Administration des Systèmes et Réseaux

LES TUNNELS INFORMATIQUES

Auteur: Bernard GIACOMONI - Autoentreprise GIACOMONI Bernard

Version	Date	Objet
1.0	23/11/2019	Version initiale

Table des matières

I.PRÉSENTATION :

En informatique de réseau, un TUNNEL est un MÉCANISME qui, mis en place entre deux nœuds N1 et N2 d'un RÉSEAU INFORMATIQUE, permet de masquer pour le reste de ce réseau certaines des informations échangées : adresses IP, numéros de ports, paramètres de connexion, charge utile, etc.

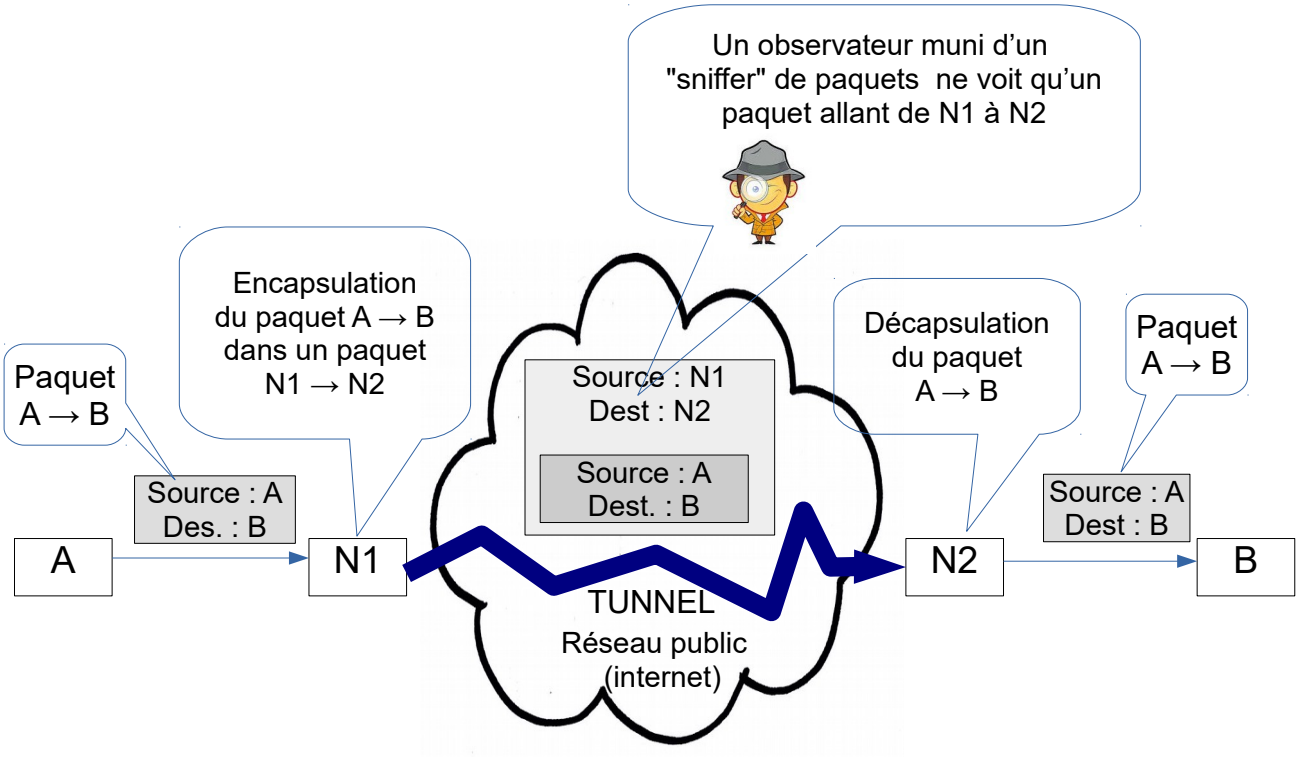
REMARQUE: *En informatique de réseau, la CHARGE UTILE (ou PAYLOAD) est l'ensemble des données d'un paquet qui correspond au MESSAGE (ou à un fragment de message si le message est fragmenté) échangé entre les systèmes émetteurs et destinataires. Les autres données du paquet sont les différentes couches de protocole.*

II.GÉNÉRALITÉS SUR LES TUNNELS:

II.1.PRINCIPE GÉNÉRAL:

Il consiste à encapsuler les PAQUETS de données passant par deux nœuds N1 et N2 du réseau dans d'autres PAQUETS émis entre N1 et N2. Pendant le trajet de N1 vers N2, tout se passe comme si les paquets encapsulés traversaient un TUNNEL qui les dissimulait aux observateur extérieurs.

EXEMPLE : Le schéma suivant détaille le fonctionnement d'un tunnel implanté au niveau de la couche de routage (tunnel IP).



COMMENTAIRES SUR LE SCHÉMA :

- A l'entrée du tunnel, le système N1 récupère les données du paquet A → B (payload et couches de protocole), les crypte éventuellement, puis les encapsule dans un autre paquet;
- Le système N1 émet alors ce nouveau paquet vers N2: pendant le trajet entre N1 et N2, les informations contenues dans le paquet A → B sont donc masquées par les couches de protocole du paquet N1 → N2;
- A la sortie du tunnel, N2 décapsule le paquet A → B puis le transmet vers son destinataire réel (B) .

DISCUSSION :

Cette encapsulation permet de cacher à d'éventuels observateurs situés sur le réseau public certaines informations comme les adresses IP d'origine et de destination, les numéros de ports et les paramètres de connexion des paquets d'origine ;

Cependant, cette dissimulation ne résiste évidemment pas à une analyse approfondie (Deep Packets Inspection) du contenu de la partie PAYLOAD (à l'aide d'un "sniffer" de paquets, par exemple). Pour éviter ce type d'intrusion, il faut CRYPTER les paquets encapsulés, ce qui permet de dissimuler leur contenu aux observateurs du réseau public même en cas d'analyse approfondie.

II.2.DIFFÉRENTS TYPES DE TUNNELS :

Les différentes solutions techniques de "tunnelisation" se distinguent les unes des autres par les "couche de communication" dans lesquelles les mécanismes sont implantés. On trouvera ainsi essentiellement :

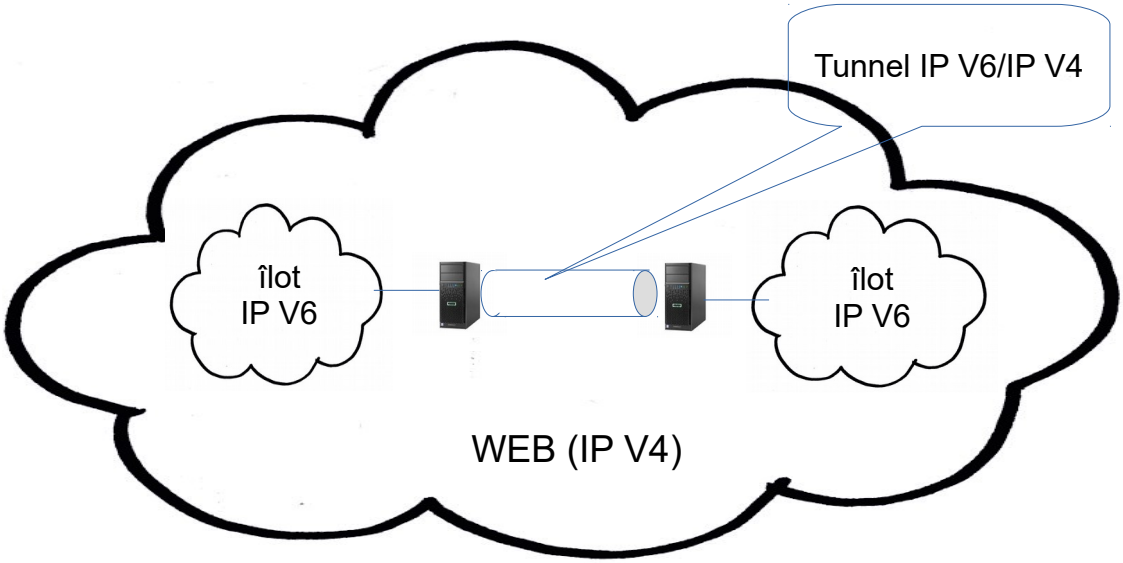
- Des tunnels de niveau applicatif, qui sont établis au niveau des applications communicantes (clients et serveur) ;
- Des tunnels IP qui sont établis au niveau des couches "réseau" des machines communicantes (tunnels IP) ;
- Des tunnels établis au niveau des couches "liaisons" (encapsulation des protocoles de liaison point à point).

II.3.CAS D'UTILISATION COURANTS:

II.3.1.LIAISON ENTRE DEUX ZONES IP V6 :

Actuellement (2019), la plus grande partie du WEB ne peut encore traiter que des échanges en protocole IP V4. Le déploiement du protocole IP V6 progresse, mais son

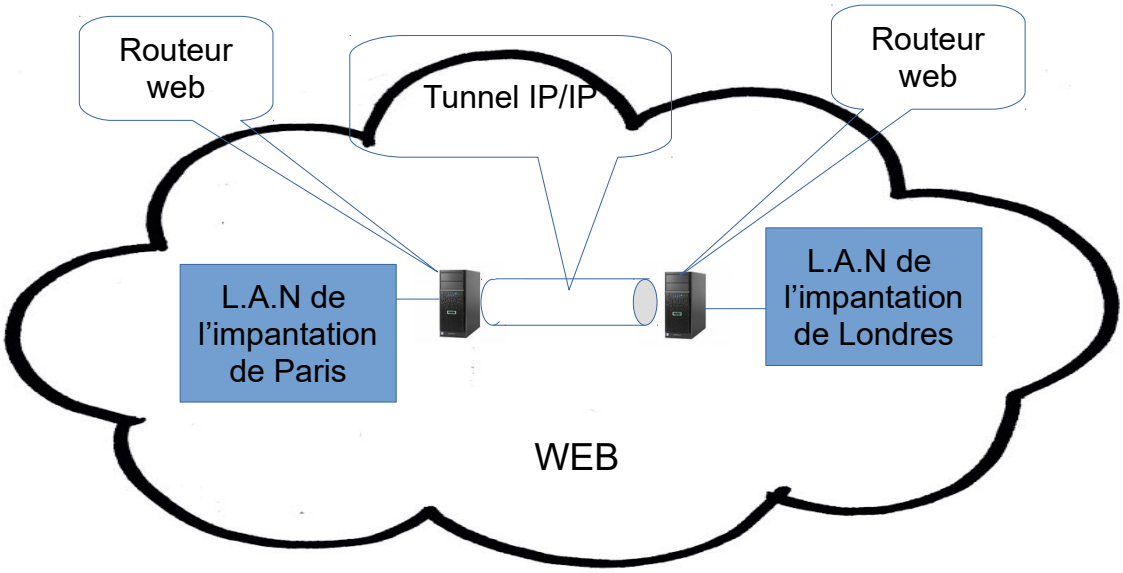
achèvement n'est prévisible qu'à une échéance encore lointaine (plusieurs dizaines d'années). Le WEB se présente donc comme un "océan IP V4" parsemé "d'îlots IP V6". Un tunnel IP permet de relier deux de ces "îlots" :



REMARQUE : dans ce cas, les extrémités du tunnel peuvent être des routeurs sur lesquels les mécanismes de "tunnellisation" (encapsulation/décapsulation, cryptage/décryptage) sont implantés.

II.3.2.LIAISON SÉCURISÉE ENTRE LES RÉSEAUX INTERNES DE DEUX IMPLANTATIONS :

Le schéma ci-dessus peut se transposer à la liaison sécurisée entre deux implantations éloignées d'une même entreprise :



Dans ce cas, la "tunnellisation" permet d'éviter l'espionnage et le piratage des informations de l'entreprise.

II.3.3.ADMINISTRATION A DISTANCE D'UN POSTE INFORMATIQUE:

Les administrateurs de systèmes sont souvent amenés à intervenir sur des postes informatiques distants en émulant des "terminaux à distance" sur le poste qu'ils utilisent en local. Ce type de fonctionnement est rendu possible grâce à l'utilisation de logiciels communicant suivant le modèle "Client/serveur" : le SERVEUR, installé dans le poste à administrer est interrogé par un CLIENT installé sur le poste de l'administrateur. Ces logiciels communiquent grâce à des protocoles spécifiques tels que telnet, rlogin, ssl, etc.

Une connexion SSH peut être ouverte à distance sur un hôte muni d'un SERVEUR SSH grâce aux "CLIENTS SSH" inclus dans les systèmes d'exploitation ou encore à des logiciels clients comme PUTTY. Le protocole SSH permet ainsi d'ouvrir un terminal à distance en utilisant une communication SÉCURISÉE. En effet, les paquets échangés grâce au protocole SSH sont CRYPTÉS grâce à une clef symétrique NÉGOCIÉE A LA CONNEXION.

Un tunnel SSH est obtenu en redirigeant le trafic entre un serveur et un client vers un flux SSH établi entre ces deux machines (c'est le Port Forwarding, offert par le protocole SSH) Ce type de tunnel chiffré permet d'encapsuler et d'acheminer d'une façon sécurisée et discrète des protocoles applicatifs non sécurisée tels que FTP, POP3, IMAP, RLOGIN, etc).

III.FONCTIONNEMENT DÉTAILLÉ :

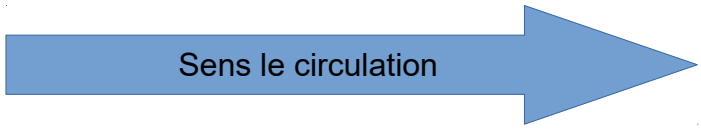
III.1.AVANT-PROPOS

Nous avons vu qu'il existe plusieurs types de tunnels. Si leur principe général reste toujours le même (encapsulation des paquets), leur fonctionnement détaillé varie notablement en fonction de ces types. Il n'est donc pas question, dans le cadre de cet ouvrage, de les décrire tous. Nous allons nous contenter d'étudier les deux types les plus utilisés : les tunnels IP (niveau routage) et les tunnels SSH (niveau application).

III.2.RAPPEL : COUCHES DE PROTOCOLES TCP/IP:

Un paquet IP circulant sur un réseau peut être représenté de la manière suivante :

Payload	Couches de protocole de niveau "application"	Couche de protocole de niveau "transport"	Couche de protocole de niveau "réseau"	Couche de protocole de niveau "liaison"
Données utiles du message (UDP) ou d'un fragment du message (TCP) échangé entre expéditeur et destinataire	Correspondent au niveaux 5,6 et 7 de l'OSI Données d'identification Et d'authentification	correspond au niveaux 4 de l'OSI ou au protocole TCP Numéros de ports des processus logiciels communicants	Correspond au niveaux 3 de l'OSI ou au protocole IP Données de routage. Adresses logiques de l'expéditeur et du destinataire	Correspond au niveaux 2 de l'OSI Données de liaison point à point. Adresses MAC de l'expéditeur et du destinataire



III.2.1.IMPLANTATION PHYSIQUE D'UN TUNNEL :

Les techniques de "tunnellisation" ("tunnelling" en anglais) consistent à encapsuler, à l'entrée d'un tunnel, un paquet dans un nouveau paquet muni des "couches de tunnelisation" nécessaires. A la sortie du tunnel, le paquet d'origine sera "décapsulé" puis émis vers son destinataire final.

Pour réaliser ces traitements, il est évidemment nécessaire de configurer les deux nœuds marquant les extrémités du tunnel et d'y implanter les logiciels qui les supportent. Pour un sens de circulation donné ceux-ci sont du type Client-Serveur (le client encapsule, le serveur décapsule et émet vers le destinataire final).

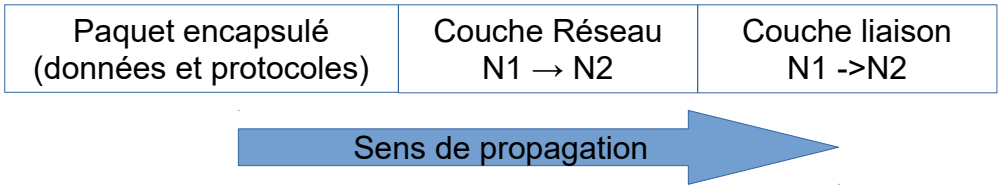
REMARQUE : un tunnel étant en général bidirectionnel, chacun des nœuds extrémités sera muni d'un client et d'un serveur.

III.3.CAS D'UN TUNNEL IP:

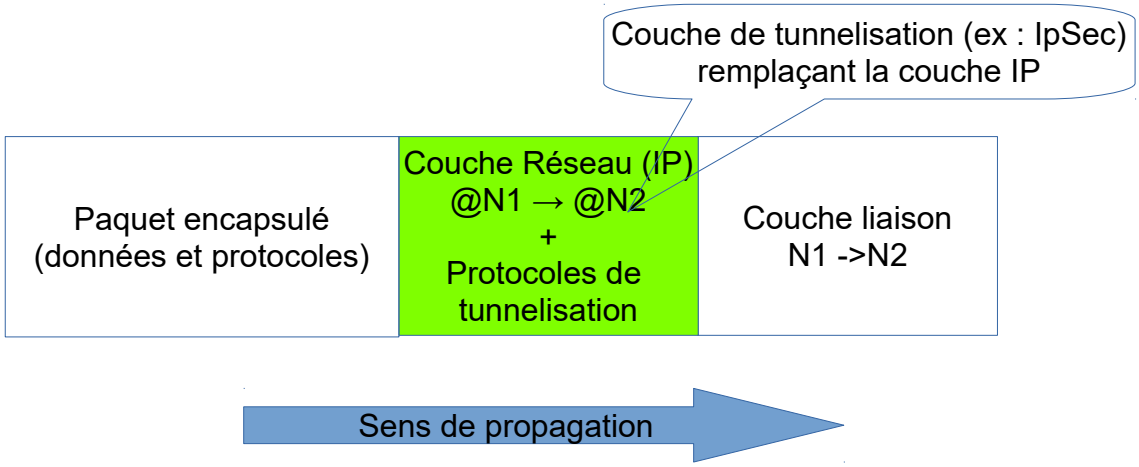
III.3.1.COUCHES DE PROTOCOLE CONCERNÉES:

Un tunnel IP masque pour les observateurs extérieurs les données de liaison et de routage des paquets pendant leur trajet entre les deux extrémités N1 et N2 du tunnel. Ceci implique que ces données soient encapsulées dans le nouveau paquet.

Ce nouveau paquet ne sera utilisé que pendant le trajet de N1 à N2 : de ce fait, il suffit que ce paquet "englobant" soit muni des couches liaison et réseau adéquates (source = @N1, destination = @N2). Dans un premier temps, on peut imaginé un paquet englobant constitué comme suit :



En fait, il faut ajouter également des informations pour signaler à la sortie du tunnel que le paquet en encapsule un autre et qu'il faut donc le traiter en conséquence (décapsulation, décryptage éventuel, émission du paquet décapsulé vers sa destination). La couche réseau du paquet englobant sera donc modifiée pour inclure ces informations:



III.3.2.IMPLANTATION D'UN TUNNEL IP :

III.3.2.1.PRINCIPES :

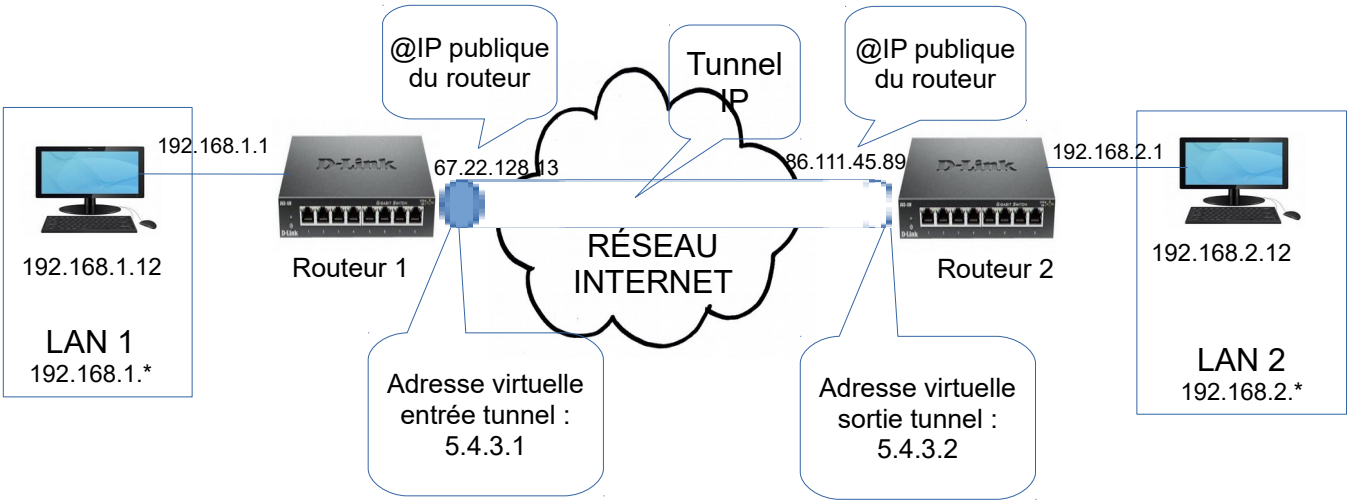
Pour créer un tunnel IP bidirectionnel, Il faut implanter dans les deux nœuds extrémités de ce tunnel :

- Un logiciel CLIENT permettant d'expédier des paquets dans le tunnel après les avoir encapsulés et éventuellement cryptés ;
- Un logiciel SERVEUR capable de décapsuler ces paquets, de les décrypter si besoin, et de les réexpédier vers leur destinataire final.

REMARQUE : *Ces logiciels agissant au niveau de la couche de routage, ils peuvent être implantés dans les ROUTEURS WEB des LAN. Dans ce cas, tous les postes du LAN peuvent utiliser le tunnel sans qu'il soit utile de modifier de leurs logiciels internes.*

D'autre part, à l'entrée et la sortie du tunnel (ou source et destination) doivent être associés des interfaces réseau virtuels, munis d'adresses IP spécifiques :

EXEMPLE :



III.3.2.2.IMPLANTATION SOUS LINUX (ubuntu) :

Dans un systèmes LINUX, une batterie de commandes en ligne permet d'installer des logiciel tunnels "libres" comme IPIP (IP on IP) ou GRE (Generic Routing Encapsulation). Les exemples suivants se basent sur l'architecture représentée par le schéma. Les paramètres d'installation sont écrits en rouge :

INSTALLATION EN LIGNES DE COMMANDES DU TUNNEL "IPIP1" ENTRE LE ROUTEUR 1 ET LE ROUTEUR 2 :

Sur le routeur 1, l'installation en ligne de commande correspond à :

```
ip tunnel add ipip1 mode ipip remote 86.111.45.89 local 67.22.128.13 ttl 255
ip link set ipip1 up
ip addr add 5.4.3.1/24 dev ipip1
```

Sur le routeur 2, l'installation en ligne de commande correspond à :

```
ip tunnel add ipip1 mode  ipip remote 67.22.128.13 local 86.111.45.89 ttl 255
ip link set ipip1 up
ip addr add 5.4.3.2/24 dev ipip1
```

INSTALLATION DANS LE FICHIER /ETC/NETWORK/INTERFACES DU TUNNEL "GRE1" AVEC OPTION MULTICAST:

Routeur 1:

```
auto gre1
iface gre1 inet static address 5.4.3.1
 netmask 255.255.255.0
 up ifconfig gre1 multicast
 pre-up iptunnel add gre1 mode gre remote 86.111.45.89 local 67.22.128.13  ttl 255
 pointopoint 5.4.3.2
 post-down iptunnel del gre1     #supprime le tunnel après son passage à l'état "down"
```

Routeur 2:

```
auto gre1
iface gre1 inet static address 5.4.3.2
 netmask 255.255.255.0
 up ifconfig gre1 multicast
 pre-up iptunnel add gre1 mode gre remote 67.22.128.13 local 86.111.45.89  ttl 255
 pointopoint 5.4.3.1
 post-down iptunnel del gre1   #supprime le tunnel après son passage à l'état "down"
```

NOTA :

- Cette installation par le fichier "interfaces" a l'avantage de "survivre" à un redémarrage du système, contrairement à l'installation en lignes de commande ;

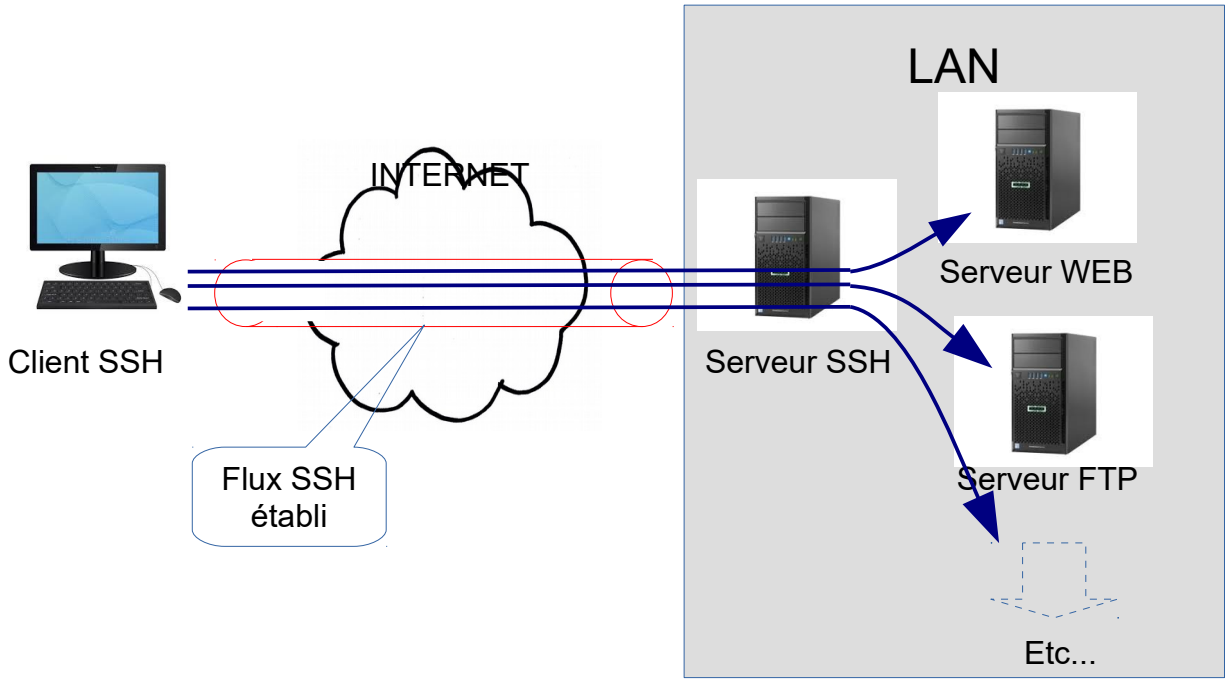
- Il faut redémarrer le service réseau des machines pour que la configuration soit prise en compte.

III.4.CAS D'UN TUNNEL SSH :

III.4.1.PRINCIPES :

Le protocole SSH permet la redirection du trafic utilisant un port TCP donné d'une machine vers un autre port (Port Forwarding) en utilisant un "flux SSH" établi: cette possibilité permet de créer des TUNNELS SSH : un port local du CLIENT est redirigé vers le port distant du serveur que l'on veut utiliser.

Le flux SSH établi entre client et serveur SSH permet d'encapsuler et d'acheminer d'une façon sécurisée et discrète des protocoles applicatifs non sécurisée tels que (FTP, POP3, IMAP, RLOGIN, etc) : A l'entrée du tunnel, les paquets sont encapsulés dans des paquets SSH. A la sortie, ces paquets sont décapsulés et expédiés vers le serveur destinataire. Les réponses du serveur subissent (en sens inverse) un traitement identique.



III.4.2.CRÉATION D'UN TUNNEL SSH :

III.4.2.1.EN LIGNE DE COMMANDES (WINDOW et LINUX) :

Les systèmes d'exploitation LINUX modernes intègrent en général un client SSH dans le configuration de base. En revanche, les serveurs SSH doivent faire l'objet d'installations supplémentaires.

Sous Debian et ubuntu, le logiciel installé est basé sur le produit libre openSSL. La commandes d'installation est :

sudo apt-get install openssh-server (installation du paquet openssh-server)

La version 10 de windows permet d'activer un client ou serveur openSSH sur la machine en utilisant le menu "applications/fonctionnalités supplémentaires".

La création du tunnel proprement dit est réalisée par une commande ssh -L dont la syntaxe générale est :

#ssh -L <PortLocal>:<HôteLocal>:<PortDistant> <NomUtilisateur>@<HôteDistant>

oû :

- <PortLocal> : Numero du port local sur lequel sera redirigé le port d'écoute du serveur. ;
- <HôteLocal > : nom (ou @ip) de la machine locale ;
- <PortDistant> : Numero du port d'écoute du serveur que l'on veut atteindre (pas celui serveur SSH, mais bien celui du serveur que l'on veut atteindre par le tunnel);
- <NomUtilisateur> : nom d'utilisateur du serveur SSH ;
- <HôteDistant> : nom (ou @ip) de la machine distante .

<PortLocal>:<HôteLocal>:<PortDistant> réalise la redirection du port local choisi vers le port du serveur que l'on veut utiliser.

EXEMPLE :
#ssh -L 5400:localhost:80 arthur@192.168.1.12

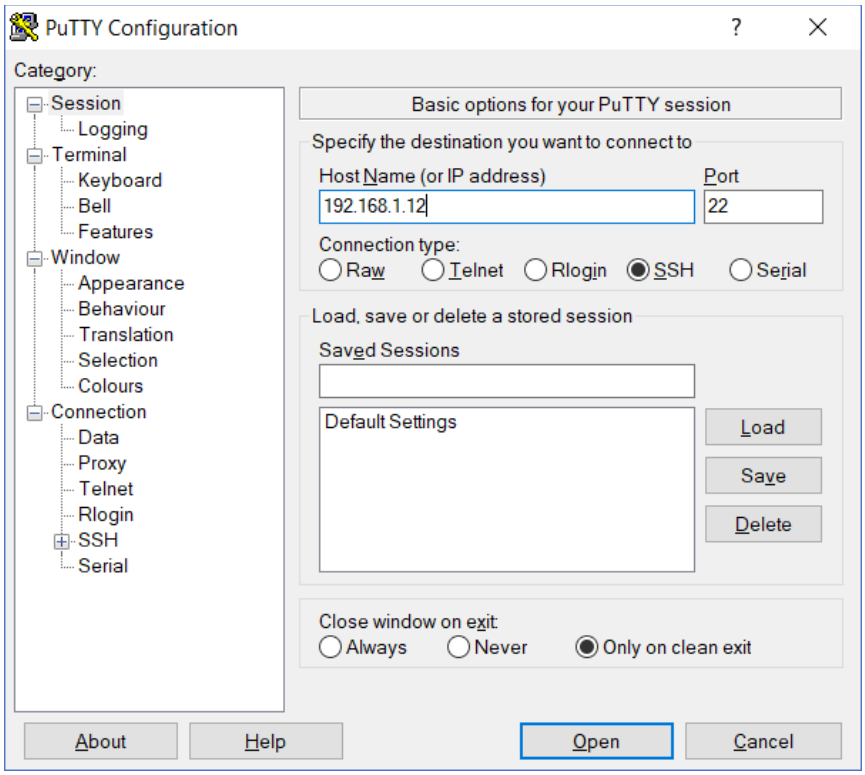
réalise l'ouverture d'un flux ssh entre la machine locale et la machine 192.168.1.12 avec pour nom d'utilisateur "arthur" et redirige le port local 5400 vers le port d'écoute 80 de la machine distante (port du serveur http local). De ce fait, la saisie dans la barre d'adresse d'un navigateur local de l'url 127.9.0.1:5400 aboutit au serveur http de la machine distante.

REMARQUE : Le tunnel SSH ne peut être utilisé que pour communiquer avec le serveur de la machine distante dont le port est spécifié dans la commande. Pour communiquer avec un autre serveur, il faudra définir un autre tunnel.

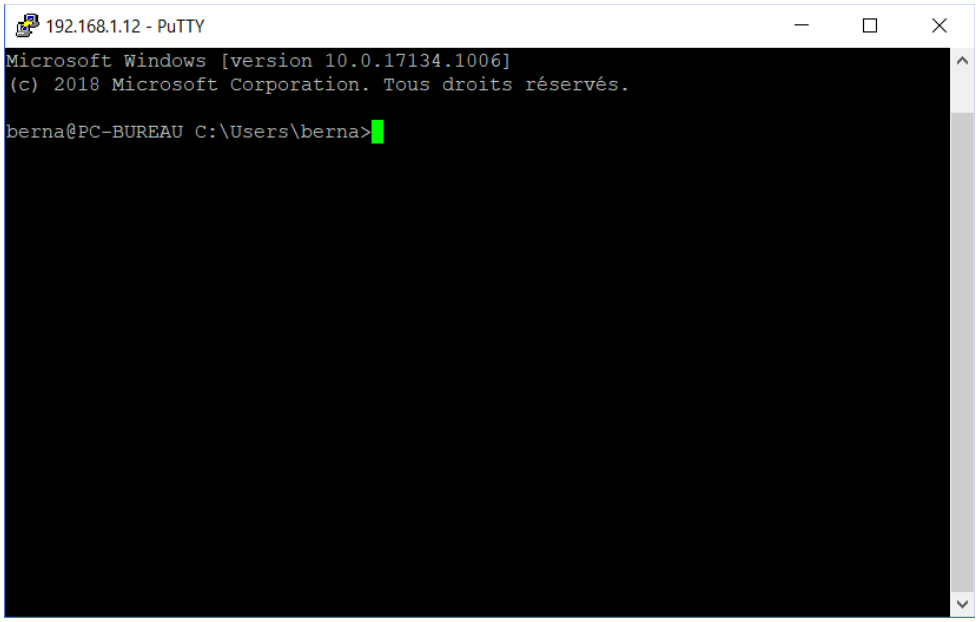
III.4.2.2.UTILISATION DE PUTTY (WINDOWS) :

Le logiciel PUTTY permet de créer des tunnels SSH par des formulaires "graphiques" :

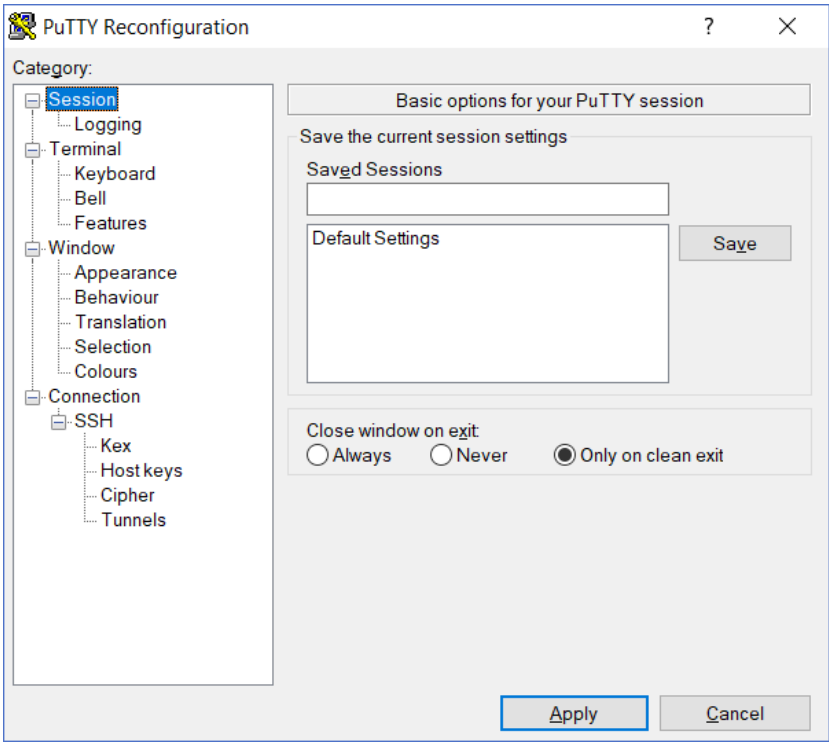
Ouvrir d'abord un tunnel SSH :



Valider par "open" ; un terminal système déporté s'affiche :



Cliquer alors en haut à gauche. Le menu de configuration de putty s'affiche de nouveau :



Sélectionner SSH puis Tunnels dans la partie gauche du menu, puis saisir les paramètres du tunnel et appliquer :