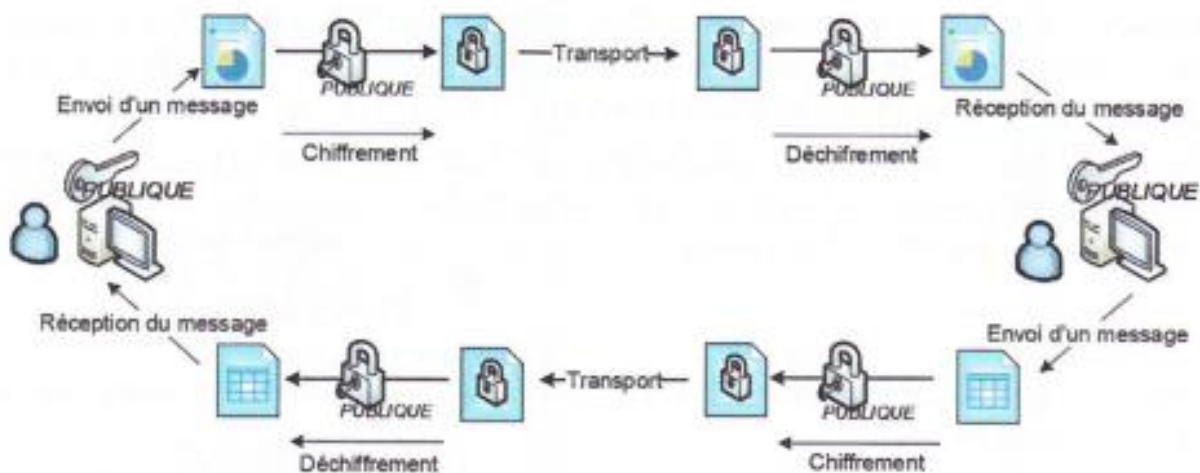
Cette technique consiste à définir une clé de chiffrement, dite clé publique, commune à l'ensemble des interlocuteurs. Cette clé va servir à chiffrer les données lors de leur envoi, et de les déchiffrer à leur réception.

Une phase initiale de définition de la clé publique est nécessaire. Cette clé est ensuite communiquée à tous les ordinateurs susceptibles d'échanger des données.



Chaque ordinateur peut à son tour être émetteur ou récepteur : selon le contexte, il utilise la clé publique pour chiffrer (en émission) ou en déchiffrer (en réception).

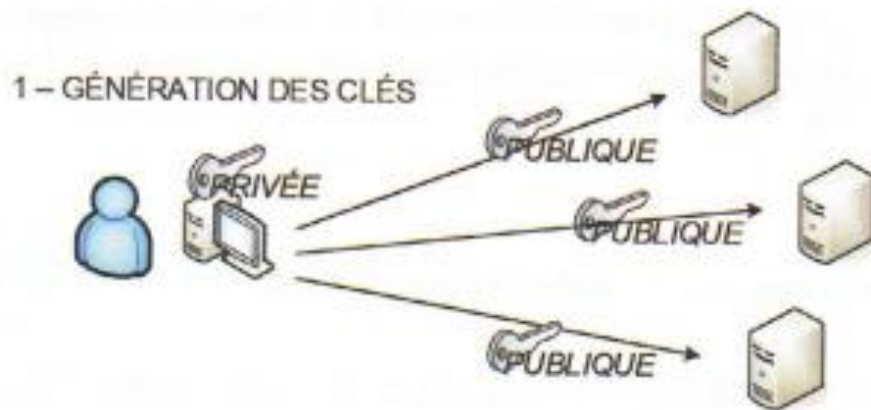
2 – ENVOIS DE MESSAGES



Chiffrement asymétrique

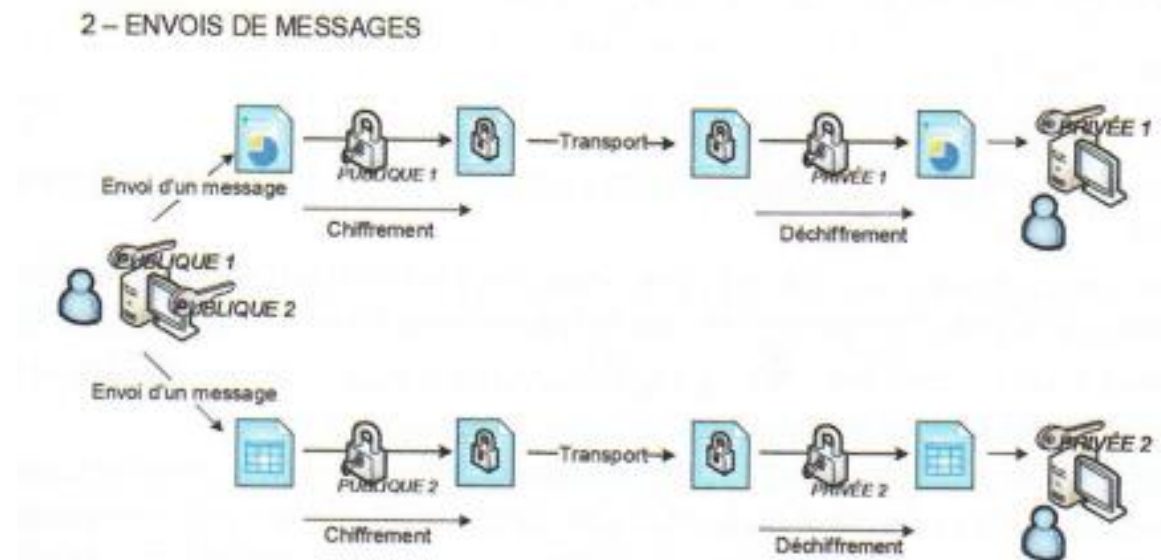
Le principe de chiffrement asymétrique (appelé aussi cryptographie asymétrique ou cryptographie à clé publique) est basé sur l'utilisation de 2 clés :

- Une **clé privée** est définie sur le poste client et stockée sur celui-ci de manière sécurisée.
- Une **clé publique** est diffusée par le client à tous les postes distants.



Le transport sécurisé des données est ensuite assuré par leur chiffrement : la clé publique sert à chiffrer et la clé privée est utilisée pour déchiffrer.

Ce dispositif nécessite qu'un ordinateur possède les clés publiques de tous les postes susceptibles de lui envoyer un message.



Notons que le chiffrement asymétrique est aussi employé pour permettre l'authentification de l'expéditeur d'un message : pour certifier qu'un message provient bien de lui, l'émetteur va utiliser sa clé privée pour chiffrer un message (l'inverse du principe général). Le récepteur le déchiffre grâce à sa clé publique : s'il peut le faire, c'est que ce message a bien été chiffré par l'émetteur, car c'est le seul à détenir la clé privée.

Le concept de signature numérique est basé sur cette technique d'authentification par chiffrement asymétrique.

HTTPS

HTTP est le protocole de base du Web : il est en charge des requêtes d'affichage : les pages Web. HTTP est un protocole non sécurisé : il a donc été nécessaire de lui ajouter des outils assurant la sécurité des transmissions des pages.

Les protocoles SSL (Secure Socket Layer), puis TLS (Transport Layer Security) apportent une couche supplémentaire permettant la sécurisation des échanges de façon transparente pour HTTP.

L'association des protocoles HTTP et TLS porte généralement le nom de HTTPS.

Basés sur un chiffrement asymétrique, SSL et TLS apportent à HTTP la sécurisation nécessaire entre un serveur Web et le navigateur Internet d'un ordinateur client :

- L'échange sécurisé des clés d'authentification du client et du serveur
- La confidentialité des transmissions par le mécanisme de chiffrement.

Autres exemples

SSH

SSH est le protocole de référence de prise de commande à distance sécurisé (Secure SHell).

Le principe de base est la possibilité de se connecter, à partir d'un client SSH, à une autre machine (sur laquelle est installé un serveur SSH) pour exécuter des commandes sur celle-ci.

Le client SSH peut prendre la forme d'une application en ligne de commandes (shell) ou d'une application graphique.

Techniquement, SSH reprend les fonctionnalités communes à toutes les communications sécurisées : authentification (basée sur une méthode de chiffrement asymétrique), confidentialité (plusieurs algorithmes de chiffrement possibles), intégrité des données, autorisations (possibilité de donner des droits différents selon les utilisateurs).

L'infrastructure sans fil Wifi

Le principe d'une infrastructure Wifi est de permettre une communication sans utiliser de support physique matériel, par onde radio.

L'infrastructure est centralisée, autour d'un point d'accès Wifi. Chaque station (ordinateur portable, smartphone, console de jeux, imprimante ...) met en place une connexion avec le point d'accès : l'ensemble des éléments ainsi connectés constitue une cellule Wifi.

L'emploi d'un protocole de chiffrement permet de sécuriser les transmissions de données entre le point d'accès et les stations. Le protocole de chiffrement utilisé actuellement est WPA2 (Wifi Protected Access), basé sur l'utilisation d'une clé de chiffrement publique, connue de tous les éléments autorisés et utilisée pour chiffrer (émission) et déchiffrer (réception) les trames. Le chiffrement est réalisé par un algorithme spécifique appelé AES.



ACTIVITES

Réseaux et sécurité

ACTIVITE 1

Analyser un datagramme IP.

Nous avons utilisé un analyseur de trames pour capturer les datagrammes échangés sur un réseau local. Nous étudions le datagramme suivant :

```
Frame 167: 74 bytes on wire (592 bits), 74 bytes
captured (592 bits) on interface \Device\NPF _
{CFFD652A-7927-4235-8C66-9F399453EEB5}, id 0
Interface id: 0 (\Device\NPF _ {CFFD652A-7927-4235-8C66-
9F399453EEB5})
Interface name: \Device\NPF _ {CFFD652A-7927-4235-8C66-
9F399453EEB5}
Interface description: Wi-Fi
Encapsulation type: Ethernet (1)
Arrival Time: Jan 14, 2020 09:35:03.487495000 Paris,
Madrid
[Time shift for this packet: 0.000000000 seconds)
Epoch Time: 1578990903.487495000 seconds
[Time delta from previous captured frame: 0.004513000
seconds) [Time delta from previous displayed frame:
0.004513000 seconds) [Time since reference or first
frame: 15.057563000 seconds) Frame Number: 167
Frame Length: 74 bytes (592 bits)
Capture Length: 74 bytes (592 bits)
[Frame is marked: False)
[Frame is ignored: False)
[Protocols in frame: eth :ethertype: ip: icmp :data)
[Coloring Rule Name: ICMP)
[Coloring Rule String: icmp 11 icmpv6)
Ethernet II, Src: Azurewav _ 68:77:ed
(f0:03:Sc:68:77:ed), Dst: Ubiquiti 3c:45:af (24: a4: 3c:
3c :45 :af)
Internet Protocol Version 4, Src: 172.1.0.102, Dst:
192.168.1.1
0100 .... = Version: 4
.... 0101 = Header Length: 20 bytes (5)
Differentiated Services Field: 0x00 (DSCP: CS0, ECN:
Not-ECT) Total Length: 60
Identification: 0x47ef (18415)
Flags: 0x0000
... 0 0000 0000 0000 = Fragment offset: 0
Time to live: 128
```

Protocol: ICMP (1)
 Header checksum: 0x84c1 [validation disabled] [Header checksum status: Unverified]
 Source: 172.1.0.102
 Destination: 192.168.1.1
 Internet Control Message Protocol
 Type: 8 (Echo (ping) request) Code: 0
 Checksum: 0x4dl9 [correct]
 [Checksum Status: Good]
 Identifier (BE): 1 (0x0001)
 Identifier (LE): 256 (0x0100) Sequence number (BE): 66 (0x0042) Sequence number (LE): 16896 (0x4200) [No response seen]
 Data (32 bytes)

- Quelle est la nature du réseau utilise ?
- Extraire l'adresse IP de l'émetteur et celle du destinataire de ce datagramme.
- Quelle est l'application qui a génère ce datagramme ?

ACTIVITE 2

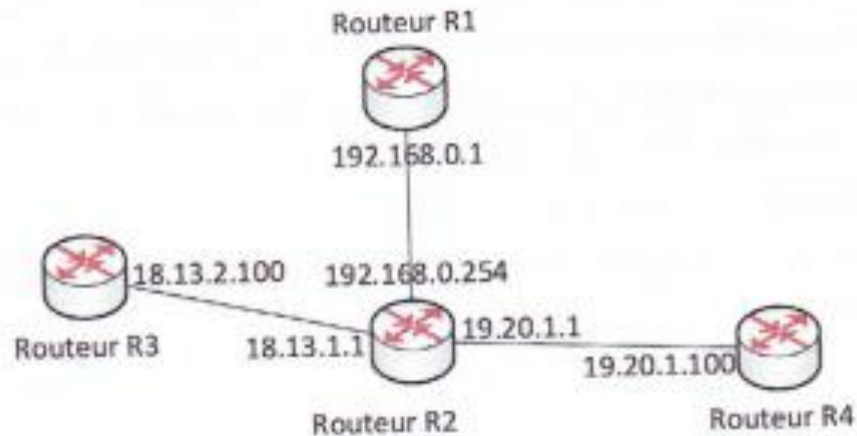
Un routeur a la table de routage suivante

Adresse de destination	Passerelle	Interface	Vecteur de distance
192.8.13.20	192.168.1.254	192.168.1.3	3
192.168.1.0	192.168.1.254	192.168.1.3	1
180.18.0.0	180.18.1.254	180.18.1.1	1
180.19.0.0	180.19.1.254	180.18.1.1	2
180.19.3.0	180.19.1.254	180.18.1.1	2
Defaut	192.168.1.254	192.168.1.3	1

Donner le message RIP émis par ce routeur

ACTIVITE 3

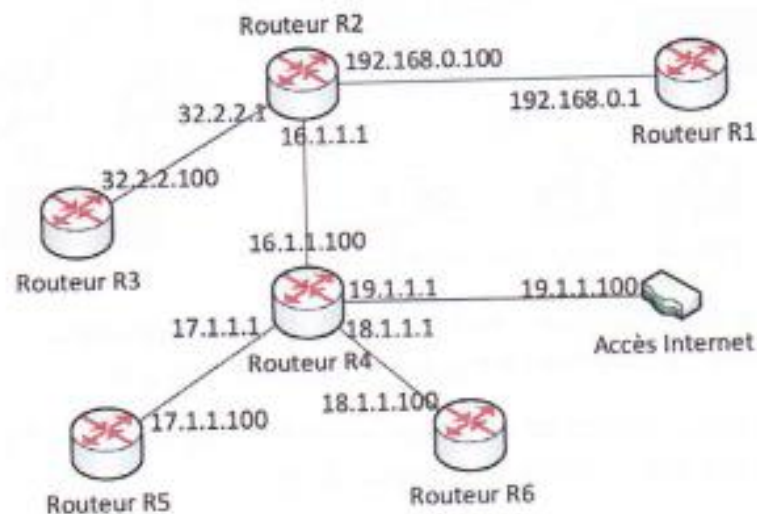
Soit le réseau suivant :



Donner la table de routage RIP du routeur R1.

ACTIVITE 4

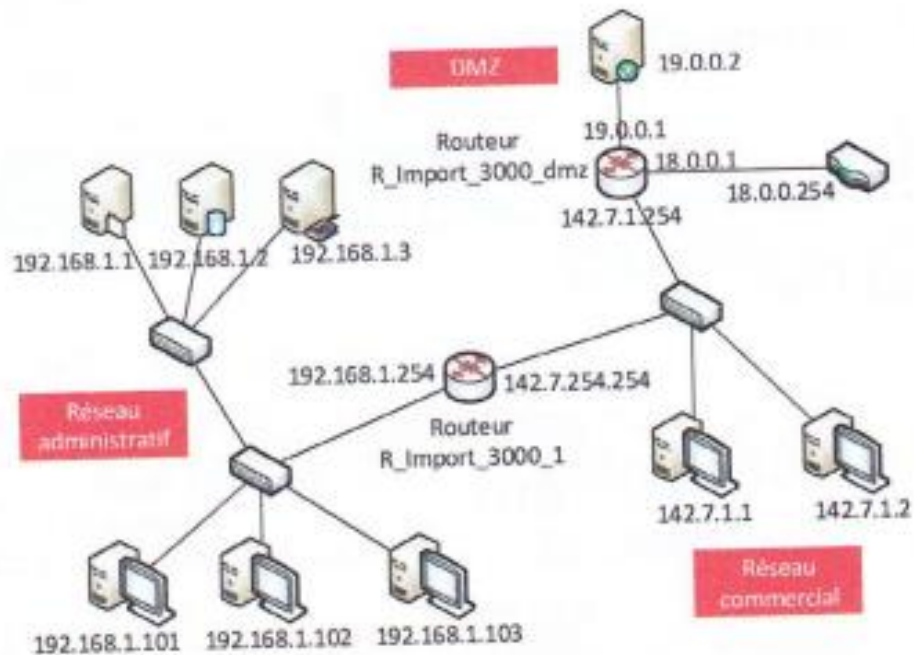
Soit le réseau suivant :



1. Expliquer comment, lorsqu'il reçoit un datagramme sur l'une de ses interfaces, le routeur R2 retransmet ce datagramme en fonction de son destinataire.
2. Donner la table de routage RIP du routeur R2.
3. Donner la table de routage RIP du routeur R4.
4. Donner la table de routage RIP du routeur R6.

La société Import3000 est spécialisée dans l'import de produits numériques et dans leur revente sur le marché français. Son réseau informatique est structuré en 3 parties :

- Le réseau administratif abritant tous les postes de travail et les serveurs de fichiers et de gestion (réseau 192.168.1.0)
- Le réseau commercial (réseau 142.7.0.0)
- La zone démilitarisée (DMZ) hébergeant les serveurs Web accessibles par Internet (réseau 19.0.0.0)



1. Donner la ligne de la table de routage d'un hôte du réseau administratif nécessaire pour qu'il puisse joindre tout hôte du réseau commercial.
2. Donner la ligne de la table de routage de cet hôte du réseau administratif nécessaire pour qu'il puisse joindre le serveur Web.
3. Donner la ligne de la table de routage d'un hôte du réseau commercial nécessaire pour qu'il puisse joindre le serveur Web.



ACTIVITE 6

Sur un serveur Linux, la commande qui permet d'afficher la table de routage est **route**.

Dans cet exercice, nous exécutons **route** sur un serveur en activité, le résultat est le suivant :

<i>Destination</i>	<i>Passerelle</i>	<i>Genmask</i>	<i>Indic</i>	<i>Metric</i>	<i>Ref</i>	<i>Use</i>	<i>Iface</i>
192.169.1.36	0.0.0.0	255.255.255.255	UH	0	0	0	eth0
192.169.1.0	0.0.0.0	255.255.255.0	U	1	0	0	eth0
195.1.1.0	0.0.0.0	255.255.255.0	U	1	0	0	eth1
70.0.0.0	0.0.0.0	255.0.0.0	U	3	0	0	eth1
127.0.0.0	0.0.0.0	255.0.0.0	U	0	0	0	lo
default	192.169.1.254	0.0.0.0	UG	1	0	0	eth0

A combien de réseaux ce routeur est-il relié ?

CORRECTION DES EXERCICES



Activités

Réseaux et sécurité

ACTIVITE 1

1. Dans la première partie, l'analyseur décrit les niveaux physique et liaison de données de la transmission : nous en extrayons les lignes intéressantes :

```
Interface id: 0 (\Device\NPF_{CFFD652A-7927-4235-8C66-9F399453EEB5})
Interface name: \Device\NPF_{CFFD652A-7927-4235-8C66-9F399453EEB5}
Interface description: Wi-Fi
Encapsulation type: Ethernet (1)
```

La 2e ligne nous indique l'interface qui est utilisée, et la 3e ligne nous précise que c'est une interface Wifi.

La 4e ligne indique que c'est un datagramme Ethernet, norme utilisée pour les réseaux locaux.

2. Dans la partie concernant le protocole réseau, nous extrayons les lignes suivantes :

```
Internet Protocol Version 4, Src: 172.1.0.102, Dst: 192.168.1.1
0100 .... = Version: 4
Source: 172.1.0.102
Destination: 192.168.1.1
```

En étudiant ces caractéristiques, nous lisons à la 2e ligne que le protocole de communication est Pv4. Les 3e et 4e lignes nous indiquent que l'adresse de l'émetteur (Source) est 172.1.0.102, et que celle du destinataire (Destination) est 192.168.1.1.

3. Au niveau supérieur, correspondant aux couches applicatives, nous trouvons les caractéristiques suivantes :

```
Internet Control Message Protocol
Type: 8 (Echo (ping) request)
```

Ce datagramme est une requête ping (basée sur le protocole ICMP d'administration de réseau) lancée par l'émetteur pour savoir s'il peut joindre le destinataire.

ACTIVITE 2

Un message RIP est émis par chaque routeur pour transmettre à ses voisins la liste des destinataires (hôtes ou réseaux) pour lesquels il possède une route dans sa table.

La 1^{re} ligne définit la route vers l'hôte d'adresse IP 192.8.13.20 (adresse de classe C, avec 192.168.1 pour l'identifiant du réseau et 20 pour l'identifiant de l'hôte dans ce réseau).

La 2^e ligne définit la route vers le réseau 192.168.1.0 (réseau de classe C).

La 3^e ligne définit la route vers le réseau 180.18.0.0 (réseau de classe B).

La 4^e ligne définit la route vers le réseau 180.19.0.0 (réseau de classe B).

La 5^e ligne définit la route vers l'hôte d'adresse IP 180.19.3.0 (adresse de classe B, avec 180.19 pour l'identifiant du réseau et 3.0 pour l'identifiant de l'hôte dans ce réseau). Cet hôte fait partie du réseau 180.19.0.0 pour lequel la route est déjà connue (4^e ligne) : cette ligne ne sera pas reprise dans le message RIP car la route vers ce : hôte est déjà comprise dans une autre ligne du message RIP.

Le message RIP émis par le routeur est :

Destinataire	Vecteur de distance
192.8.13.20	3
192.168.1.0	1
180.18.0.0	1
180.19.0.0	2

ACTIVITE 3

La table de routage liste les routes d'accès à chaque réseau, chacune de ces routes étant définie par 4 champs.

Un routeur n'a pas obligatoirement de route vers chaque réseau, mais une route par défaut permet de transmettre les datagrammes destinés à des réseaux pour lesquels il ne connaît pas de route.

En reprenant les réseaux présents dans le schéma, une table de routage de R1 peut-être :

Adresse de destination	Passerelle	Interface	Vecteur de distance
192.168.0.0	192.168.0.1	192.168.0.1	1
18.13.0.0	192.168.0.254	192.168.0.1	1
19.20.0.0	192.168.0.254	192.168.0.1	1
défaut	192.168.0.254	192.168.0.1	1

ACTIVITE 4

1. Lorsque le routeur R2 reçoit un datagramme sur l'une de ses interfaces, il l'analyse pour en extraire le destinataire. R2 recherche ensuite dans sa table de routage s'il connaît une route vers ce destinataire (le destinataire lui-même ou le réseau qui le contient), puis retransmet le datagramme au routeur de saut suivant défini dans cette route, par son interface elle aussi définie dans la route.

Listons les possibilités de routage au niveau de R2, à partir des réseaux que nous trouvons sur le schéma :

- Les trames destinées au réseau 192.168.0.0 seront retransmises à la passerelle 192.168.0.100, c'est-à-dire sa propre interface 192.168.0.100, et ce réseau sera accessible directement sur cette interface (vecteur de saut à 1).
 - Les trames destinées au réseau 32.0.0.0 seront retransmises à la passerelle 32.2.2.1, c'est-à-dire sa propre interface 32.2.2.1, et ce réseau sera accessible directement sur cette interface (vecteur de saut à 1).
 - Les trames destinées au réseau 16.0.0.0 seront retransmises à la passerelle 16.1.1.1, ici aussi sa propre interface 16.1.1.1 pour un réseau accessible directement sur cette interface.
 - Les trames destinées au réseau 17.0.0.0 seront retransmises à la passerelle 16.1.1.100, par son interface 16.1.1.1, et ce réseau sera accessible par 1 saut de routeur.
 - De même, les trames destinées au réseau 18.0.0.0 et au réseau 19.0.0.0 seront retransmises à la passerelle 16.1.1.100, par son interface 16.1.1.1, et ces réseaux seront accessibles chacun par 1 saut de routeur.
 - Les trames dont le destinataire n'est pas listé dans la table (destinataire défaut) seront retransmises à la passerelle 16.1.1.100, par son interface 16.1.1.1. (Pour la route par défaut, le vecteur de distance est toujours spécifié à 1).
2. Nous devons proposer une route vers chaque réseau, et ajouter une route par défaut qui sera utilisée pour émettre tous les datagrammes dont le destinataire n'est pas listé dans la table.
Dans un premier temps, nous proposons une table exhaustive, c'est-à-dire qui liste la totalité des destinataires du schéma, nous la simplifierons ensuite.

Une table de routage de R2 peut donc être

Adresse de destination	Passerelle	Interface	Vecteur de distance
192.168.0.0	192.168.0.100	192.168.0.100	1
32.0.0.0	32.2.2.1	32.2.2.1	1
16.0.0.0	16.1.1.1	16.1.1.1	1
17.0.0.0	16.1.1.100	16.1.1.1	1
18.0.0.0	16.1.1.100	16.1.1.1	1
19.0.0.0	16.1.1.100	16.1.1.1	1
default	16.1.1.100	16.1.1.1	1

Dans un second temps, nous simplifierons la table en regroupant en une seule ligne les routes qui utilisent la même passerelle.

La table de routage ci-dessus devient

Adresse de destination	Passerelle	Interface	Vecteur de distance
192.168.0.0	192.168.0.100	192.168.0.100	1
32.0.0.0	32.2.2.1	32.2.2.1	1
16.0.0.0	16.1.1.1	16.1.1.1	1
default	16.1.1.100	16.1.1.1	1

Pour simplifier la table de routage et diminuer le nombre de routes qu'elle contient, est intéressant de supprimer des routes qui sont communes

- plusieurs routes peuvent être regroupées par une seule ligne lorsqu'elles sont accessibles par le même routeur
- si certains réseaux connus sont accessibles par l'adresse défaut, il n'est pas nécessaire de les lister.

3. D'après le schéma du réseau, une table de routage exhaustive de R4 peut être :

Adresse de destination	Passerelle	Interface	Vecteur de distance
192.168.0.0	16.1.1.1	16.1.1.100	1
32.0.0.0	16.1.1.1	16.1.1.100	1
16.0.0.0	16.1.1.1	17.1.1.1	1
17.0.0.0	17.1.1.1	17.1.1.1	1
18.0.0.0	18.1.1.1	18.1.1.1	1
19.0.0.0	19.1.1.1	19.1.1.1	1
default	19.1.1.100	19.1.1.1	1

Cette table présente une seule route commune : pour les réseaux 192.168.0.0 et 32.0.0.0). Il n'existe pas de manière de regrouper en un seul réseau ces deux réseaux (de classe différente, non consécutifs), donc cette simplification n'est pas possible, et la table ne peut pas être réduite.

4. Après étude du schéma, une table de routage de R6 peut être :

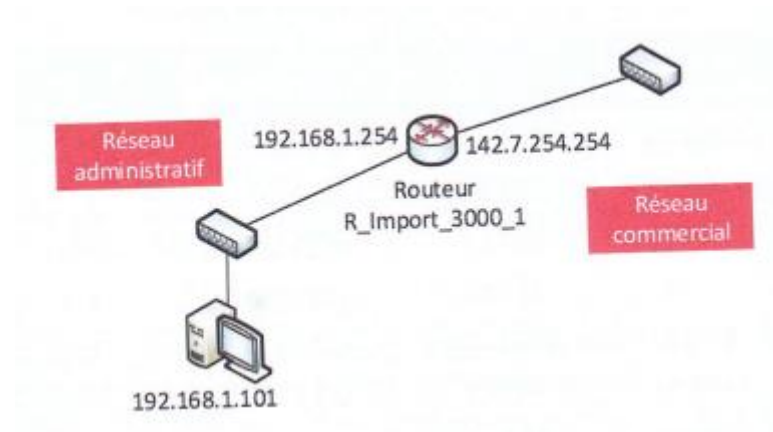
Adresse de destination	Passerelle	Interface	Vecteur de distance
192.168.0.0	18.1.1.1	18.1.1.100	2
32.0.0.0	18.1.1.1	18.1.1.100	2
16.0.0.0	18.1.1.1	18.1.1.100	1
17.0.0.0	17.1.1.1	17.1.1.100	1
18.0.0.0	18.1.1.100	18.1.1.100	1
19.0.0.0	18.1.1.1	18.1.1.100	1
default	18.1.1.1	18.1.1.100	1

Nous essayons ensuite de simplifier si possible cette table. A l'exception de l'accès au réseau 18.0.0.0, directement accessible sur l'une des interfaces du routeur R6, toutes les autres routes sont définies en utilisant le routeur R4 comme routeur de saut suivant (tous les datagrammes sont retransmis à la passerelle 18.1.1.1) : ces routes peuvent être regroupées en une seule, et la table de routage de R6 peut devenir :

Adresse de destination	Passerelle	Interface	Vecteur de distance
18.0.0.0	18.1.1.100	18.1.1.100	1
default	18.1.1.1	18.1.1.100	1

ACTIVITE 5

1. Nous extrayons de l'architecture la partie concernant l'interconnexion des deux réseaux (administratif et commercial)



Nous choisissons arbitrairement l'hôte 192.168.1.101. Dans sa table de routage, nous ajoutons l'entrée correspondant au réseau commercial : pour joindre ce réseau 142.7.0.0, l'hôte 192.168.1.101 émet ses datagrammes à destination du routeur R_Import_3000_1 d'adresse 192.168.1.254 via son interface 192.168.1.101.

Adresse de destination	Passerelle	Interface	Vecteur de distance
142.7.0.0	192.168.1.254	192.168.1.101	1

2. Dans la table de routage de notre hôte 192.168.1.101 nous devons ajouter l'entrée qui permet d'accéder à la DMZ. La seule route possible est de passer ici aussi par le routeur de saut R_import_3000_1 d'adresse 192.168.1.254

Adresse de destination	Passerelle	Interface	Vecteur de distance
19.0.0.0	192.168.1.254	192.168.1.101	2

3. Si nous considérons par exemple l'hôte 142.7.1.1 du réseau commercial, nous devons ajouter dans sa table de routage la route d'accès à la DMZ, en passant par le routeur R_Import_3000_1dmz d'adresse 142.7.1.254

Adresse de destination	Passerelle	Interface	Vecteur de distance
19.0.0.0	142.7.1.254	142.7.1.1	1

ACTIVITE 6

Trois réseaux sont reliés à ce routeur : le réseau 192.169.1.0, le réseau 195.1.1.0 et le réseau 70.0.0.0.

Le destinataire 192.169.1.36 correspond à l'interface locale, le destinataire 127.0.0.0 est l'adresse locale de rebouclage et la route par défaut définit la passerelle de sortie pour les datagrammes dont le destinataire n'est pas liste.

Exercice 1 :

HTTP est le protocole de base du WEB : c'est lui qui transmet les requêtes de pages Web et assure le transport de ces pages Web entre le serveur et le client, pour que le navigateur de celui-ci puisse les afficher.

Les transferts générés par http ne sont pas sécurisés : il a donc été nécessaire de lui ajouter des outils assurant la sécurité des transmissions des requêtes et pages.

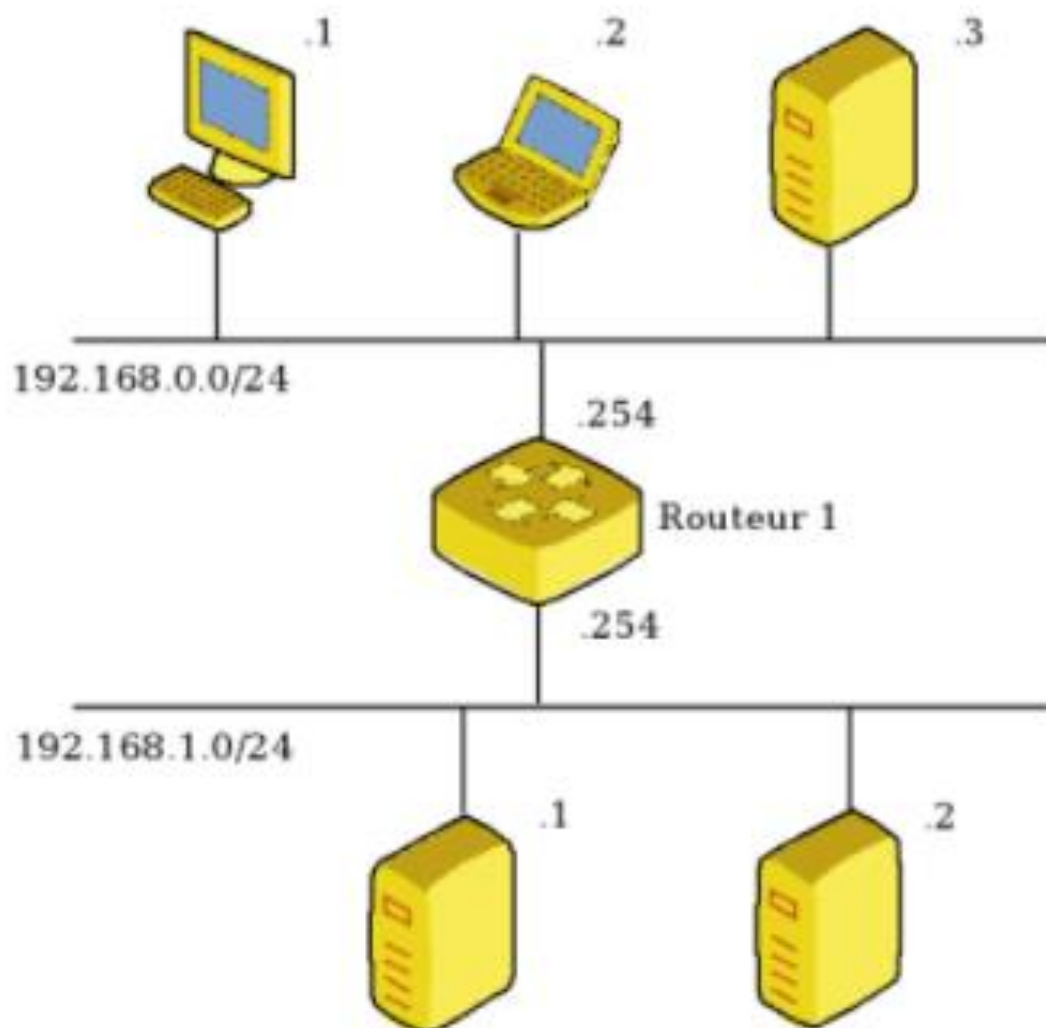
La sécurité a été ajoutée à http par les protocoles SSL, puis TLS, donnant un complexe qui a pris le nom d'HTTPS.

HTTPS intègre la sécurité aux différents niveaux d'un échange, par l'utilisation des techniques de chiffrement :

- L'échange sécurisé de clés,
 - L'authentification du client et du serveur,
 - La confidentialité des transmissions (requêtes et pages) par un mécanisme de chiffrement.
-
1. Sachant que HTTPS assure la confidentialité des données par un chiffrement symétrique, représenter par un schéma la transmission sécurisée de la requête d'une page WEB d'un client à destination d'un serveur WEB.
 2. Ajouter à ce schéma la transmission de la page Web du serveur vers le client.
 3. Nous avons dit que la clé publique utilisée par le client et le serveur pour cette transmission des pages est générée au départ de l'échange par le serveur, puis transmise au client.
Quelle est la problématique qui se pose à ce niveau
 4. Proposer une solution pour sécuriser cette transmission de la clé publique de chiffrement symétrique.
 5. Compléter le schéma de la question 2. En intégrant la diffusion de la clé publique symétrique.

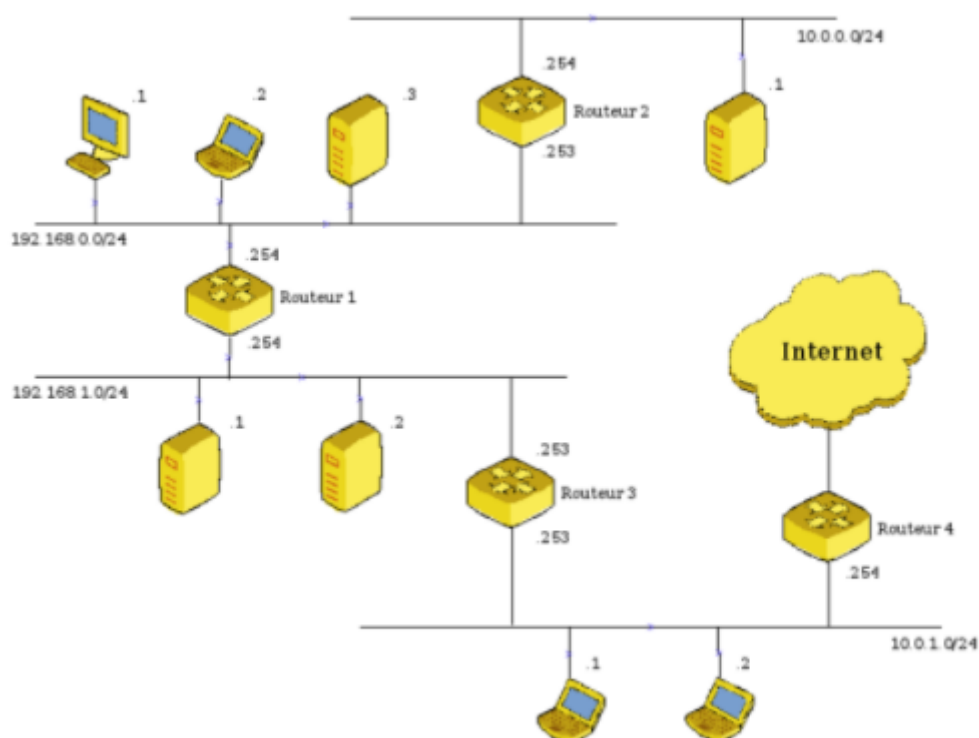
Exercice 2 :

Dresser la table de routage du Routeur 1



Exercice 3 :

Dresser la table de routage des Routeur 1 , Routeur 2, Routeur 3 et Routeur 4.



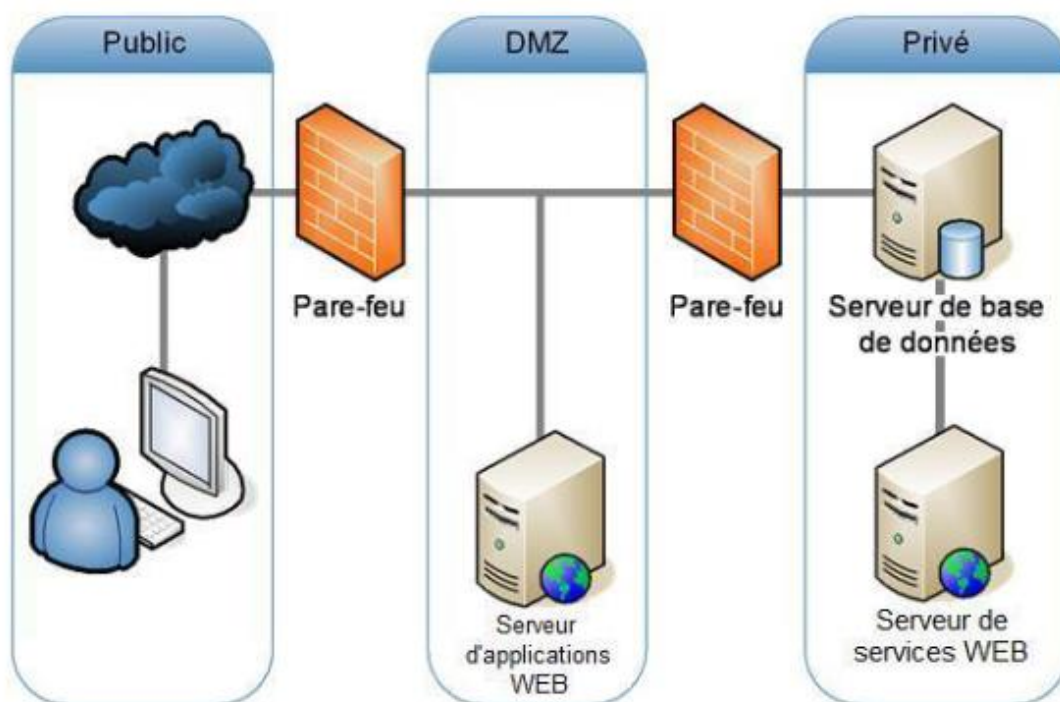
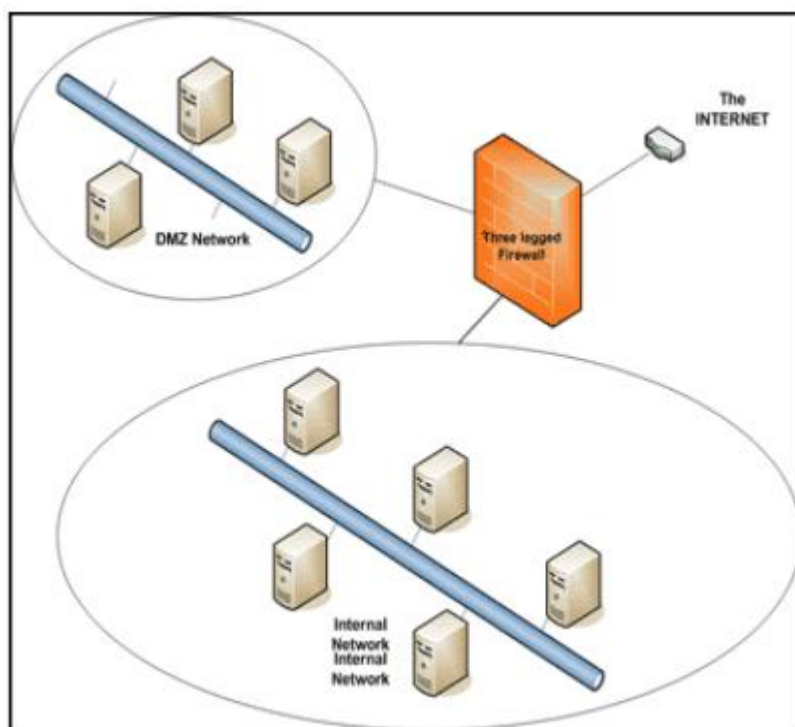
Notion de cloisonnement

Les systèmes pare-feu (firewall) permettent de définir des règles d'accès entre deux réseaux. Dans la pratique, les entreprises ont généralement plusieurs sous-réseaux avec des politiques de sécurité différentes. C'est la raison pour laquelle il est nécessaire de mettre en place des architectures de systèmes pare-feux permettant d'isoler les différents réseaux de l'entreprise. On parle ainsi de « cloisonnement des réseaux » (le terme isolation est parfois également utilisé)

Architecture DMZ

Lorsque certaines machines du réseau interne ont besoin d'être accessibles de l'extérieur (serveur web, un serveur de messagerie, un serveur FTP public, etc.), il est nécessaire de créer un réseau à part, accessible aussi bien du réseau interne que de l'extérieur, sans pour autant risquer de compromettre la sécurité de l'entreprise. On parle ainsi de « zone démilitarisée » (notée DMZ pour DeMilitarized Zone) pour désigner cette zone isolée, hébergeant des applications mises à disposition du public. La DMZ fait ainsi office de « zone tampon » entre le réseau à protéger et le réseau hostile.

Deux exemples d'architectures :



La politique de sécurité mise en oeuvre sur la DMZ est généralement la suivante :

- Traffic du réseau externe vers la DMZ autorisé ;
- Traffic du réseau externe vers le réseau interne interdit ;
- Traffic du réseau interne vers la DMZ autorisé ;
- Traffic du réseau interne vers le réseau externe autorisé ;

La DMZ possède donc un niveau de sécurité intermédiaire, mais son niveau de sécurisation n'est pas suffisant pour y stocker des données critiques de l'entreprise.

Il est à noter qu'il est possible de mettre en place des DMZ en interne afin de cloisonner le réseau selon différents niveaux de protection et ainsi avoir des accès allant du moins sécurisé au plus sécurisé