Formation à l'utilisation et l'exploitation des routeurs Netline

Sommaire

I. Introduction aux réseaux locaux et distants

I.1. Rappels préliminaires sur le modèle OSI

Le modèle OSI décompose les communications numériques en différents niveaux aux rôles distincts. Nous ne verrons ici que les 3 premiers niveaux de ce modèles, puisqu'il s'agit de rappel et que seuls ceux-ci sont nécessaire s à la compréhension du fonctionnement d'un routeur.

Le premier niveau concerne la couche physique, c'est à dire le moyen physique utilisé pour transmettre les informations. Il peut définir l'utilisation de signaux électriques, d'ondes ou de signaux lumineux. Les normes utilisées à ce niveaux définissent donc, par exemple le 0 et le 1 comme des niveaux électriques de plus ou moins quelques volts.

La couche physique définit le fonctionnement de l'émetteur et du récepteur du signal utilisé lors de la communication.

Le second niveau est appelé couche liaison, son rôle est de permettre une communication entre deux ou plusieurs équipements reliés sur un même fil. On parle très souvent de connexion point à point. Ce niveau définit la forme des échanges entre deux équipements d'une façon logique : il permet entre autre d'identifier un équipement sur le fil grâce à son adresse. Cette adresse est une adresse de niveau 2, ce peut être par exemple une adresse MAC (réseaux Ethernet). La couche liaison permet aussi la sécurisation des transmissions de données entre deux équipements en calculant pour chaque paquets émis un checksum ou CRC qui garantira que les données reçues n'ont pas subit de modification durant leur transport. Enfin, le niveau liaison prévoit des mécanismes de numérotation des paquets permettant de reconstituer les messages transmis dans leur ordre de départ et la détection de perte de paquets.

La couche liaison adresse, sécurise et ordonne les communications entre deux équipements reliés par un même câble.

Le troisième niveau est appelé couche réseau, son rôle est de s'affranchir de la limite de transmission "point à point" en utilisant des machine tierces pour joindre une destination extérieure au réseau local. On parle de routage de paquets, on appelle les équipements capables de cela des routeurs. Le principe d'aiguillage repose sur l'utilisation d'un couple adresse / masque pour chaque équipement comprenant à la fois un identifiant de réseau et un identifiant d'équipement dans le réseau. Un routeur est un équipement capable de trouver une route pour joindre un réseau distant. Le niveau trois assure que les données sont juste, connaît l'ordre des trames et leur perte.

La couche réseau apporte la capacité de joindre un réseau distant et de s'affranchir de l'aspect point à point de la couche liaison.

I.2. Les réseaux locaux

Né du besoin, au sein des groupes de travail, d'échanger de plus en plus d'informations numériques, il s'agit de réseaux reliant au sein d'un même espace des équipements informatiques. Le principal exemple étant un réseaux de micro-ordinateurs au sein d'une même agence, d'un même lieu. Ces équipements sont reliés au travers d'une couche physique généralement de type Ethernet à un débit de 10 ou 100 Mbits/s. La distance séparant deux entités dans ce réseau est limitée à quelques centaines de mètres.

Les équipements peuvent être connectés au travers d'un bus, mais cette technologie tend à disparaître. Ils sont donc généralement branchés à des concentrateurs dont le rôle est de propager les paquets émis par les uns vers tous les autres.

Les réseau locaux sont des solutions à haut débit n'entraînant généralement pas de coût d'utilisation, si bien qu'ils sont constamment connectés.

Les protocoles de communication peuvent être assez divers, et même spécifiques au dessus de la couche liaison. Actuellement TCP/IP est le plus répandu,mais certains comme IPX/SPX peuvent toujours exister. Au sein d'un réseau local, les équipements peuvent échanger des paquets dit de "broadcast". Ceux-ci sont émis par un équipement et traité par tous les autres. Il est donc possible dans un réseau local de découvrir les équipements présents et il n'est donc pas forcement nécessaire de réaliser un plan d'adressage des équipements. Toutefois, l'utilisation généralisée de TCP/IP nous contraint maintenant à cette tâche.

Les réseaux locaux offrent un très haut débit sur une faible distance avec un coût d'installation et d'exploitation très faible.

I.3. Les réseaux distants

Les réseaux dits distants, aussi appelés WAN pour Wide Area Network, sont des réseaux qui généralement permettent l'interconnexion de réseaux locaux (appelés LAN pour Local Area Network). Ces réseaux couvrent de très longues distances en utilisant des technologies similaires à celle utilisées par la téléphonie. Les moyens de communication sont divers : RNIS, Satellites, ATM, liaisons louées... Ces réseaux ont un coût de mise en place et d'exploitation bien plus important que les réseaux locaux.

Selon le moyen choisi, il peut engendré une facturation de son utilisation au mois ou à la minute, voire selon la quantité d'information ayant transité. Dans beaucoup de cas, l'utilisation de ces réseaux passe d'abord par une connexion au réseau de sorte à ne pas être facturé alors que celui-ci n'est pas utilisé.

Dans de nombreux cas l'accès à ces réseaux est un service offert par un opérateur spécialisé. Cette opérateur aura construit son réseau WAN mêlant des technologies RNIS/ATM/téléphonies/... et couvrant un territoire donné.

Les réseaux distants offrent un débits généralement faible sur de très longues distances avec un coût d'installation et d'exploitation élevé.

I.3.1. Le réseau RNIS

RNIS ou Réseau Numérique à Intégration de Service est un support permettant de disposer des services tels que la voix, la visiophonie et la transmission de données. Pour ce faire, RNIS utilise deux type de canaux de communication : le premier sert à la transmission des données précédemment citées il est appelé canal B. Le second est utilisé pour la signalisation, il est appelé canal D.

Le canal D sert principalement à l'établissement des communications, dans certain cas, il permettra l'envoie de données pour des services spécialisés. Ce canal doit être connecté en permanence pour la réception des appels. Le canal D est à très faible débit car peu d'informations y circulent. Ce canal peut être utilisé pour de nombreux services annexes aux communications. Il est possible de connecter plusieurs équipements sur un même accès RNIS de base appelé bus S0. Ces équipements vont alors se partager l'utilisation de ce canal de signalisation.

La canal B est utilisé pour la communication, il supportera les données générées par la voix, l'image ou tout équipement informatique connecté. Il permet un débit de base de 64Kbits/s. Toutefois, pour des applications plus gourmandes, il est possible de grouper plusieurs canaux B de sorte à avoir à disposition une bande passante multiple de 64Kbits. Une communication utilise un canal B dans son ensemble, il ne peut donc être partagé. Par contre, dans le cas d'un bus S0, il est possible que plusieurs équipements se partagent, au fur et à mesure des communications, les canaux B présents. Ainsi, lorsque que sont mis à disposition deux canaux dans un accès de base, il est possible d'avoir beaucoup d'éléments en attente d'appel, par contre simultanément ne seront possibles que deux communications.

La facturation des appels se fait en fonction du temps d'occupation des canaux B. Chaque canal B donne lieu à sa propre facturation. Le canal D peut être utilisé pour des transferts de données sur des services comme X25. La facturation se fait alors en fonction de la quantité d'information transmise.

Les liaisons RNIS offrent une liaison numérique d'un débit conséquent avec une facturation à l'appel. Elles permettent de faire transiter à la fois des informations numériques et vocales vers n'importe quel usager.

I.3.2. Les lignes spécialisées

Il s'agit de liaisons séries dont l'utilisation est louée à un opérateur téléphonique. Ces liaisons sont permanentes et relient point à point deux sites. La bande passante va de quelques Kbits/s à quelques Mbits/s.

La Tarification des lignes louées est forfaitaire, des frais sont perçus à l'installation en plus d'une redevance mensuelle dépendant du débit et de la distance.

Les lignes spécialisées offrent une liaison numérique au débit paramétrable dont la facturation est forfaitaire. Ces communications ont un seul et unique destinataire.

II. Rappels sur les protocoles de communication

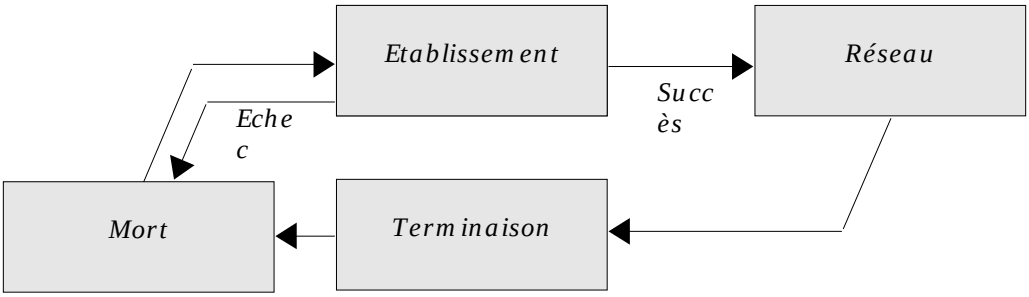
II.1. Le protocole PPP

PPP signifie Protocole Point à Point ; Il s'agit d'un protocole de communication dédié à l'échange de données entre deux équipements distants reliés par une liaison point à point. Cette dernière peut être la liaison RNIS ou la ligne louée dont nous avons précédemment parlé. PPP est, dans le modèle OSI, un protocole de niveau liaison, il ne prend donc pas en charge de routage, mais permet de faire transiter des données de niveau 3. Il sera donc adapté au transport de protocoles tels que IP.

PPP assure la configuration de la liaison, l'établissement d'adresses réseau, l'authentification, la compression... La liste de ses services n'est pas exhaustive : PPP est un protocole ouvert et peut donc être étendu.

Les services de PPP sont des sous-protocoles s'occupant d'une tâche bien précise lors de la communication ou de son établissement, nous les verrons par la suite.

Lors de l'établissement d'une communication, les deux entités réseaux échangent des trames PPP pour la négociation des paramètres à utiliser lors de la transmission à venir. La communication est alors dans un état appelé Etat d'établissement ; dès lors que les deux extrémités se seront accordées, cet état deviendra Etat Réseau et la transmission débutera. En fin de communication de nouveaux messages de fermeture de connexion seront échangés, l'état sera alors Etat de terminaison enfin la ligne sera raccrochée et l'état Etat Mort.



PPP est un protocole permettant le transport de données comme le protocole IP sur une liaison point à point comme celle établies par un modem, un accès RNIS ou une liaison louée. PPP s'occupe de l'authentification lors de la connexion, la compression de données et autres.

II.1.1. Le protocole LCP

LCP est un protocole transportant les paquets utilisés pour l'établissement, le maintien et la terminaison de connexion PPP. LCP permet la négociation des options utilisées lors de la communication, les deux entités doivent se mettre d'accord sur celles-ci, si tel est le cas, la communication aboutit, sinon elle se termine. Lorsqu'une communication aboutit, d'autres paquets LCP seront échangés pour sa surveillance.

LCP, comme la plupart des protocoles étendant PPP, fonctionne sur le principe de la négociation d'options interactive : une extrémité propose une liste d'options qu'elle souhaiterai que l'autre utilise. Si celle-ci rejette une proposition, l'extrémité source fera une nouvelle proposition. Cet échange se poursuit jusqu'à ce qu'un accord soit trouvé. Chaque entité négocie les options pour son sens récepteur ; les options peuvent donc être différentes en émission et en transmission.

Une option contient généralement un paramètre, l'entité recevant la demande d'option peut soit la rejeter, soit l'accepter telle quelle, soit l'accepter sous réserve de modifier le paramètre, alors elle propose sa valeur.

Les options qui peuvent être ainsi négociées sont par exemple, la taille maximum de trame reçue (MRU), le protocole à utiliser pour l'authentification,la compression des champs de contrôle et d'adresse...

LCP est le principal sous-protocole de PPP permettant de négocier les options de la liaison PPP. Les options sont négociées de façon interactive entre les extrémités. Elles peuvent être différentes en réception et en émission.

II.1.2. L'authentification

L'authentification permet de limiter l'accès à un équipement à un groupe restreint d'équipements distants identifiés par un login et un mot de passe. Plusieurs protocoles existent, PAP et CHAP.

PAP, pour Password Authentication Protocol, est le plus simple des protocoles, l'utilisateur transmet en une trame le nom de l'utilisateur (login) et son mot de passe. Il est accepté ou refusé. Les données sont transmises en clair sur la liaison, ce peut être dans certains cas un trou de sécurité.

CHAP, pour Challenge Handshake Authentication Protocol, est un protocole plus sécurisé : l'authentificateur soumet au client un nom et une chaîne aléatoire qu'il doit modifier suivant un algorithme précis à l'aide d'une clef secrète. Le client retourne son nom et la chaîne ainsi modifiée. L'authentificateur vérifie le résultat avec sa propre clef secrète et autorise ou non la communication. Les informations permettant l'authentification ne sont donc transmises sur la liaison que cryptées. Cette méthode est donc plus sure, mais elle nécessite un traitement un peu plus long.

L'authentification est un moyen de limiter la communication à certains utilisateurs, les procédés fonctionnent sur l'échange d'une clef associée à un nom. Cet échange peut se faire en clair par l'utilisation de PAP ou de façon cryptée par l'utilisation de CHAP.

II.1.3. Le protocole IPCP

IPCP est utilisé pour négocier les options relatives au transport d'IP. Il est ainsi possible de configurer la compression de données IP par exemple. Mais il est surtout possible de configurer la liaison IP et alors de recevoir une adresse IP dans le réseau distant que l'on cherche à joindre. Les paramètres de DNS peuvent dans le même temps être transmis. Cette possibilité est intéressante car elle permet d'utiliser les adresses IP de façon dynamique, allouée à chaque connexion et non de façon statique, allouée alors à chaque utilisateur.

Une fois les options IPCP établies, il est possible de transmettre sur la liaison PPP des trames IP.

II.2. Le protocole Frame Relay

Frame relay est une évolution simplificatrice de la commutation par paquets X.25, il est initialement conçu comme un protocole du réseau RNIS. Il utilise des principes identiques pour le routage sans assurer toutefois l'intégrité des données et le contrôle de flux. Les paquets sont transportés dans les trames noeud après noeud : seule la couche liaison est utilisée. Frame Relay est donc un protocole de même niveau que PPP, par contre, il permet de multiplexer statiquement des communications sur une même liaison physique. Une telle communication s'établit sur un circuit virtuel (ou CV) qui peut être permanent ou établi à la demande. Le CV est unidirectionnel.

Frame Relay est parfaitement adapté à l'interconnexion de réseaux locaux pour des protocoles divers et plus particulièrement pour X25 de façon transparente.

Frame relay permet un débit de 64Kbits/s à 34Mbits/s avec de très faibles temps de réponse ; il est particulièrement adapté aux forts trafics aléatoires : la facturation sur Frame-Relay se fait à la quantité de données transmises plutôt que fonction du débit alloué.

L'accès à Frame Relay est fourni par un opérateur qui met à disposition son réseau composé de noeuds reliés les uns aux autres par un réseau maillé de connexions à haut débit. L'usager se connecte au noeud, ses trames sont insérées dans le réseau haut débit et aiguillées selon des tables de commutation. L'adressage des équipements se fait grâce à des DLCI (identifiant de connexion).

Frame relay permet une attribution de bande passante aux utilisateurs et la garantie d'un débit moyen.

Contrairement à PPP, Frame Relay permet d'établir plusieurs communications (CV) sur un seul canal physique. La tarification se fait à la quantité et non au temps.

II.3. La pile TCP/IP

IP (Internet Protocol) est, à l'origine, un développement du DoD mis dans le domaine public et implémenté sur les système UNIX. Il connaîtra un rapide développement dans le mode des réseaux locaux.IP est une couche de niveau réseau sur laquelle reposent la plupart des applications réseaux actuelles. Elle permet une communication inter-réseaux sans avoir à se soucier du (ou des) support(s) physique(s) utilisés pour joindre la destination. IP est une interface commune à la plupart des systèmes communicants, c'est le coeur du réseau Internet.

II.3.1. L'adressage IP

IP permet de joindre n'importe quelle destination, située n'importe où dans le monde dès lors qu'il existe une route possible ; peu importe les canaux physiques utilisés. Les deux points de la communication sont identifiés par des adresses répondant à une codification particulière : l'adresse comprend une partie réseau et une partie équipement. Un équipement fait donc parti d'un réseau dans lequel il possède une adresse. Les adresses sont codées sur 4 entiers de 8bits que l'on sépare généralement par un point.

Prenons par exemple l'adresse 192.168.0.15 elle comprend un numéro de réseau : 192.168.0.xx et un numéro de machine 15. Nous parlons de l'équipement 15 dans le réseau 192.168.0.

Il existe plusieurs classes de réseaux répondant à des besoins différents en terme de taille. Les réseaux de classe A n'utilisent qu'un seul octet pour spécifier le réseau et doivent être compris entre 1 et 126 le premier bit doit être zéro. Il est à noté qu'on l'on n'utilisera pas les adresses composées uniquement de 0 ou de 1. Leur nombre est faible mais en conséquence, le nombre d'équipements en leur sein est très important puisque codifié sur les 3 octets restants :

Par exemple l'adresse 10.0.0.2 signifie que l'on parle de l'équipement 0.0.2 dans le réseau 10

La classe B utilise 2 octets pour le réseau et 2 octets pour le numéro d'équipement, il est nécessaire de faire débuter le numéro de réseau par les bit "10" alors les plages possibles sont 129.0.xx.xx à 191.255.xx.xx.

La classe C utilise elle 3 octets pour le réseau et 1 pour l'équipement, il est nécessaire de faire débuter le numéro de réseau par les bits "110" alors les plages sont 192.0.0.xx à 223.255.255.xx.

D'autres classes permettent la diffusion de groupe et l'expérimentation, nous ne les détailleront pas.

Certaines adresses sont particulières car dites non routables il s'agit d'adresses privées qui ne peuvent exister sur l'Internet. Elle sont donc choisies de façon privilégiées lors de la mise en place de réseaux privés. L'utilisation d'autres adresses, hors mis dans certains cas précis, passe par une demande d'attribution de plage d'adresse à un opérateur. Ces adresses sont 10.xx.xx.xx pour les réseaux de classe A et 192.168.00.xx à 192.168.254.xx pour les réseaux de classe C.

Il est possible de scinder un réseau en plusieurs sous-réseaux. Avec une adresse de classe C, par exemple, on a un réseau avec 254 équipements possibles ou 4 sous-réseaux avec chacun 62 équipements possibles. On utilise pour cela un indicateur toujours présent avec l'adresse IP : le masque réseau.

Il s'agit d'une valeur sur 4 octets (comme l'adresse) où chaque bit à 1 indique que le bit correspondant dans l'adresse fait parti du numéro de réseau. Un bit à 0 indique un bit d'équipement. Bien sur, un masque ne doit pas être composé de 1 et de 0 intermittents, le réseaux est composé des bits de gauche, la séparation réseau/équipement doit être franche :

masque 228.0.0.0
masque (binaire) 11100100.00000000.00000000.00000000
N'EST PAS UN MASQUE VALIDE

masque 255.192.0.0
masque (binaire) 11111111.11000000.00000000.00000000
EST UN MASQUE VALIDE

Appliqué à l'exemple précédemment évoqué, notre réseau 192.168.0.0 peut être découpé en 4 sous réseaux avec le masque suivant :

masque 255.255.255.192
masque (binaire) 11111111.11111111.11111111.11000000

Les équipements dans ces réseaux auront pour adresses : 192.168.0.1 à 192.168.0.63
192.168.0.65 à 192.168.0.126
192.168.0.129 à 192.168.0.190
192.168.0.193 à 192.168.0.254

On notera que les adresses extrêmes d'un réseaux ne sont pas utilisées. Lorsque tous les bits de la partie équipement sont à 1, il s'agit d'une adresse de diffusion : la trame sera destiné à l'ensemble des équipements du réseau. On évitera d'utiliser les adresses avec tous les bits à 0.

Les classes standard correspondent à des masques eux aussi standard :

La classe A a pour masque 255.0.0.0

La classe B a pour masque 255.255.0.0

La classe C a pour masque 255.255.255.0

L'adresse IP est un élément important qui permet de d'organiser les équipements en réseaux. Différentes classes existes permettant d'adapter la taille des réseaux aux besoins réels. Dans un soucis d'organisation et d'optimisation, il est possible de re-décomposer un réseaux en sous-réseaux grâce au masque de réseau.

II.3.2.Résolution d'adresse

Le procédé de résolution d'adresse permet de faire le lien entre une adresse de niveau 3 (celui d'IP) et une adresse de niveau inférieur. Lorsqu'un message part d'un équipement A (192.168.0.1) à destination de l'équipement B (192.168.0.2) ayant pour masque réseau 255.255.255.0, l'équipement A, va tout d'abord vérifier que B est bien dans le même réseau que lui. Si tel est le cas, il sait qu'il peut joindre B de façon directe car B est sur le réseau local. Hors, le réseau local repose sur une couche de niveau 2, il est donc nécessaire d'indiquer à cette couche une adresse de B compatible.

Ce procédé est appelé résolution d'adresse et suit un protocole appelé ARP pour Address Resolution Protocol. Le principe de fonctionnement est simple : il suffit de demander par une trame de diffusion qui est le propriétaire de l'adresse IP que nous cherchons à joindre (192.168.0.2). Ce dernier répondra et transmettra ainsi son adresse de niveau 2. Il est à noté que ces correspondances entres adresses de niveau 3 et adresses de niveau 2 sont conservé par les piles de protocoles dans des caches. De cette façon, le réseau n'est pas sans cesse encombré par ces interrogations.

Notons que dans le cas où la destination n'est pas dans le réseau local, il n'est pas envisageable de demander une adresse de niveau 2 dans tous les réseaux interconnectés, celle-ci n'aurait d'ailleurs pas de sens du fait que les couches 2 ne sont pas forcement identiques. La trame sera alors transmise à la passerelle, équipement du réseau local capable de joindre des réseaux distants. Nous reviendrons là-dessus par la suite.

Le protocole ARP permet de connaître la correspondance entre les adresses de niveau 2 et celle de niveau 3, son principe repose sur l'interrogation de l'ensemble des équipements au travers d'une trame de diffusion.

II.3.3. Le protocole ICMP

ICMP transporte des messages de diagnostics indiquant les erreurs réseaux. Il permet l'interrogation d'information réseau comme la date du jour et les masques des adresses IP. L'utilitaire ping produit les messages ICMP : il envoie un message de demande d'écho devant provoquer une réponse de d'écho. Cette outil est primordial pour le tests du bon fonctionnement de l'équipement par l'utilisateur. Les messages peuvent aussi être envoyé par les équipements indiquant par exemple qu'un service demandé n'est pas disponible ou qu'un équipement demandé n'est pas joignable.

ICMP est utilisé pour le diagnostic, il sert a l'utilisateur pour verifier le bon fonctionnement d'un équipement et à la pile IP pour transmettre des messages d'erreur.

II.3.4. Les protocoles TCP et UDP

TCP et UDP sont des protocoles de niveau 4, ils reposent sur IP. C'est sur ces protocoles que reposent les applications, TCP et UDP fournissent un adressage par port en plus de l'adresse IP. Ainsi, le couple adresse IP + port TCP ou UDP indique directement une application particulière sur un équipement précis. Il est alors possible d'adresser plusieurs applications sur un même équipement.

TCP fournit un service orienté connexion aux applications souhaitant envoyer à un destinataire un flot de données de façon fiable. TCP inclus une numérotation des trames et prévoit des fenêtres pour les acquittements et détection de pertes.

UDP fournit un service sans connexion, les trames reçues ne sont pas acquittées, la détection de perte de trame ne se fera que par l'absence de réponse aux requêtes.

TCP et UPD interfacent les applications avec le niveau IP en ajoutant la notion de port à l'adressage. TCP offre un service orienté connexion contrairement à UDP.

II.3.5. Le service d'annuaire

Retenir des couples d'adresses IP et de ports n'étant pas vraiment simple, un service d'annuaire appelé DNS a donc été mis en place, il fait correspondre une adresse IP avec un nom plus simple. L'adresse 192.168.0.1 peu correspondre a équipement1.monréseau.fr par exemple. Nous noterons que le nom est composé de plusieurs sous parties séparées par des points. Ceci est due à une organisation hiérarchique des serveurs d'annuaires. Ainsi, lorsque la correspondance sera recherchée, on pourra interroger le serveur recensant les .fr, lui même saura qui est le serveur pour monréseau, ce dernier étant à même de répondre à la question : Quelle est l'adresse IP de l'équipement équipement1?

Dans le même esprit de simplification, les ports TCP et UDP ont été pour certains nommés de sorte à ce qu'ils soient simple à retenir : le port 80 est appelé par exemple http. Ainsi, les certains ports ont pris le nom des services auxquels ils sont associés.

Ainsi, l'adresse http://équipement1.monréseau.fr équivaut dans notre exemple à 192.168.0.1:80.

Les serveur DNS, sont capable de s'interroger les uns les autres et intègrent généralement une mémoire cache pour éviter les ré-interrogation intempestives.

Le service DNS simplifie la mémorisation des adresses en leur associant un nom. Celui-ci est organisé de façon hiérarchique pour simplifier les recherches au sein des bases.

II.3.6. Le service telnet

Le service telnet repose sur la pile TCP/IP pour proposer une émulation de terminal sur un équipement distant. Ce service est primordial pour administrer une machine au travers d'un réseau. Telnet est un protocole non sécurisé transférant tous les caractères tapés en clair.

II.3.7. Le service ftp

FTP pour File Tranfer Protocol est un service permettant le transfert de fichiers au travers d'un réseau. Il permet de transmettre des fichiers caractères ou binaires. FTP comprend un certain nombre de commandes permettant d'envoyer et recevoir les fichiers, de naviguer dans les arborescences et créer des répertoires. Un grand nombre d'outils graphiques simplifient l'utilisation de FTP.

II.3.8. La translation IP

Il existe dans le monde plus de systèmes informatiques connectés à Internet que d'adresses IP disponibles, par conséquent, il faut trouver une solution, IPV6 est en cours de développement. Toutefois, sur l'ensemble des ordinateurs connectés, peu d'entre eux ont réellement besoin de pouvoir être joint directement : peu ont donc besoin d'une adresse propre. La translation IP permet de partager une seule adresse pour un groupe d'équipements ou plus simplement pour un réseau privé. Tous les postes accédant à l'extérieur vont alors utiliser l'adresse unique fournie à la connexion par le fournisseur d'accès Internet.

L'appareil gérant la translation utilise cette adresse publique, il va modifier les trames émises de sorte à en devenir la source, ainsi, une réponse pourra lui être retournée. Il modifiera aussi le port utilisé de sorte à ce que grâce à l'adresse du site distant et ce numéro, il puisse retrouver sur le réseau local le bon équipement destinataire.

III. Le routage des paquets au travers des réseaux

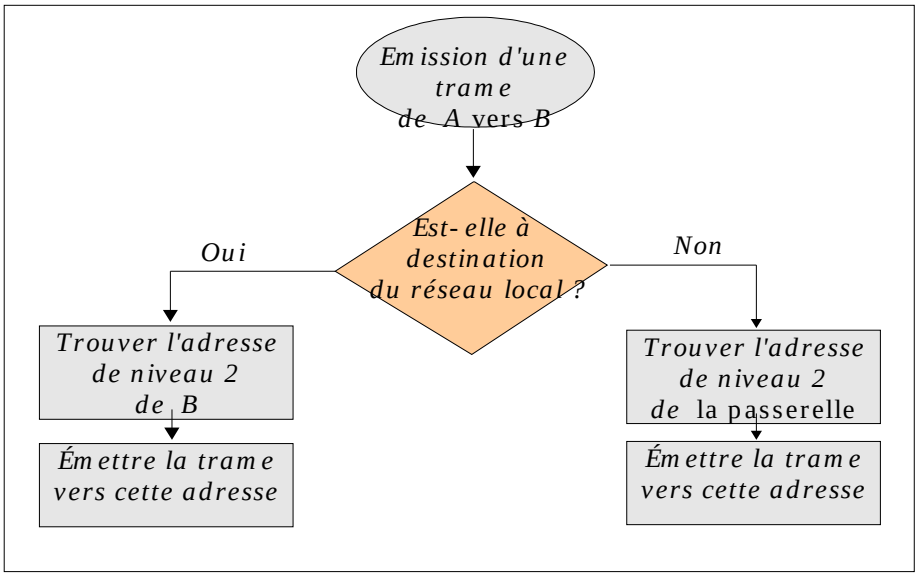
Comme nous l'avons vu précédemment, le routage des informations se fait au niveau de la couche réseau permettant ainsi l'élimination des problèmes liés au support : les couches de niveau inférieur. Le routage est la technique qui permet d'aiguiller au sein du réseau une trame , de sa source à sa destination. On ne parle de routage que lorsqu'un paquet doit être transmis hors du réseau local de la source. Le routage est à la charge d'équipements spécifiques que sont les routeurs. Un routeur est un équipement autonome, mais ce peut être aussi un ordinateur. Pour des raisons évidentes de sécurité, stabilité et maintenance, l'utilisation d'un appareil spécialisé est préférable.

Un routeur est un équipement reliant deux réseaux ou plus, ceux-ci sont connectés au travers de liaisons de type Ethernet ou télécoms. Un routeur est un élément actif du réseau qui ressemble, bien que dissimulé au coeur d'un boîtier à un système informatique, il nécessite par conséquent une configuration. Bien qu'associé à une tâche d'apparence simple, un routeur est un équipement aux fonctionnalités multiples. Nous verrons par la suite la configuration des équipements Netline, mais avant tout, nous allons revenir sur le fonctionnement même d'un routeur.

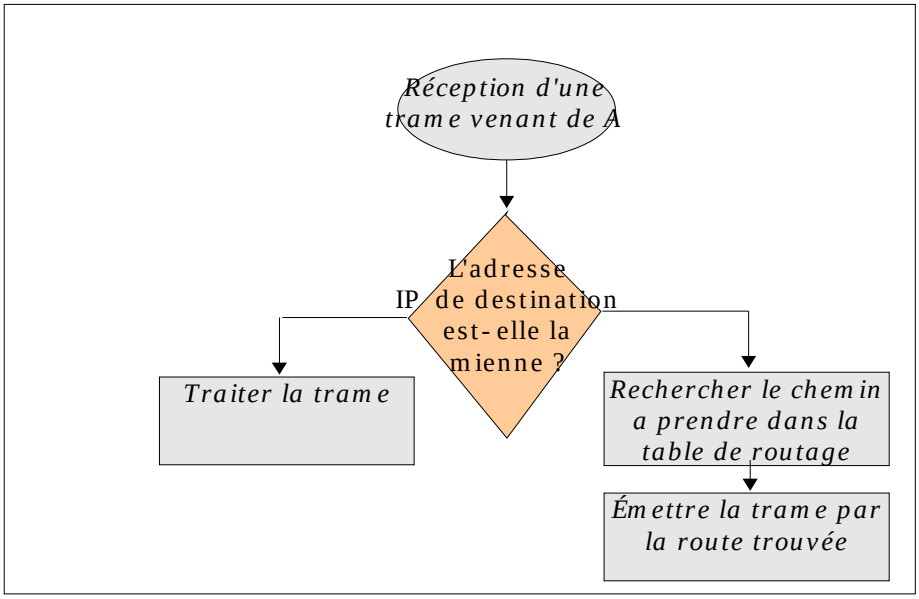
III.1. Principes de base et routage statique

Nous avons évoqué précédemment le cheminement d'une trame à destination du réseau local, revenons là dessus et voyons le cas d'une trame à destination d'un réseau distant : imaginons la machine A (192.168.0.1) souhaitant joindre un équipement B (192.168.1.1). Lorsque A souhaite contacter B, la pile IP découvre que la destination est hors du réseau local, alors, elle va émettre la trame à destination de l'équipement routeur du réseau. L'adresse IP de cet équipement lui étant connu dans sa configuration. Seulement il n'est pas possible de modifier l'adresse IP au sein de la trame, sans quoi l'information de destination serait perdue, par conséquent, l'adresse IP sera toujours celle de B, par contre l'adresse de niveau 2 qui sera insérée sera celle de l'équipement routeur. Ce dernier, recevant une trame ne lui étant pas destiné comme indiqué par le niveau 3, l'aiguillera vers le lien propice à joindre B. Pour trouver la route adéquat, le routeur va se référer à une table de routage. Il s'agit d'un tableau donnant pour chaque réseau joignable le périphérique vers lequel émettre la trame.

Les algorithmes ci-dessous résume cela :



Principe d'émission



Principe de réception d'une trame IP dans un routeur

Pour émettre une trame à destination d'un réseau distant, celle-ci est transmise à un équipement : le routeur, celui-ci recherchera dans sa table de routage vers quelle interface ré-émettre la trame pour qu'elle arrive à sa destination.

Maintenant que le principe de base du routage est compris, il nous faut examiner les tables de routage et voir de quelle façon celles-ci sont renseignées. Le procédé le plus simple est l'utilisation de routes statiques.

Une route statique est une route définie de façon durable et immuable par l'administrateur du système. Elle indique quel chemin utiliser pour joindre une destination donnée. Il est possible d'indiquer une route par défaut, que toutes les trames dont la destination n'est pas explicitement indiquée suivront.

Voyons un exemple de table de routage sur un système LINUX:

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
192.168.63.0	*	255.255.255.0	U	0	0	0	eth0
192.168.64.0	*	255.255.255.0	U	0	0	0	eth1
193.56.124.0	192.168.64.254	255.255.255.0	UG	0	0	0	eth1
default	192.168.63.254	0.0.0.0	UG	0	0	0	eth0

Tout d'abord, nous trouvons les routes des réseaux locaux sur lesquels nous sommes connectés, une carte réseau est branchée au réseau 192.168.63.xx et une autre est branchée sur 192.168.64.xx. Les deux cartes réseaux sont identifiées par les noms d'interface eth0 et eth1. Ces réseaux ne nécessitent pas de passer par un routeur : le champ Gateway n'est pas renseigné.

Viennent ensuite les routes vers les réseaux distants : pour joindre le réseau 193.56.124.xx la table indique que la trame doit être envoyée à la passerelle (routeur) d'adresse 192.168.64.254, et qu'il faut emettre la trame sur la carte eth1.

La dernière ligne est une route par défaut : toutes les trames dont on a pas encore identifié la destination comme connue seront transmises au routeur 192.168.63.254 sur l'interface adapté : eth0.

La table de routage peut être remplie de façon statique, comme il en est le cas dans l'exemple, cette opération est réalisé par l'administrateur à l'aide d'une commande route.

Ajouter une route à destination de 193.56.125.xx se fera de la façon suivante sur un routeur Netline :

route **add**193.56.125.0 192.168.64.254

Sont précisés l'opération a effectuer dans la table, ici un ajout (**add**) , l'adresse du réseau a joindre et l'adresse du routeur à utiliser, celle-ci étant l'adresse de l'interface à utiliser sur le routeur.

D'autres paramètres peuvent être ajoutés : un masque différent de celui par défaut :

route add 193.56.125.0 192.168.64.254 **masque 255.255.255.128**

Il est aussi possible de préciser un métrique. Il s'agit de donner un coût à cette route, exprimé en nombre de saut. S'il y a plusieurs chemin possible, celui de plus faible métrique sera adopté.

route add 193.56.125.0 192.168.64.254 **metrique 2**

Une route sera supprimée de la table de la même façon à l'aide de la commande del :

route **del** 193.56.125.0 192.168.64.254

Les routes statiques sont ajoutées et supprimées par l'administrateur à l'aide de la commande 'route'. Ces routes sont définies de façon durables. Elles indiquent quelle passerelle joindre pour atteindre une destination donnée. La passerelle peut être le routeur lui même alors identifié par son adresse interne d'interface.

III.2. Routage à la demande

Certaines routes ne peuvent être maintenues actives, c'est la cas de celle empruntant des lignes RNIS par exemple : la facturation à la minute d'utilisation rendrait leur coûts trop élevé. Par conséquent, certains routages ne se font qu'à la demande, c'est à dire, uniquement lorsqu'il y a nécessité réelle d'envoyer des données. Un programme dédié établira la communication pour que la donnée puisse être envoyée et raccrochera la ligne après. Généralement, le raccrochage s'effectue à l'expiration d'une temporisation durant laquelle aucune autre données n'a utilisé la ligne. Cette temporisation permet de minimiser les surcoûts liés à l'établissement sans pour autant consommé trop de temps de communication inutile. Toutefois, le temps choisit pour cette temporisation n'est qu'empirique et il est difficile de trouver une valeur idéale valable pour tous les réseaux.

Examinons une table de routage de N2211 pour voir en quoi le routage à la demande influe :

Table de routages

	Destination	Masque	Passerelle	Flags	CnxRef	Utilisat.	Intf
(1)	localhost	0xFFFFFFFF	localhost	U H L	0	0	lo0
(2)	autocom	0xFFFFFFFF	autocom	U H ldb	0	0	vfr0
(3)	127.0.0.2	0xFFFFFFFF	127.0.0.2	U H L	0	0	nul0
(4)	0.0.0.3	0xFFFFFFFF	192.168.3.1	U H ldb	0	0	sl0
(5)	0.0.0.4	0xFFFFFFFF	n2211b_3.2	U H ldb	0	0	cB1
(6)	0.0.0.5	0xFFFFFFFF	n2211b_3.2	U H ldb	0	0	cB2
(7)	192.168.3.0	0xFFFFFF00	autocom	UG l	0	0	vfr0
(8)	20.0.0.0	0xFFFFFF00	n2211b_3.2	U L	0	0	il0
(9)	193.56.124.0	0xFFFFFF00	autocom	UG b	0	0	vfr0
(10)	192.168.62.0	0xFFFFFF00	192.168.62.3	U L	0	0	il1

Nous voyons différents périphériques comme précédemment : nul0 identifie un périphérique utilisé pour la destruction de trames, lo0 le loopback, sl0 la liaison série, cB1 et cB2 les canaux RNIS, il0 et il1 les interfaces Ethernet. Il reste une interface supplémentaire : vfr0, il s'agit du système de routage à la demande.

Regardons les routes existantes et principalement celles nécessitant une connexion : les canaux RNIS. Nous voyons que pour joindre 193.56.124.xx nous devons passer par la passerelle autocom (ligne 9). Il en est de même pour les routes utilisant la liaison série (ligne 7).

Lorsque la communication sera établie par autocom la table deviendra par exemple :

(9)	193.56.124.0	0xFFFFFF00	autocom	UG b	0	0	cB1

Nous constatons que la passerelle reste autocom, par contre, l'interface de sortie devient cB1, la canal physique ouvert par autocom.

La configuration de l'appel à la demande se fait au travers d'un fichier de configuration que nous étudierons par la suite.

Le routage à la demande permet d'ouvrir une connexion que lorsqu'il est nécessaire de transmettre des données, un périphérique appelé autocom se charge d'établir et couper les communications, il met à jour lui même la table de routage.

III.3. Routage dynamique par RIP

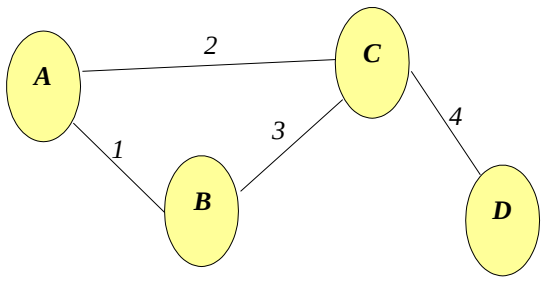
RIP pour Routing Information Protocol est un des protocoles les plus simple, il fonctionne sur le principe de routage à vecteur distance. La distance entre deux points est exprimé en nombre de sauts, c'est à dire en nombre de passerelles à traverser pour atteindre la destination.

Contrairement au routage statique, RIP à son propre protocole de communication inter-routeur, dès lors, les équipements sont capable d'échanger leur table de routage. Au démarrage, le routeur utilisant RIP ne connaît que lui-même, il découvrira par la suite ses voisins. Ceux-ci vont lui transmettre leur propre table de routage, le routeur incrémentera le coût de chaque destination et ajoutera ces informations à sa table. Si le routeur possède déjà une route existante de coût moindre il ignorera la nouvelle route qui lui a été transmis de sorte que seul le plus court chemin soit mémorisé. Le coût maximum d'une destination est de 16, à partir de cette valeur, elle sera considérée comme inaccessible.

Les tables de routage sont diffusées périodiquement (toutes les 30sec) ce qui est gourmand en bande passante. Si aucune information n'est reçue d'un lien au bout de 180 secondes, on considère que le lien est tombé, alors la table de routage est nettoyée pour supprimer les routes utilisant ce lien. Il en va de même lorsqu'une route n'est plus rafraîchie : elle sera supprimé au bout de 180 secondes. Ces différents principes impliquent un temps de convergence élevé durant lequel des trames peuvent être perdues.

Sur les lignes nécessitant une connexion, la mise à jour est spécifique de sorte à éviter les connexions intempestives qui se réveilleraient onéreuses.

Examinons le remplissage des tables de routage du réseau de routeur suivant :



A, B, C, D identifient des réseaux

1,2,3,4 identifient des liens

Etat des tables de routage au démarrage des équipements :

De A vers	Lien	Coût
A	Local	0

De B vers	Lien	Coût
B	Local	0

De C vers	Lien	Coût
C	Local	0

De D vers	Lien	Coût
D	Local	0

Après 30 secondes, chaque équipement reçoit le table de son ou ses voisins:

De A vers	Lien	Coût
A	Local	0
B	1	1
C	2	1

De B vers	Lien	Coût
B	Local	0
A	1	1
C	3	1

De C vers	Lien	Coût
C	Local	0
A	2	1
B	3	1
D	4	1

De D vers	Lien	Coût
D	Local	0
C	4	1

Après 30 nouvelles secondes, ces nouvelles tables sont à nouveau échangées :

- On notera que par exemple A reçoit de B une route vers C avec un coût total qui sera alors de 2. A possède déjà une route vers C avec un coût égal à 1, cette route de B ne sera pas conservée.
- B apprend lors de cette échange qu'il peut joindre D en passant par C avec un coût total de 2.

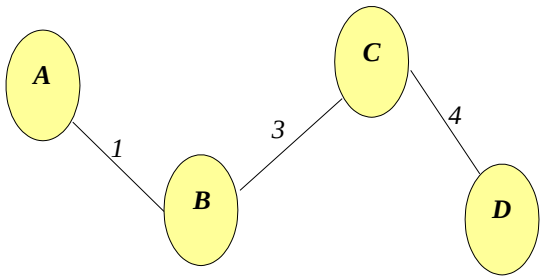
De A vers	Lien	Coût
A	Local	0
B	1	1
C	2	1
D	2	2

De B vers	Lien	Coût
B	Local	0
A	1	1
C	3	1
D	3	2

De C vers	Lien	Coût
C	Local	0
A	2	1
B	3	1
D	4	1

De D vers	Lien	Coût
D	Local	0
C	4	1
A	4	2
B	4	2

Imaginons que le lien 2 tombe :



les tables deviennent les suivantes :

De A vers	Lien	Coût
A	Local	0

De B vers	Lien	Coût
B	Local	0

De C vers	Lien	Coût
C	Local	0

De D vers	Lien	Coût
D	Local	0

De A vers	Lien	Coût
B	1	1

De B vers	Lien	Coût
A	1	1
C	3	1
D	3	2

De C vers	Lien	Coût
B	3	1
D	4	1

De D vers	Lien	Coût
C	4	1
A	4	2
B	4	2

Lors de l'échange des tables nous avons donc une remise à jour des chemins :

De A vers	Lien	Coût
A	Local	0
B	1	1
C	1	2
D	1	3

De B vers	Lien	Coût
B	Local	0
A	1	1
C	3	1
D	3	2

De C vers	Lien	Coût
C	Local	0
A	3	2
B	3	1
D	4	1

De D vers	Lien	Coût
D	Local	0
C	4	1
A	4	2
B	4	2

RIP a permis de trouver de nouveaux chemins pour remplacer le lien perdu : les routes qui précédemment avaient été ignorées car trop coûteuses sont maintenant utilisées.

RIP est une méthode de routage permettant la découverte des réseaux distants et des routes a suivre pour les joindre. Le trames suivent la route ayant le moins de saut. Les routes sont dynamiquement remises à jour de façon périodique prenant donc en compte l'état des lignes.

III.4. Routage dynamique par OSPF

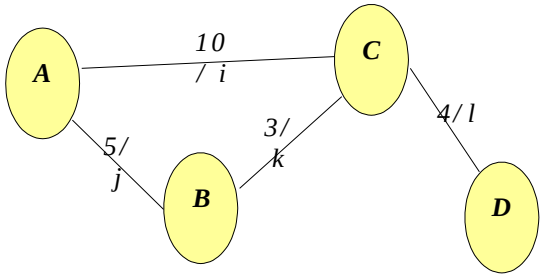
OSPF pour Open Shorted Path First est un protocole de routage dynamique tout comme RIP. Il est, par contre un protocole à état des liaisons et autorise le routage multivoie. Contrairement à RIP, seules les informations ayant changées sont diffusées sur le réseau, dès lors la bande passante est préservée, les mises à jour ont lieu dès qu'une modification est détectée et l'ensemble du réseau est réactualisé toutes les 30 minutes.

OSPF fonctionne non pas sur le nombre de sauts mais sur la bande passante offerte par un lien, la route choisie sera celle dont la bande passante est la meilleure, prenant en compte la globalité du chemin à parcourir. Pour cela, un routeur OSPF à en mémoire toutes les informations de tous les liens du réseau. Contrairement à RIP, OSPF ne supprime pas une information dès lors qu'il a une solution meilleure en mémoire, de cette façon, si la meilleur solution venait à disparaître, OSPF pourrait calculer une nouvelle route de plus faible coût. En effet, OSPF calcule sa route sur l'ensemble des données du réseau, la méthode utilisée garantie l'emploie de la route de plus faible coût et élimine les risques de boucles.

OSPF permet un découpage du réseau en zones (les area) de sorte à limiter la taille des réseaux et les ressources nécessaires aux stockage et calculs.

OSPF est un protocole de routage plus évolué que RIP, basant ses calculs sur la bande passante disponible. Il permet une mise à jour plus rapide des tables de routage, le support de plus important réseaux et garantie le choix de la route de plus faible coût sans création de boucles.

Voyons par l'exemple la calcul de routes sur l'exemple suivant :



A,B,C,D sont des réseaux

5,3,4,10 sont les coût des liens, plus le coût est faible plus la bande passante est grande.

i,j,k,l sont les noms donnés aux liens.

Après échange des informations initiales entre les routeurs, nous avons les tables suivantes :

De	Vers	Coût
A	A	0
A	B	5
A	C	10
B	C	3
C	D	4

De	Vers	Coût
B	B	0
A	B	5
A	C	10
B	C	3
C	D	4

De	Vers	Coût
C	C	0
A	B	5
A	C	10
B	C	3
C	D	4

De	Vers	Coût
D	D	0
A	B	5
A	C	10
B	C	3
C	D	4

A partir de ces données, chacun peut construire sa table de routage des plus courts chemins :

De A vers	Lien	Coût
A	Local	0
B	j	5
C	j	8
D	j	12

De B vers	Lien	Coût
B	Local	0
A	j	5
C	k	3
D	k	7

De C vers	Lien	Coût
C	Local	0
A	k	8
B	k	3
D	l	4

De D vers	Lien	Coût
D	Local	0
A	l	12
B	l	7
C	l	4

Imaginons que le lien j tombe, l'ensemble des routeurs du réseau sera alors prévenu de cette modification et relancera le calcul de ses routes :

De A vers	Lien	Coût
A	Local	0
B	i	13
C	i	10
D	i	14

De B vers	Lien	Coût
B	Local	0
A	k	13
C	k	3
D	k	7

De C vers	Lien	Coût
C	Local	0
A	i	10
B	k	3
D	l	4

De D vers	Lien	Coût
D	Local	0
A	l	14
B	l	7
C	l	4

Chaque routeur retient toutes la topologie du réseau et à partir d'une simple information concernant une modification dans celle-ci il sera capable de recalculer de nouvelles routes.

III.5. Création de réseaux privés virtuels

Les réseaux privés virtuels (VPN) ont deux vocations : la première est de faire transiter des données entre deux réseaux privé de façon transparente, en utilisant un comme support un réseau publique. La seconde est de sécuriser les échanges de données entre deux réseaux privés par des méthodes de cryptage et d'authentification. Le cryptage consiste à rendre les données échangées incompréhensibles par un tiers. L'authentificateur garantie, l'intégrité et la provenance des données reçues.

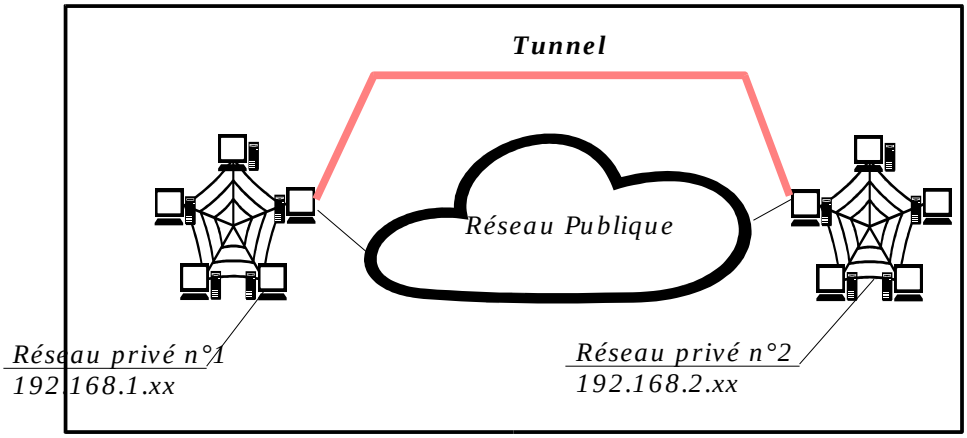
Dans de nombreuses circonstances, la nécessité de cryptage et d'authentification n'est pas évidente, dans ce cas, le VPN peut utiliser des technologies appelées tunnels.

III.5.1. Utilisation de tunnel pour la réalisation de VPN

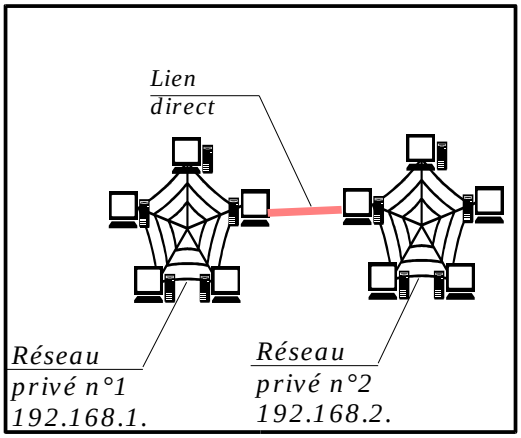
Le principe du tunnel est assez simple : toute trame à destination d'un réseau distant déterminé est mis dans une nouvelle trame à destination du point d'entrée de ce réseau. Arrivée, elle sera extraite de sa trame support et émise telle-qu'elle sur le réseau. Ainsi, les deux réseaux, bien que distant seront virtuellement directement connectés. Ceci répond à deux besoins :

- Router des trames entre deux réseaux privés d'adresses non routables au travers d'un réseau publique.
- S'affranchir des contraintes liées à la translation d'adresse entre deux réseaux identifiés.

Un tunnel peut être représenté de la façon suivante, de façon physique et de façon logique :



Représentation physique d'un tunnel



Représentation logique d'un tunnel

Il existe plusieurs protocoles pour réaliser un tunnel : GRE, IPIP, PPTP, PPPOE, L2TP. Les différences entre ces protocoles vient principalement du type de trame qui peut être transporté dans le tunnel ou du support utilisé pour le tunnel. GRE est le protocole le plus souvent utilisé.

Un tunnel permet de créer un lien direct mais virtuel entre deux réseaux privés en utilisant comme support un réseau publique. Le tunnel permet de s'affranchir des contraintes liées à la translation IP.

III.5.2. Utilisation d'IpSec pour la réalisation de VPN

IpSec fonctionne de façon similaire au tunnels que nous venons de voir.Les données ne sont plus simplement encapsulées dans une trame, elles sont aussi cryptées et authentifiées. Dans le cas d'un tunnel IPSEC, on parle de connexion entre les deux réseaux privés, car il y a avant tout échange une discussion entre les deux passerelles réalisant l'encapsulation IpSec. En effet, celles-ci doivent s'échanger des clefs qui seront employées pour le cryptage et l'authentification, elles doivent aussi s'accorder sur les méthodes à utiliser pour ces opérations. Il y a donc deux phases dans la communication IpSec : la négociation et la communication elle-même.

La négociation est elle-même cryptée, c'est d'ailleurs la partie la plus sécurisée de la communication car transite alors les informations nécessaires au chiffrement/déchiffrement du flux qui sera transporté par le tunnel. Il existe différentes méthodes pour le chiffrage de cet échange, mais l'idée repose sur l'utilisation de clefs publiques et de clefs privés. La liaison sécurisé, il sera possible de faire transité dessus des informations capitales comme les clef utilisées pour le chiffrement du flux du tunnel où des méthodes plus légères sont utilisées. Les méthodes utilisées pour la négociation repose sur les principes de clés pré-partagées et de groupe de Diffe-Hillman 768 à 2048 bits.

Pour plus de sécurité, les communications sont re-négociées à période régulière, de sorte à renouveler les clefs utilisées. La durée de vie d'une communication est paramétrable.

La négociation terminée, la communication entre les réseaux privés est établie, l'échange peut commencer, les trames sont donc cryptées/authentifiées. L'authentification garantie l'origine de la trame et son intégrité, en effet, il est impossible à un tiers modifiant une trame de reconstituer l'authentification. Dans ce cas, toute intervention par un tiers sera identifiable. Le cryptage, garantit le secret des données transportées, toutefois, les puissances processeurs disponibles ne permettent pas une sécurisation fiable de chaque donnée prise individuellement. Par conséquent, c'est la sécurisation utilisée liée à la quantité importante de données transmises qui assurera le secret. Les méthodes de cryptage couramment utilisées sont DES, 3DES, CAST128, BlowFish, RC5, IDEA. Le méthode d'authentification sont MD5, SHA1, SHA2-256.

L'utilisation de cryptage demande la réalisation de très nombreux calculs, ceux-ci peuvent faire chuter les performance du tunnel. Il faut donc bien veiller à avoir une adéquation entre le débit souhaité, le niveau de sécurisation choisi, et le matériel utilisé.

IpSec apporte aux tunnels la notion de sécurité et d'authentification. IpSec demande une phase de négociation des paramètres qui seront utilisé pour la communication après quoi les données peuvent être échangées. IpSec est un protocole gourmand en calcul.

III.6. Qualité de service

Lorsque l'on route des informations d'un réseau local vers un réseau distant, on passe généralement d'un support à haut débit vers un support à bien plus faible capacités si bien que la bande passantes utilisée pour communiquer avec l'extérieur est rapidement toute utilisée. Nous rencontrerons généralement ce problème lors de partage d'accès à Internet entre plusieurs ordinateurs : il suffit que l'un d'eux effectue un téléchargement pour que les autres souffrent d'un manque évident de bande passante.

La notion de qualité de service répond à ce problème par deux moyens distincts :

- Il pourrait être envisagé que la bande passante soit partagée entre les différents utilisateurs, auquel cas le téléchargement d'un des utilisateurs serait restreint à une certaine utilisation de débit. Cette solution permet aux autres utilisateur l'accès à un minimum de bande passante, leur garantissant un confort minimum. Bien entendu, ces limitations sont convenues de façon réfléchie, il faut utiliser au mieux la bande passante et donc ne pas créer de restrictions inutiles.
- Certains trafic sont particuliers, on les caractérise généralement d'interactif car ils sont générés par des échanges entre un utilisateur et un système informatique. Il s'agit principalement de services tels que telnet. Ils demandent la transmission de faibles flux de données séparés par de longue périodes vide. Tous ralentissements sur ces flux est directement ressenti par l'utilisateur qui subit un temps de latence désagréable. Pour éviter cela, nous pouvons envisager de rendre ce trafic prioritaire sur tous les autres. Il est a noté que cette priorité l'affectera que très faiblement les autres trafics du fait du peu de bande passante utilisé par les services interactifs.

Nous distinguons donc deux classe de trafics : ceux qui sont interactifs, que nous appelons par convention EF et les autres appelé AF. L'information classant les trafics est contenu dans les trames, car indiqué par le service ayant créer le flux.

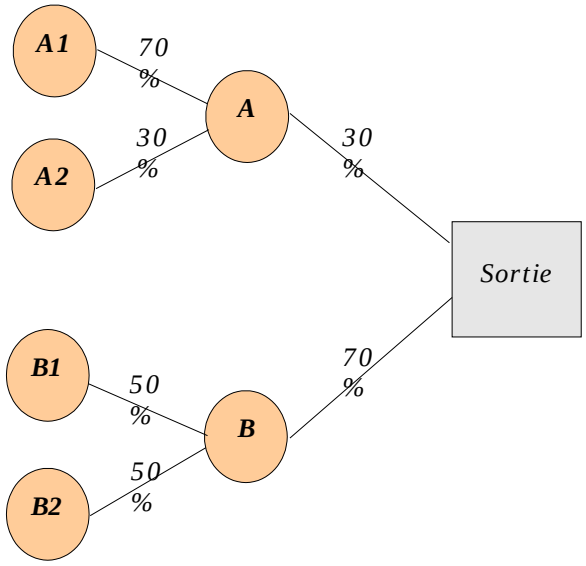
A partir de là, il existe de nombreuses méthodes pour classer et ordonnancer les flux AF, nous ne les détaillerons pas toutes ici. Par contre, il est a noter que le paramétrage spécifique et réfléchi d'un équipement est la clef d'une bonne qualité de service.

La Qualité de service, permet le partage organisé d'un accès à bas débit. Il permet de garantir aux utilisateur une bande passante minimum et ainsi offrir une utilisabilité correcte des services réseaux.

III.4.1. Qos-CBQ

CBQ pour Class Based Queuing est une méthode actuellement disponible sur les routeurs Netline. Son principe repose sur la définition de classes organisées hiérarchiquement. Un classe est définit sur un critère de trie des trames entrantes, ce peut être fonction des adresses IP, des services utilisés... Ensuite, est accordé à chaque classe une partie de la bande passante disponible. Le système veillera à ce que les trafics issus de ces classes respectent leur attribution de bande passante.

Dans le schéma ci-dessous, est représenté une hiérarchie de classe et leur critères :



Dans le schémas ci-contre, nous avons distingué deux classes principales A et B, pouvant correspondre, à une classification par adresse IP. Celles-ci séparant les serveurs (A) des utilisateurs (B). 70% de la bande passante est alors allouée aux utilisateurs, contre 30% pour les serveurs.

Des sous classes ont été définit la dessus, Au niveau de la classe A, une sous classification, par exemple par type de service a été faite accordant 70% des 30% de bande réservée à http (A1) et 30% à l'échange de mails (A2).

Au sein des utilisateurs, des sous-classe accorde par exemple a chacun d'eux une bande passante égale à 50% des 70% qui leur est tous réservé.

Au final, les utilisateurs auront, au minimum 35% de la bande passante totale chacun, le service http 21% et le service mail 9%.

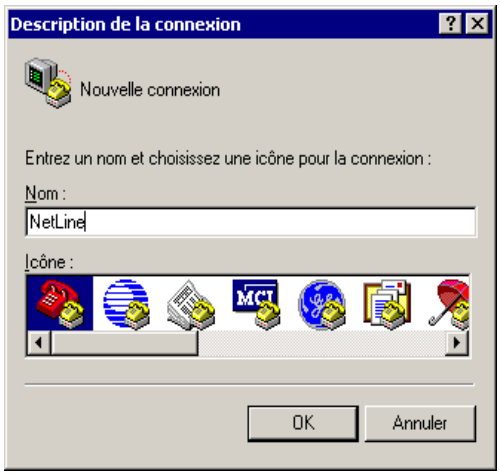
IV. Installation d'un routeur Netline

IV.1. Prise en main de l'équipement

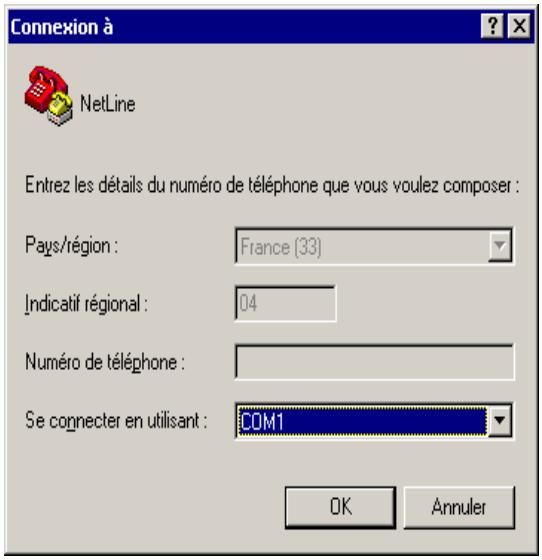
IV.1.1. Configuration de la console

L'équipement N2211 nécessite un configuration préalable avant de pouvoir être utilisé sur un réseau Ethernet. En effet, il est nécessaire de lui attribuer ses paramètres réseau. Le port console situé en façade de l'équipement permet de raccorder le routeur à un ordinateur. Un câble est fournit avec le matériel dans cet optique. Ce câble va vous permettre d'accéder au pupitre de l'équipement au travers d'un émulateur de terminal, outil disponible sous tous les système d'exploitation. Dans un environnement Windows, recherchez le programme HyperTerminal, nous vous conseillons toutefois de rechercher d'autres outils plus performants comme TeraTerm par exemple.

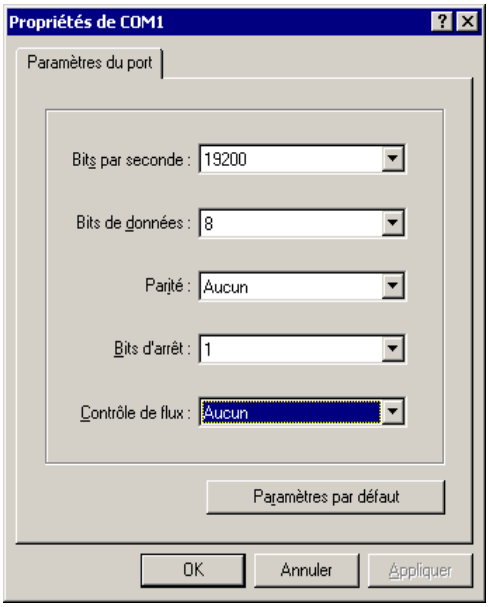
Le câble connecté sur un port série identifié de votre station, lancez HyperTerminal. Donnez un nom à votre connexion (par exemple Netline) et validez.



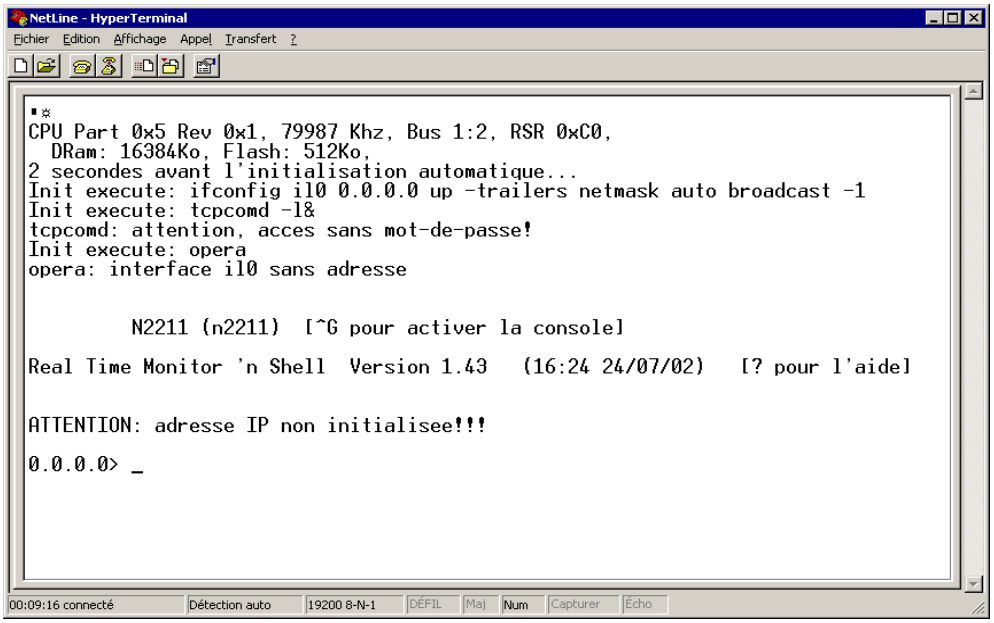
Sélectionnez sur l'écran suivant le port série sur lequel est connecté le routeur :



Configurez alors les paramètres du port série : vitesse 19200, 8 Bits de données, pas de parité, 1 bit d'arret et aucun contrôle de flux.



Mettez enfin l'équipement Netline sous tension pour voir apparaître les messages de démarrage. Notez que si l'équipement était déjà branché, cela ne pose aucun problème, mais vous ne verrez rien à l'écran. Dans tous les cas, enfoncez simultanément les touches [Ctrl] et G du clavier pour ouvrir la console.



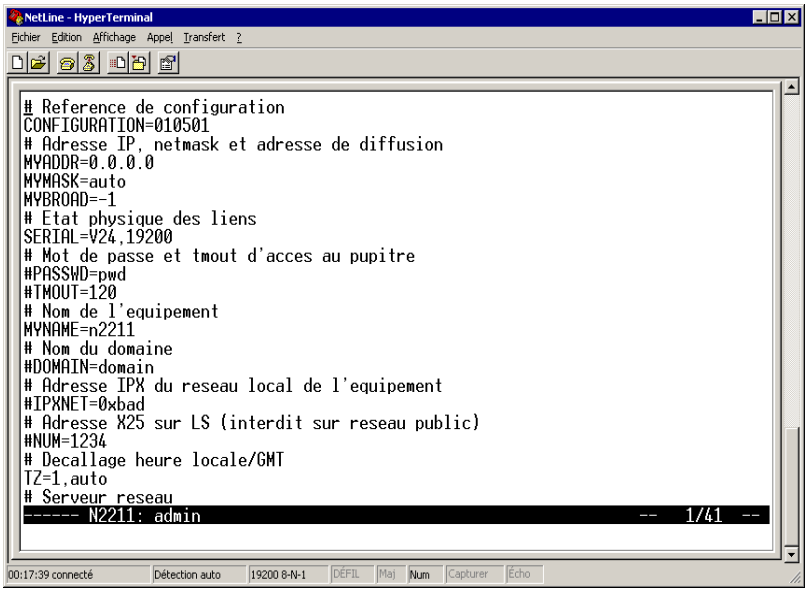
L'équipement indique qu'aucune adresse n'est initialisée.

Vous voici devant le pupitre de l'équipement, il est possible de passer des commandes, nous verrons les principales par la suite, pour l'instant, limitons nous à l'utilisation de l'éditeur. En effet, la première configuration de l'équipement se fait en modifiant quelques fichiers de configuration.

Dans certains cas, comme la configuration simple d'un accès internet, il existe des scripts réalisant la configuration entière de l'équipement, il vous ait possible de les lancer en tapant la commande internet. Vous devrez répondre alors à une série de questions et suivre les indications.

Le programme edit ouvre un fichier de configuration est permet de le modifier. Un des principal fichier est admin ; il contient une grande partie des variables utilisées pour la configuration générale.

Tapez edit admin sur HyperTerminal :



Vous voyez apparaître le contenu de se fichier; les commandes sont les suivantes :

- déplacement du curseur : flèches du clavier ou [ctrl]+p|n|b|f.
- suppression de caractère : delou [ctrl]+d|h.
- ajout d'une ligne en dessous du curseur : entrée ou [ctrl]+j.
- ajout d'une ligne au dessus du curseur : [ctrl]+o.
- suppression d'une ligne : [ctrl]+k en début de ligne.
- suppression de la fin d'une ligne : [ctrl]+k a partir du caractère a supprimer.
- couper/coller : coupez avec [ctrl]+k et collez avec [ctrl]+y .
- quitter l'éditeur : [ctrl]+x avec enregistrement taper **o** sinon **n .**

Nous allons maintenant pouvoir configurer l'équipement...

IV.1.2. Configuration IP de l'équipement

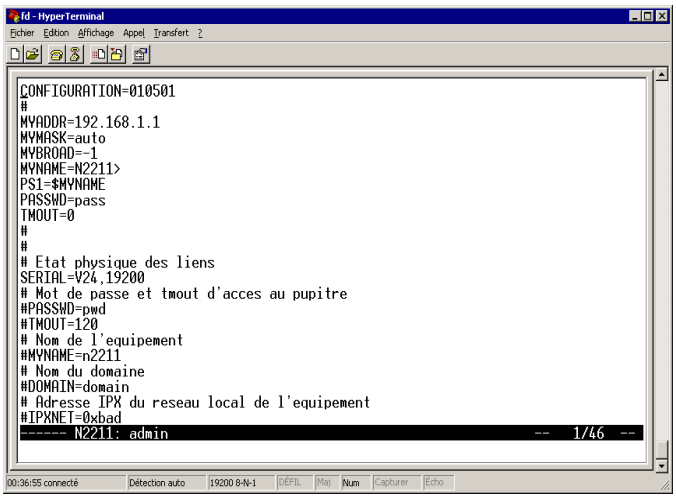
Le fichier admin permet de positionner un certain nombre de variables permettant la configuration de l'équipement. Entre autre, l'adresse IP, le masque ...

- **MYADDR** : est l'adresse IP qui sera utilisée pour le port Ethernet principal de l'équipement.
- **MYMASK** : est le masque associé à cette adresse. Ce peut être auto dans ce cas, l'équipement détermine seul sont masque en fonction de la classe de réseau utilisée.
- **MYBROAD** : définit l'adresse qui sera utilisée pour effectuer les broadcast. On utilise -1 pour que l'adresse dans la partie machine soit 255, 0 pour que ce soit 0.
- **DNS** : définit l'adresse IP du serveur de nom.

D'autres variables sont intéressantes pour la configuration de base :

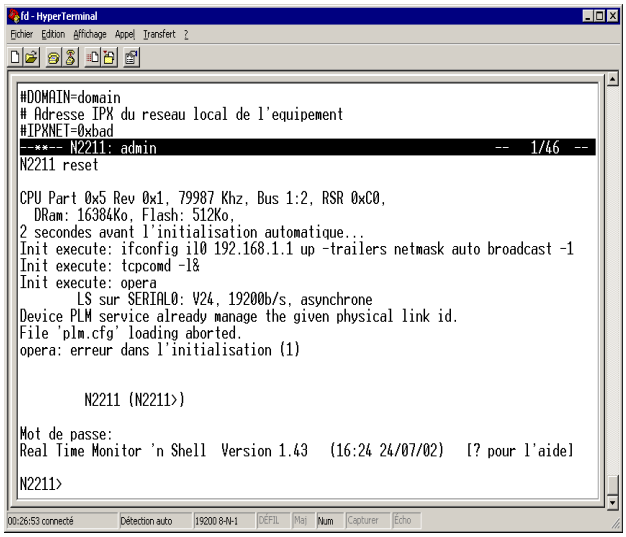
- **MYNAME** : donne un nom symbolique a l'équipement.
- **PS1** : définit la chaîne qui sera utilisé à l'invite de l'équipement.
- **PASSWD** : définit un mot de passe limitant l'accès au shell. Il sera demander pour tout prochain accès à l'équipement.
- **TMOUT** : définit le temps d'inactivité d'une session avant fermeture en seconde. Si elle est non définie ou égale à 0, aucune déconnexion automatique ne sera effectuée.

Voici à quoi peut ressembler le fichier admin après configuration :



Notons, qu'il est possible d'utiliser une variable définie précédemment en ajoutant un **$** devant son nom. Il est aussi possible de commenter une ligne en ajoutant **#** en son début.

Enregistrons le fichier et relançons l'équipement au moyen de la commande **reset** pour qu'il prenne en compte les nouveaux paramètres. Il nous faudra saisir le mot de passe pour ouvrir la console.

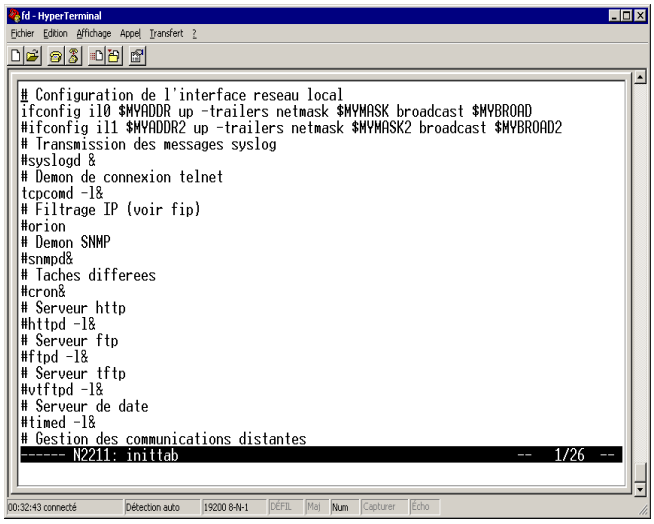


Nous remarquons que l'équipement a pris en compte les modifications effectuées, il indique par exemple :

Init execute: ifconfig il0 192.168.1.1 up -trailers netmask auto broadcast -1

Ligne sur laquelle nous retrouvons les paramètres réseaux choisis.

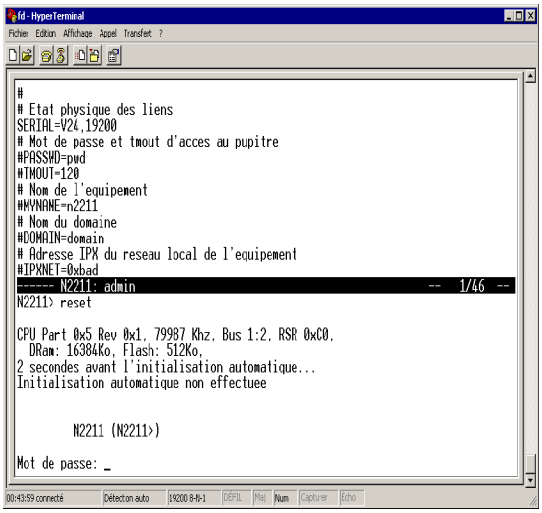
Il est à noter que l'équipement lors de son démarrage lance un certain nombre de commande, indiquées sur la console par "Init exectute: ...". Il s'agit en fait de la trace d'execution d'un autre fichier important du système : **inittab**. Dont voici un extrait :



Nous retrouvons ici les lignes exécutées, nous remarquons aussi que les variables

que nous avons positionnées dans le fichier admin sont ici utilisées. Ce fichier peut être modifié, comme admin, pour, par exemple configurer le second accès Ethernet, en supprimant le commentaire placé en ligne 3.

En cas de problème dans les fichiers de configuration, il est possible que ceux-ci gênent le demarrage correct de l'équipement. Il est alors possible d'en éviter l'exécution en appuyant sur la barre d'espace lors du démarrage du routeur.



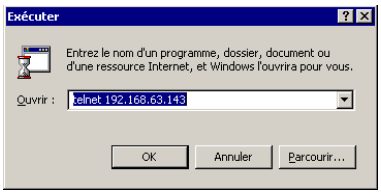
Il est alors indiqué "Initialisation automatique non effectuée".

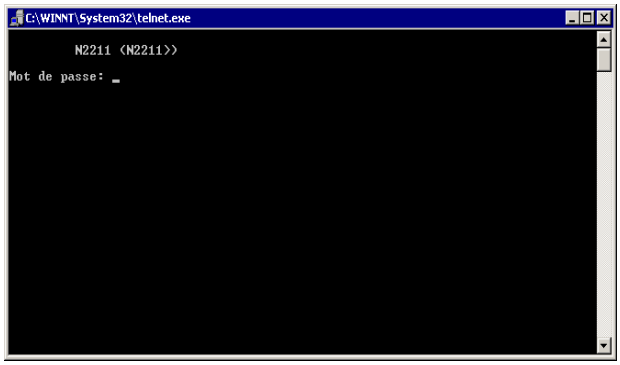
Le fichier "admin" permet de fixer les paramètres de configuration du réseau IP sur Ethernet. Ceux-ci sont utilisés lors de l'exécution des commandes, au démarrage de l'équipement, inscrites dans le fichier "inittab". Ces fichiers sont modifiables à l'aide de la commande "edit" et pris en compte dès le redémarrage suivant, provoqué par la commande "reset".

Note : Il est possible de configurer de façon plus fine les parties IP du routeur. Par exemple, il est possible d'augmenter la taille de sa mémoire réservée au trames à router. Cette fonctionnalité permet d'éviter que des trames soient détruites en cas de saturation momentanée d'une des voies de communication. La variable **IP_QLEN** placée dans admin définit le nombre de paquets qui seront mis en attente, par défaut, ils seront 25.

IV.1.3. Accès à l'équipement

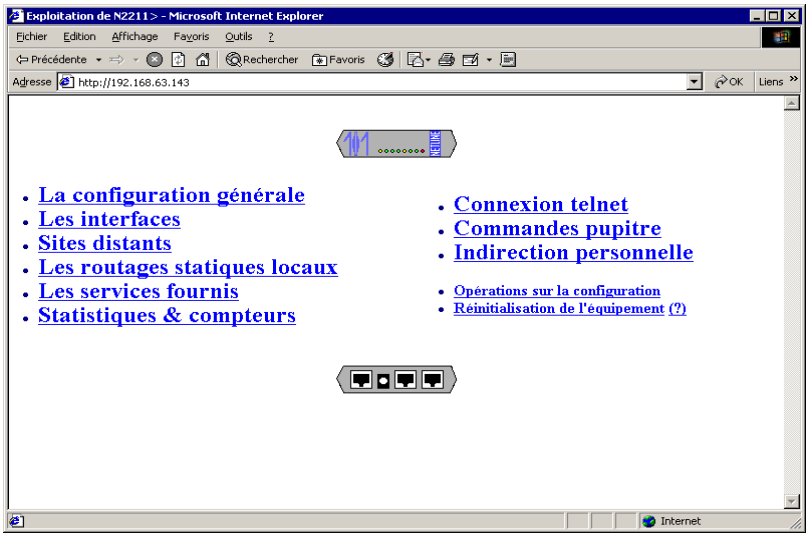
Nous pouvons maintenant accéder à l'équipement par le service telnet au travers du réseau. L'accès par câble série reste toujours possible, mais moins pratique que telnet.





Vous noterez que nous avons modifié l'adresse IP définie précédemment pour les besoins de notre réseau...

L'équipement peut aussi être configuré au travers d'un browser web. Pour cela, il faut lancer le démon http sur le routeur. Tapez simplement la commande **httpd &** sur une console, telnet ou série. Alors, vous pouvez lancer l'explorateur et taper l'url : http://192.168.63.143.



Au travers de différents menus, vous pouvez ainsi accéder à la configuration des différentes parties du routeur.

IV.1.4. Protection de l'accès à l'équipement

Nous avons vu précédemment l'utilisation de la variable **PASSWD** pour définir un mot de passe qui sera utilisé pour l'accès console. Ce même mot de passe est utilisé pour les accès telnet, ftp et autres. Il peut être nécessaire d'avoir recours à une gestion plus évoluée des mots de passe. La variable **PASSWD** recevra alors la valeur '*' indiquant le recours au fichier passwd.

Des comptes utilisateurs sont créés dans le fichier passwd à l'aide de la commande **defuser**. Un utilisateur est associé à un groupe et possède un ID. Grâce à ces informations, IDU et IDG l'utilisateur possède des droits propres, il est alors possible de lui autoriser ou interdire l'accès à certaines fonctions. L'ajout d'un utilisateur se fait donc par l'exécution de **defuser**, mais aussi par l'exécution de la commande **passwd** permettant de lui associer un mot de passe. Il est important de noter que le mot de passe est crypté, contrairement à l'utilisation de la variable **PASSWD** où le mot de passe est visible dans le fichier admin.

Si le fichier passwd n'existe pas sur le système, il doit être créé en tout premier lieu, la commande à utiliser est : **eestore passwd**

Voyons un exemple de création de compte utilisateur :

```
N2211> defuser disk 1 0
defuser: utiliser "passwd" pour mettre un mot de passe
N2211> passwd disk passwd
passwd: le mot de passe de disk est mis a jour
N2211>
```

Le compte disk a été créé avec l'IDU 1 et IDG 0 et a comme mot de passe "passwd". Une entrée a été ajouté au fichier passwd : **disk:d2UnSSZ:1:0:::**

L'utilisateur root est particulier : sont IDU/IDG est à 0. Il est le super-utilisateur. La valeur IDU doit être unique car elle identifie l'utilisateur.

Les droits d'accès aux commandes et fichiers sont figés, c'est donc le choix de l'IDU/IDG de l'utilisateur que va déterminé ses droits :

IDG0 => Donne accès en lecture/écriture aux fichiers de configuration
Donne accès à toutes les commandes, seul l'utilisateur root peut toutefois lancer les démons tels que ftpd ...

IDG1 => Donne accès en lecture aux fichiers de configuration

IDG2 => Interdit l'accès aux fichiers. Permet l'exécution de beaucoup de commandes, mais pas la configuration des interfaces ou de l'équipement de façon générale.

La liste complète des droits est consultable sur le site de Netline.

IV.2. Les fichiers de configuration

IV.2.1. Manipulation

Nous avons vu que la commande **edit** permettait de modifier un fichier. D'autres commande permettent de les créer ou de les supprimer :

- **eestore** => Permet de créer un nouveau fichier, avec un contenu ou sans. La commande est :**eestore** fichier contenu. Fichier est le nom a donner au fichier. Contenu est le contenu du fichier qui sera créé. Ce dernier paramètre est facultatif.
- **eeclear** => Permet d'effacer un fichier existant. La syntaxe est **eeclear** fichier. Où fichier est le nom du fichier à supprimer. Il est possible d'utiliser **eeclear -i** auquel cas tous les fichiers seront concerné s et une confirmation sera demandée pour chacun.

IV.2.2. Description

Nous connaissons déjà deux des fichiers utilisés pour la configuration : admin et inittab. L'équipement utilise d'autres fichiers permettant la configuration de protocoles, ou modules utilisés par le routeur.

- **numip**

  Il est utilisé pour définir les routes établies par autocom. Il sert donc à la configuration des routes RNIS et Frame-Relay. Il est aussi utilisé pour les tunnels et IpSec.

- **hosts**

  Sert a mémoriser une correspondance entre un nom symbolique et une adresse IP. Ce fichier simplifie l'adressage pour l'administrateur. Il opère un peu comme un service DNS, mais les informations se sont accessible que localement ; de plus elles sont statiques et non centralisées. Il s'agit donc d'une aide et non d'une solution globale de gestion des nom symboliques.

  La syntaxe est la suivante :

  #Adresse_IP_de_l'équipement nom_symbolique

  192.168.63.143 monnomsymbolique

- **ppp.conf**

  Permet la configuration des connexion ppp. Les paramètres désirés sont placés ici, ils peuvent être différents pour chaque route crée.

- **secret**
  Contient les mots de passe de connexion pour l'utilisation de chap.

- **trans.conf**
  Configure la translation d'adresse IP.
- **rcmd**
  Gère les droits d'exécution distante de commandes sur l'équipement.
- **ospf.conf**
  Permet la configuration d'OSPF
- **ipsec.conf**
  Permet la configuration d'IpSec
- **cbq.conf**
  Permet la configuration de Qos-Cbq
- **rap.conf**
  Permet la configuration de RIP

IV.3. Configuration des interfaces Ethernet

L'équipement N2211 dispose de 2 interfaces Ethernet, la première (ETH0) offre un débit de 10 ou 100 Mbits/s. Nous avons vu précédemment comment la configurer et allons maintenant revenir là dessus de façon plus générale. Il est en outre possible de créer des interfaces secondaires virtuelles greffées sur les interfaces physique. Ainsi, un équipement peu répondre sur plusieurs adresses IP différentes et router des données entre plusieurs réseaux virtuels établit sur un même réseau physique.

Les interfaces Ethernet se configurent à l'aide de la commande ifconfig. Il est à noté que cette commande configure l'ensemble des périphériques, mais dans la plupart des autres interfaces, son appel est automatisé par des démons comme opéra. Le cas des interfaces Ethernet étant un peu différent, l'utilisateur est amené à manipuler ifconfig qui est exécuté au démarrage par inittab comme nous l'avons vu.

La syntaxe est la suivante (bien sur, les paramètres sont normalement sur la même ligne) :

ifconfig **interface** => indique l'interface configurée { il0, sec0, ter0, qua0, il1, sec1, ter1, qua1}.

[adresse] => adresse IP de l'interface.

[up|down] => respectivement active ou désactive l'interface.

[netmasque] => masque de réseau de l'interface, peut être auto pour une gestion automatique.

[broadcast] => peut être 0 ou -1 et force alors la portion machine a 0 ou 255 pour les trames de diffusion.

[metric] => permet de donner un coût à l'utilisation de l'interface, il sera utilisé pour le choix des routes.

[mtu] => taille maximale admise pour les trames utilisant cette interface.

[trailers] => autorise ou non l'utilisation de l'encapsulation en mode trailer.

[arp|-arp] => autorise ou non l'utilisation d'ARP pour la résolution d'adresse de niveau inférieur.

[multicast|-multicast] => autorise ou non l'utilisation du multicast sur l'interface.

[speed 10|100] => configure la vitesse du port Ethernet à 10 ou 100Mbits (valable pour ETH0 seulement).

[duplex half|full] => configure l'interface en half ou full duplex.

[translation|-translation]=> valide ou invalide la translation d'adresse sur cet interface.

La commande "ifconfig -a " permet de lister la configuration de toutes les interfaces.

Les interfaces physiques sont **il0** et **il1**, les interfaces virtuelles qui peuvent êtres crées sur ces interfaces physiques sont appelée sec,ter et qua, suivit du numéro de l'interface. Ainsi **sec0**est l'interface secondaire de **il0**.

Pour que les interfaces Ethernet soient configurées automatiquement au démarrage de l'équipement, la ligne de commande ifconfig doit être placée dans le fichier inittab. Il sera bien sure possible de fixer les paramètres au travers de variables

d'environnements qui pourront être placées dans admin. La manipulation à faire est la même que celle que nous avons vu dans le chapitre précédent. Vous pouvez créer de nouvelles variables d'environnement à votre guise.

Lors de l'appel à ifconfig, une route statique correspondante est ajoutée dans la table de routage.

La commande ifconfig permet la configuration et l'activation des interfaces Ethernet. La route statique associée est automatiquement ajoutée.

IV.4. Configuration des interfaces séries

Les interfaces séries sont considérées comme des périphériques nécessitant un routage à la demande. Leur gestion est donc assurée par le périphérique autocom. Un processus appelé opéra se charge de configurer ces interfaces et de manager le périphérique autocom.

Il y a donc deux parties dans la configuration de l'interface série : sa configuration matérielle : type de support, vitesse... et sa configuration de routage : quel réseau joindre, comment se connecter. La configuration matérielle est partagé entre deux fichiers admin et numip. La configuration de connexion est principalement liée à numip, toutefois selon le protocole utilisé, celui-ci sera aussi configuré dans un autre fichier.

IV.4.1 Configuration matérielle

Une interface série est donc caractérisée par une norme, un débit et quelques autres informations. La configuration s'effectue à partir de variables d'environnements placées dans le fichier admin. Celle-ci est **SERIAL.** Cette variable comporte 1 ou 2 options séparée par le caractère ','. Tout d'abord est indiquée la norme électrique utilisée, qui peut être **V11 (ou RS422), V24 (ou RS232), V35, RS485 (limité), RS449** ou **EIA530**. Suit la vitesse de transmission souhaitée exprimée en bits/seconde.

Il est possible d'ajouter un **'o'** à la devant la vitesse pour indiquer, en mode synchrone, que l'interface devra générer une horloge.

La syntaxe est donc par exemple :

SERIAL = V24,19200

ou

SERIAL = V11,o64000

Le choix entre une utilisation synchrone ou asynchrone de l'équipement se fait au travers de la configuration effectuée dans numip pour cette même interface série.

La commande **stty** permet une configuration plus fine des liens séries utilisés en mode asynchrone. Il est alors possible de configurer, en plus de la vitesse le nombre de bits par caractère, la façon dont est géré le contrôle de flux... Cette commande étant très complète nous ne la détaillerons pas, mais vous la trouverez dans la documentation en ligne Netline.

L'interface série est configurée à l'aide de la variable d'environnement SERIAL dans admin. Plus généralement c'est la commande stty qui permettra une configuration complète de ce type d'interface utilisé en mode asynchrone.

IV.4.2. Configuration du routage à la demande

La seconde partie de la configuration concerne le routage à la demande, il est effectué dans le fichier appelé numip. D'une façon générale, numip contient les informations nécessaire au routage vers toutes les destinations nécessitant une connexion.

Chaque entrée respecte le format suivant :

adresse_distante = accès_au_distant1:chien_de_garde1[:options1],
[accès_au_distant2:chien_de_garde2[:options2],...]

adresse_distante est l'adresse qu'il est possible joindre par l'accès qui est ici définit. Ce peut être l'adresse d'un hôte, alors la partie "machine" de l'adresse est différente de zéro ou ce peut être un réseau, alors elle est égale à 0. Sans précision particulière, le routeur prend le masque par défaut correspondant à la classe de l'adresse IP pour déterminer la partie "machine" de la partie "réseau". Il est donc possible d'associer à une adresse un masque.

Voyons des exemples :

192.168.1.0 = ... => indique le réseau 192.168.1.xx puisque le masque par défaut est 255.255.255.0 et la partie machine est à 0.

192.168.1.10 = ... => indique l'équipement 192.168.1.10 puisque avec le même masque, la partie machine est différente de 0.

192.168.1.80/0xFFFFFFC0
=> indique un réseau car un masque de sous réseau spécifique est indiqué, il s'agit de 255.255.255.192, qui, ici, devra être représenté sous une forme hexadécimale. Alors, la partie "machine" est égale à 0.

0.0.0.0= ...

defaut = ...	=> ces destinations concernent toutes les trames pour lesquelles nous n'avons pas encore trouvé de chemin. Elles signifient : "toutes les autres destinations"

La seconde partie de la ligne de configuration concerne l'interface, le protocole, le moyen mis en oeuvre pour joindre la destination. Il est possible que plusieurs moyens existent, il est donc possible de lister ces moyens en les séparant par des virgules. Contrairement à ce que la syntaxe ci-dessus laisse penser, il n'est pas permis de passer à la ligne. Si celle-ci venait à être trop longue, il suffirait alors de créer une seconde ligne ayant la même adresse_distante juste au dessous.

Le moyen d'accès_au_distant définit l'interface utilisée et ses paramètres.

- Dans le cas de l'interface série qui nous intéresse ici, ce peut être **Ci**, on identifie ainsi une communication synchrone utilisant la voie physique indiqué par **i**, dans le cas du N2211, nous aurons alors 0. Le protocole utilisé est alors PPP.
- Lors de l'utilisation de la voie série en mode asynchrone, la syntaxe sera **Ci_A**, il est possible de préciser un script de connexion par **Ci_Anomduscript**. Le protocole utilisé sera alors PPP asynchrone
- Lors de l'utilisation de la voie série pour une connexion X25 sur LS, la syntaxe est **Ci_Xadr**, où **adr** est l'adresse X28 à joindre.

Voyons des exemples :

– **192.168.1.0 = C0**	Indique que 192.168.1.xx est joint par l'interface série configurée en mode synchrone avec comme protocole PPP.
– **192.168.1.0 = C0_A**	Indique que 192.168.1.xx est joint par l'interface série configurée en mode asynchrone avec comme protocole PPP asynchrone
– **192.168.1.0 = C0_Acnx1**	Indique la même chose, mais précise l'utilisation d'un script de connexion.
– **192.168.1.0 = C0_X123**	Indique que 192.168.1.xx est joint par l'interface série configurée en mode synchrone sur X25 en appelant **123**.

Le chien_de_garde permet de fixer un délai d'inactivité au delà duquel la communication sera automatiquement fermée. Dans le cas d'une connexion permanente, la valeur 0 sera utilisée. Les lignes séries ne sont généralement pas facturées à la minute d'utilisation, ce paramètre sera donc généralement à 0.

Les options sont assez diverses, plusieurs peuvent être utilisées, alors elles se suivent et sont séparées par un ':'.

- **a** => Indique que cet accès au distant ne peut être utilisé qu'en entrée, il ne sera pas possible d'établir une communication sortante de cette façon.
- **b** => Indique que cet accès au distant ne peut être utilisé qu'en sortie.

- **f** => Indique lors d'une connexion Frame Relay le DLCI à utiliser. Sans précision particulière, celui indiqué par la variable DLCI est pris.
- **p**n => Indique que cet accès, s'il utilise le protocole ppp devra choisir les paramètres de la section n pour la connexion. De sorte, il est possible d'utiliser des paramètres différents pour chaque connexion PPP.
- **t** => Indique que l'on souhaite effectuer sur cette destination une translation d'adresse IP.
- **z** => Indique que l'on souhaite une transmission compressée sur X.25, la variable **PACK_X** permettra de choisir la méthode de compression.

Par exemple, **192.168.1.0 = C0:0:p0:t** indique que l'on pourra joindre 192.168.1.xx en utilisant la voie série, que les options de ppp seront définies dans la section 0 et qu'une translation d'adresse devra être effectuée.

Le routage à la demande utilisé sur les liaisons séries est configuré à l'aide du fichier numip. La syntaxe générale définit pour chaque destination une ou plusieurs route à suivre ayant chacune leur options propres.

IV.5. Configuration des interfaces RNIS

Tout comme pour les interfaces séries, les interfaces RNIS se configurent à plusieurs niveaux, il faut en effet définir des paramètres liés à RNIS et des paramètres liés au réseau.

IV.5.1. Paramétrage RNIS

Un accès RNIS est d'abord caractérisé par un numéro par lequel il est possible de le joindre. Le fait de fixer ce numéro limitera la réception d'appel et ce numéro précis. Il est définit par la variable **NUM**i présente dans le fichier admin. L'utilisation de cette variable n'est utile que lors de l'utilisation de l'équipement derrière un autocommutateur privé. On donne alors le numéro de poste associé à la ligne. i identifie le numéro de canal auquel on associe le numéro, s'il n'est pas précisé, le numéro s'applique à tous les canaux. Il est possible de préciser une sous adresse à l'aide de la variable **SUB**i . Dans le cas d'une liaison X.25 **NUM**i correspond au numéro X.25 local.

Le fichier **isdn.cfg** permet de configurer la liaison physique RNIS. Il est possible de choisir entre plusieurs variantes des protocoles utilisés par RNIS. La variable descriptor peut prendre les valeurs de vn3.ipo, etsi.ipo, tr6.ipo, vn6.ipo et ainsi s'adapter à différents réseaux publiques ou autocommutateurs privés.

Les fichiers **isdn.prf, plm.cfg, plp.cfg** permettent une configuration avancée de l'équipement pour son utilisation RNIS et principalement X.25. Une documentation complète est disponible sur le site de Netline.

Il est possible de demander la compression des données transitant sur RNIS. Cette compression sera fonctionnelle lors d'une communication entre deux routeurs Netline, dès lors que la variable **PACK_B** sera définit dans admin. Cette variable permet de demander une compression LZW, RLL ou LZW rapide. Ce choix se fait au travers de valeurs qui sont respectivement : 0x20/0x40/0x80. LZW est la méthode la plus efficace mais aussi la plus gourmande en ressources processeur, LZW rapide est la méthode la moins performante, mais la plus légère.

Lors d'une utilisation de RNIS sur X.25, la variable utilisée est **PACK_X** et permet la compression des canaux virtuels. Seul le mode N est utilisable, la valeur à choisir est alors 0x20. Cette option s'utilise conjointement avec l'option **z** dans numip.

IV.5.2. Configuration du routage à la demande

Tout comme pour les liaisons séries, le routage à la demande est configuré au travers du fichier numip. Nous ne reviendrons pas sur la syntaxe générale définie au chapitre IV.4.2.

Le moyen d'accès_au_distant peut être :

- **un numéro** => Dans ce cas, on indique le numéro à composer pour joindre l'équipement distant. Le numéro peut être précédé du caractère '**.**' pour indiquer la possibilité d'utiliser un préfixe d'opérateur. Un caractère '*' permet de séparer l'adresse de la sous-adresse dans le numéro.
- **B_nnn_xxx** => Dans ce cas, on indique une connexion X25 dans B où nnn est le numéro RNIS distant et xxx l'adresse X25 distante.
- **C1_Dxxx** => Dans ce cas, on indique une connexion X25 dans D où xxx est l'adresse X28 du destinataire.
- **tous** => Dans ce cas, tous les appels entrant seront acceptés.

Dans le cas d'un numéro, le protocole utilisé sera PPP.

De nouvelles options sont utilisables, **a,b,p,t** vues précédemment sont aussi applicables aux connexions RNIS.

- **c** => Indique que cet accès peut être utilisé en serveur de Call-back (ou rétro-appel). Le rétro-appel est utilisé principalement pour des raisons de facturation des appels. Lorsqu'un serveur de rétro-appel reçoit un appel demandant son utilisation, il identifie l'appelant puis clôture l'appel et enfin établie lui même une nouvelle connexion. Alors, l'appel émanant du serveur lui sera facturé. Le retro-appel mis en oeuvre par l'option **c** fonctionne plus précisément de la façon suivante :

Un équipement A appelle le serveur.

Le serveur répond et débute une connexion PPP avec authentification CHAP.

Si l'authentification est correcte, le serveur mémorise le login reçu .

Ce login doit correspondre à une adresse IP dans le fichier hosts.

Le serveur ouvrira une communication à destination de l'adresse IP indiquée.

Le procédé de rétro-appel, permet une sécurisation des accès, et une répartition coté serveur des coûts.

- **r** => Indique l'utilisation de l'accès comme serveur de rétro-appel de type RSD. Le principe est différent du précédent, car l'appel reçu par le serveur est directement rejeté : il n'y a donc pas de connexion et le coût pour l'appelant est alors nul. Le serveur rappelle automatiquement le demandeur ; ce dernier doit être explicitement identifié dans la première parti de l'accès. Cette méthode peut être interdite d'utilisation dans certains pays, il faudra donc veiller à s'informer de la lois.
- **CB** => Indique l'utilisation du rétro-appel en temps que client.
- **s** => Indique que l'accès concerné est utilisé en secours.

Il est à noter que lors de l'utilisation de liaison RNIS, il peut arriver que le numéro présenté lors de la réception d'un appel diffère de celui composé pour l'émission, alors numip devra avoir deux entrées pour l'accès au distant : un premier limité en entrée et un second limité en sortie avec les numéro adéquats. Il se peut toutefois, dans un tel cas que l'utilisation du caractère "." en temps que joker soit une solution.

Lors de l'utilisation d'un routeur derrière un autocommutateur sur lequel il serait nécessaire de composer un préfixe pour les appels sortant, il est conseillé de configurer la variable **AUTOCOM**. Celle-ci se trouve dans le fichier admin et définit le préfixe qui sera utilisé.

Si **AUTOCOM=00** et que dans numip on a xxx = 2244:10, le numéro composé sera 002244

Il est possible de préciser une taille minimale pour laquelle se préfixe sera ajouté alors :

Si **AUTOCOM=00,5** et que dans numip on a xxx = 2244:10, le numéro composé sera 2244,

Par contre si on a dans numip on a xxx = 12244, le numéro composé sera 0012244

L'utilisation du chien_de_garde est importante en RNIS, car la communication est facturée en fonction du temps, alors toute connexion restant active de façon superflue entraîne un surcoût . Il faut toutefois se méfier de la temporisation choisie, en effet, l'utilisation d'une temporisation trop longue entraînera un surcoût. De même l'utilisation d'une temporisation trop faible peut entraîner elle aussi un surcoût qui peut être plus important :

Généralement, les opérateurs facturent au minimum une unité complète, celle-ci peut représenter plusieurs minutes de communication. Alors, si une temporisation trop courte vient stopper la communication et que quelques instant plus tard de nouvelles données viennent la ré-ouvrir, plusieurs unités seront facturées. Ainsi, avec une temporisation de 10 secondes, et une unité durant 180 secondes, on pourrait avoir une facturation de 18 unités au lieu d'une seule pour le même trafic.

IV.6. Configuration des interfaces virtuelles

Dans ce chapitre, nous allons voir la configuration des périphériques virtuels que sont les tunnels et IpSec. En effet, leur configuration est assez proche des précédentes réalisée dans le fichier numip. IpSec demandera une configuration complémentaire dans un fichier dédié : ipsec.conf.

Les interfaces virtuelles que sont les tunnels reposent sur l'utilisation d'interfaces physiques pour transporter les données émises par le tunnel. Ainsi, il doit exister dans numip une entrée configurant le tunnel et une entrée indiquant au routeur comment transmettre les données de ce tunnel. C'est à dire une entrée indiquant comment joindre l'autre bout du tunnel.

IV.6.1. Configuration des tunnels

Les tunnels sont configuré dans numip simplement en utilisant la syntaxe suivante :

réseau_distant = **unl_**ipdistante:chien_de_garde:**gre**

Le paramètre ipdistante est l'adresse IP de l'autre bout du tunnel, ce peut être l'adresse IP publique obtenue lors de la connexion à Internet. Le chien de garde doit avoir une durée de vie inférieur à la connexion supportant le tunnel. L'option **gre** indique l'utilisation d'un tunnel respectant le protocole du même nom.

Lorsque l'on souhaite utiliser plusieurs tunnels simultanément, il faut indiquer dans la variable **TUNLCS** leur nombre. Cette variable peut être définie dans le fichier admin.

Les tunnels nécessitent simplement leur déclaration dans le fichier numip. Il faudra veiller à ce qu'une entrée "support" existe bien, permettant de joindre l'autre extrémité du tunnel.

IV.6.2. Configuration des tunnels IpSec

Le tunnel IpSec s'utilise de la même façon qu'un tunnel classique, alors la ligne à ajouter dans le fichier numip se présente sous la même forme. Toutefois, l'option gre sera remplacée par **ipsec**. La syntaxe est donc la suivante :

réseau_distant = **unl_**ipdistante:chien_de_garde:**ipsec**

Par exemple : 192.168.1.0 = unl_62.4.18.45:60:ipsec

Le réseau 192.168.1.xx est joignable au travers d'un tunnel de type IpSec en passant par le routeur 62.4.18.45.

IpSec nécessitant une configuration plus avancée, entre autre pour le choix des méthodes de cryptage/authentification, un second fichier est utilisé en complément de numip. Ce fichier est ipsec.conf.

ipsec.conf décrit tous les tunnels IpSec, chaque entrée commence par l'ip distante concernée. Suivent les paramètres. La forme sera donc la suivante :

ipdistante1

param1 : val1

param2 : val2

…

ipdistante2

…

Les paramètres peuvent être pour le transport des données :

– **protocole** => Spécifie les protocoles utilisés pour véhiculer les informations.

La valeur peut être **esp** (cryptage + authentification) et/ou **ah** (authentification).

L'utilisation des deux valeurs (alors séparées par un espace) indique que le fonctionnement dans les deux modes est accepté, la première valeur sera choisie en priorité.

Ce paramètre est obligatoire et doit être placé en premier.

– **hash_ah** => Lorsque **ah** a été choisi comme **protocole**, ce champ est **obligatoire** et définit la méthode qui sera utilisée. Les valeur peuvent être : **md5**et/ou **sha**. Il est aussi possible de mettre les deux paramètres, la priorité sera comme précédemment décroissante.

– **hash_esp** => Lorsque **esp** a été choisi comme **protocole**, ce champ est **obligatoire** et définit la méthode qui sera utilisée pour authentifier les données. La liste de valeurs possible est la même que pour **hash_ah**.

– **crypt_esp** => Lorsque **esp** a été choisi comme **protocole**, ce champ est **obligatoire**, il définit la méthode qui sera utilisée pour crypter la communication. Les valeurs peuvent être **3des_cbc des_cbc** et/ou **cast128**. Comme précédemment, il est possible de placer plusieurs choix par ordre de préférence.

D'autres paramètres servent à définir la liaison :

– **life** => Ce paramètre détermine la durée de vie des clefs du tunnel. Au terme du temps indiqué ici, le tunnel est détruit puis automatiquement recréé, les clefs sont modifiées. Cette modification des clef permet d'accroître la sécurité de la liaison. La valeur est un nombre suivit d'une lettre (h,m ou s) exprimant une durée en heures (h), minutes (m) ou secondes (s). Par exemple :

life : 1200 s indique 1200 secondes.

– **reseau_local** => Indique notre réseau local et son masque associé. Ce paramètre est **obligatoire**. Par exemple :

reseau_local : 192.168.70.0/24

Indique que le réseau local a pour adresse 192.168.70.xx. Ici, le masque de sous réseau est représenté en indiquant le nombre de bits à 1. Donc 24 signifie 255.255.255.0. 255.0.0.0 s'écrirait 8.

– **reseau_distant**=> Indique de la même façon le réseau distant et son masque. Ce paramètre est **obligatoire**.

Enfin, d'autres servent à paramétrer l'échange des clefs :

– **session** => Indique la durée de vie des clefs utilisées pour l'échange des informations de cryptage du flux. Cette durée est exprimé en nombre de renégociation. La valeur conseillée est 2.

– **timeout** => Indique un temps avant retransmission des trames de négociation en cas de non réponse du destinataire. La valeur conseillée est 5 et s'entend en seconde.

– **retry** => Indique le nombre de trames renvoyées en cas de non réponse avant de stopper les tentatives de communication.

- **group** => Indique la forme des clefs qui seront calculées pour l'échange des informations de cryptage du flux. La valeur pourra être : **modp768** ou **modp1024** ou **modp1536**. Le nombre indiquant combien de bits sont utilisés. Plus il est grand et plus la sécurité est importante. Le temps de calcul nécessaire devient par contre, lui aussi, plus important. Ce paramètre est **obligatoire.**
- **auth_method** => Indique la méthode utilisée pour l'échange des paramètre. Seule **psk** est actuellement possible. Ce paramètre est **obligatoire.**
- **psk_str** => Est la clef partagée entre les deux points extrémités du tunnel qui permet de générer les autres clefs nécessaire au cryptage. Ce paramètre est donc la clef de toute la sécurité qui sera mise en oeuvre. Il s'agit d'une chaîne de caractère débutant et se terminant par '"'. Ce paramètre est **obligatoire.**

Voici un exemple de fichier de configuration :

```
62.4.18.45
protocole : esp ah
hash_esp : sha
crypt_esp : 3des_cbc des
hash_ah : sha md5
life : 10 h
reseau_local : 192.168.2.0/24
reseau_distant : 192.168.1.0/24
group : modp1024
session : 2
timeout : 5
auth_method : psk
psk_str : "ma clef secrete"
```

Un script de configuration interactif est aussi accessible au travers de la commande **script ipsec** qui peut être taper sur la console de l'équipement.

La fonctionnalité IpSec est une option des routeur N2211 nécessitant une clef d'activation à commander.

IpSec se configure comme un tunnel classique mais possède en plus un fichier dédié : ipsec.conf dans lequel sont définit ses paramètres. Un script interactif permet de simplifier la configuration. Ipsec est optionnel sur les routeurs et demande une activation.

Note : IPSEC précalcule certaines clefs au démarrage de l'équipement. En effet, leur calcule ne peut se faire en temps réel, pénalisant les ouvertures de connexions IPSEC. La variable **KEYSERV** permet de définir le nombre de clefs que l'on veut calculer par avance, il est recommandé de l'initialiser à 2 fois le nombre de tunnels ipsec possibles. La variable **KEYLEN** indique la taille des clefs qui seront ainsi calculées, cette information dépend de la configuration choisie dans ipsec.conf, les valeurs sont 768,1024 ou 1536. Ces variables sont définies dans le fichier admin.

IV.7. Configuration ADSL

Bien que l'utilisation des routeurs Netline pour l'ADSL n'ait pas un lien direct avec les VPN, l'utilisation de protocoles PPTP ou PPPOE entre le routeur et le modem ADSL Ethernet n'est autre qu'un procédé de tunneling. Ainsi, la configuration à ajouter dans numip pour l'utilisation pptp est de la forme :

réseau_distant = **pptp_**adresse_modem:chien_de_garde:t

L'adresse_modem etant l'adresse IP du modem sur le réseau local.

La connexion peut être réalisé au travers d'une connexion pppoe, alors la forme est la suivante :

réseau_distant = **pppoe**_x.:chien_de_garde:t

x est alors l'interface réseau sur laquelle le modem ADSL est connecté. 0 pour ETH0 ; 1 pour ETH1.

L'option t indique une translation d'adresse, lors d'un accès Internet, elle est généralement précisée, raison pour laquelle elle est indiquée ici. Toutefois, si la translation n'est pas nécessaire, cette option sera retirée.

Les connexions ADSL nécessitent une configuration ppp que nous verrons dans le chapitre suivant. Il est à noter qu'un script simplifie les connexions Internet. Il est accessible par la commande **script internet**.

Les routeurs Netline supportent, selon les modèles, l'encapsulation de trames à destination d'un modem ADSL. Les protocoles PPTP et PPPOE sont supportés et nécessitent une configuration dans numip.

IV.8. Configuration PPP

De nombreux accès utilisent PPP pour établir la connexion et transporter les données IP. Nous avons vu précédemment le fonctionnement de PPP, nous allons maintenant étudier sa configuration pour les routeurs Netline. Le fichier ppp.conf contient la liste des options de ce protocole. Ce fichier contient une section globale sans nom indiquant les options commune à toutes les sections contenues dans le fichier. Il contient ensuite de façon facultative des sections. Celle-ci permettent de configurer des paramètres différents pour chaque connexion ppp décrite dans le fichier numip. Rappelons que la section est identifiée dans numip par l'ajout d'une option **p**n (cf IV.4.2). Chaque ligne du fichier indique une option ; le caractère **#** en début de ligne indique un commentaire.

Le fichier est de la forme :

```
#options globales
option1
option2
...
#options s'appliquant au profil numéro 0
[0]
option3
...
#options s'appliquant au profil numéro 1
[1]
option4
...
```

Les options 1 et 2 s'appliquent aux profil 0, 1 et aux connexion sans profil. Les options 3 et 4 s'appliquent seulement aux profil 0...

Le protocole PPP se configure dans le fichier ppp.conf qui contient des options globales et permet de définir des profils de connexions.

Certaines options définissent le comportement de PPP : le mode actif ou passif permet d'engager, ou d'attendre l'engagement par le tiers, de la négociation. Les options sont :

- **active** => Force le mode actif, la négociation sera débutée par l'équipement.
- **passive** => Force le mode passif, alors la négociation sera à l'initiative du distant.

D'autres permettent de choisir, pour l'authentification, l'utilisation de PAP ou CHAP, alors il est possible de préciser un login (et password).

- **+chap** => Impose l'utilisation de la méthode CHAP pour l'authentification.
- **-chap** => Interdit l'utilisation de la méthode CHAP pour l'authentification.
- **chaprange** min max => Fixe les bornes min et max lors du challenge CHAP. (voir doc Netline).
- **myname** name => Indique le login à utiliser pour la connexion CHAP et remplace le login par défaut définit par le couple $MYNAME.$DOMAIN.
- **-pap (-ua)** => Interdit la négociation par PAP.
- **+pap (+ua)** => Force la négociation par PAP.
- **ppp_user** user => Indique le login qui sera utilisé par PAP. Ce login remplace celui par défaut définit par la variable $PPP_USER.
- **ppp_pwd** pwd => indique le mot de passe qui sera utilisé par PAP. Il remplace celui par défaut définit par la variable $PPP_PWD.

Lors de connexion à un site distant, il peut être nécessaire de recevoir une adresse IP dans le réseau où l'on se connecte, auquel cas, de nouvelles options peuvent être activées :

- **autotoute** => Détermine de façon automatique l'adresse IP à utiliser lors de la connexion. Pour plus de détails, il est conseillé de se référer à la documentation en ligne.
- **+sa** => Propose d'utiliser l'adresse IP indiquer dans le fichier numip. Ceci permet d'attribuer des adresses IP aux équipements distants se connectant, en fonction de leur identification par numéro d'appelant. Le fichier poolip permet en plus d'attribuer des adresse extraites d'un pool.
- **+smip** => Transmet au distant notre adresse IP. Cette option est nécessaire pour le fonctionnement face à certains équipements.

Lors de l'utilisation d'une liaison avec rétro appel, certaines options peuvent être utiles :

- **callwait** => Cette options est utilisée en mode call-back et exprime le temps (en 1/50èmes de secondes) qu'il faut attendre entre la réception de l'appel demandant le rétro-appel et sa réalisation effective.
  Lorsque ce délai est trop court, il est possible que la lignes occupée par l'appel initiateur ne soit pas encore libéré lors du rétro-appel, auquel cas celui-ci peut échouer. Indiquer ici une attente de quelques secondes est donc important.

Lorsque la liaison PPP est établie sur une ligne RNIS, il est possible de grouper

des canaux pour augmenter la bande passante disponible. Les routeur Netline utilisent un système "intelligent" permettant de contrôler l'ajout et la suppression de canaux en fonction des besoins instantanés :

- **mp** => Active les fonctionnalités de débordement : agrégation de canaux.
- **mpssnh** => Demande et accepte l'utilisation de numéro de séquence courts.
- **mpminnchan** n => Indique par n le nombre de canaux minimum à ouvrir vers une destination.
- **mpmaxnchan** n => Indique par n le nombre de canaux maximum à ouvrir vers une destination.
- **mpbasenchan** n => Indique par n le nombre de canaux à ouvrir lors de la connexion.
- **mpseuilhaut** n => Indique par un entier exprimant un pourcentage n le seuil d'utilisation à partir duquel un nouveau canal est ouvert.
- **mpseuilHhaut** n=> Indique par un entier exprimant un pourcentage n le seuil à partir duquel tous les canaux sont ouverts par tranche de **mpinc** canaux.
- **mpinc** n => Indique par n le nombre maximum de canaux ouverts lorsque le seuil mpseuilHhaut est atteind.
- **mpseuilbas** n => Indique par un entier exprimant un pourcentage le seuil a partir duquel un canal doit être libéré.
- **mpseuilBbas** n => Indique par un entier exprimant un pourcentage le seuil a partir duquel on revient à **mpminnchan** en fermant les canaux par lot de **mpdec** .
- **mpdec** n => Indique le nombre de canaux qui seront fermés en une fois lorsque est atteint **mspseuilBbas.**
- **mpdebin** n => Indique par n qui prend pour valeur 0 ou 1 si lors d'un appel sortant il est autorisé de répondre à une demande de débordement entrante sur la même liaison.
- **mpbwsampling** n => Indique par n entier la période d'échantillonnage, en seconde, permettant d'exprimer le débit de la ligne.
- **mpbxchgdelai** n => Indique par n entier la période, en seconde, pendant laquelle, suite à l'ajout/suppression de canaux, plus aucun ajout/suppression ne sera fait.
- **mppmgt** => Autorise l'accès à l'équipement à partir d'un gestionnaire travaillant au dessus de MP+.

Une option permet la surveillance de l'état de la ligne :

- **-kap** I R C => Test la ligne en permanence la connexion et la ferme si elle ne fonctionne plus. L'option I permet de définir un intervalle de temps entre 2 tests exprimée en seconde. R définit le nombre d'essai infructueux avant la prise de décision. C définit si après une déconnexion on demande automatiquement une nouvelle connexion (la valeur est 0 ou 1). Note : I,R,C sont remplacés, dans cet ordre strictement par leur valeur.
  Exemple : **kap 10 5 1** signifie l'activation d'un test
  toutes les 10 secondes, avec 5 essais en cas de non réponse et une réouverture de la communication après coupure.

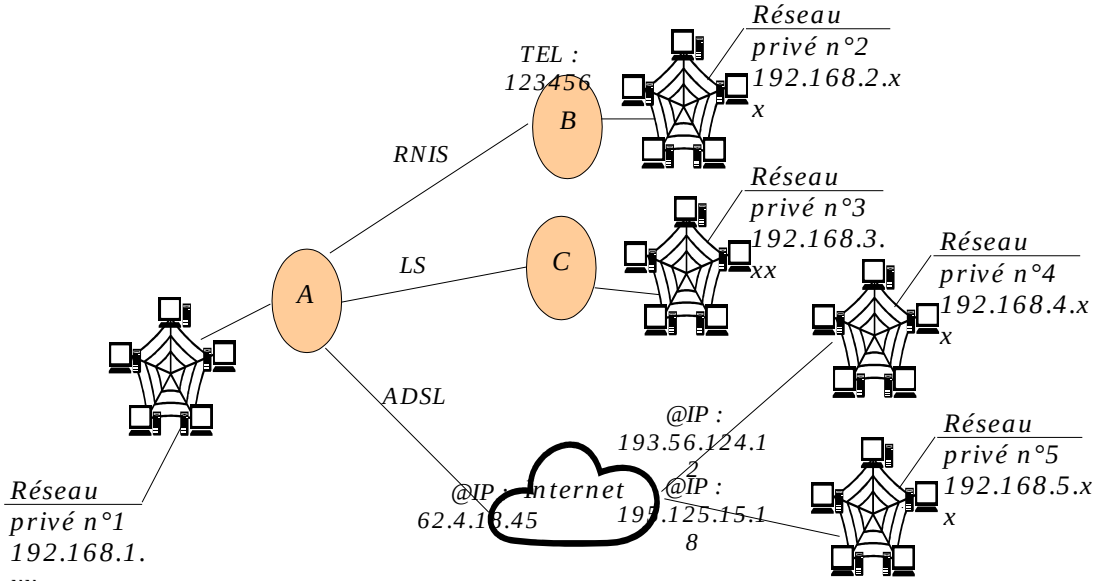
Enfin, certaines options sont utilisées pour rendre l'équipement plus compatibles à ses congénères, elles permettent d'activer ou non des options de PPP principalement liées à la compression :

- **-all** => Invalide toutes les options, y compris celle par défaut.
- **-ac** => Interdit la compression de champs de contrôle et adresse.
- **-am** => Invalide la négociation des tables de caractères précédant les séquences d'échappement.
- **-as** val => positionne la table de caractères précédant les séquences d'échappement. Val indique quel caractère prendre en représentant par décalage le caractère à utiliser (cf Documentation Netline).
- **-cipx** => Interdit la compression d'entêtes IPX.
- **-d** => Valide l'option de débuggage
- **-mn** => Interdit la négociation du magic number.
- **-mru** => Interdit la négociation de la taille maximum des trames en réception.
- **mru** n => Autorise la négociation de la mru, à la valeur n.
- **-pc** => Interdit la compression de champ protocole des trames PPP.
- **-vj** => Interdit la compression Van Jacobson des entêtes TCP/IP.
- **vjold** => Autorise la compression respectant l'ancienne RFC.
- **vj1172** => Autorise la compression respectant la RFC 1172.
- **vj1332** => Autorise la compression respectant la RFC 1332.

Note : Lors de l'utilisation de pap comme procédé d'authentification, le login et le password qui seront utilisés pour se connecter au site distant depuis le routeur sont ceux indiqués par la variable **PPP_USER/PPP_PWD** ou par les options **ppp_user/ppp_pwd**. Lors de connexions entrantes, les login et password utilisés pourront être placés dans le fichier passwd**.** Vous pourrez consulter le chapitre IV.1.4.

IV.9. Exemple

Nous allons maintenant voir une application de la configuration de communications pour un routeur Netline, ainsi nous verrons une synthèse de l'utilisation des fichiers numip, ppp.conf et ipsec. La topologie sur la quelle nous nous basons est la suivante :



Le réseau décrit est donc composé de 5 réseaux distincts + le réseau Internet. 3 routeurs sont utilisés (au minimum), Ils sont noté **A,B,C** sur le schéma. Nous allons examiner la configuration du routeur **A**.

Tout d'abord, le fichier numip doit contenir les routes vers les réseaux 2 et 3, vers Internet et les tunnels vers les réseaux 4 et 5. Le tunnel vers le réseau 5 sera de type IpSec.

Fichier numip :

 # route rnis vers le réseau 2.

192.168.2.0 = 123456:60:p0

 # route LS vers le réseau 3.

192.168.3.0 = C0:60:p1

 # route tunnel vers le réseau 4.

192.168.4.0 = unl_193.56.124.12:500:gre

 # route IpSec vers le reseau 5.

192.168.5.0 = unl_195.125.15.18:500:ipsec

 # route Adsl vers Internet.

defaut = pppoe_0.:1000:p2

Maintenant que les routes sont établies, il faut configurer le fichier ppp.conf pour indiquer les options de chaque connexion, ici, pour l'exemple, nous utiliserons un profil par connexion de sorte à voir des cas différents :

192.168.2.0 demande une connexion de type CHAP et supporte l'agrégation de canaux : un second canal sera ouvert si plus de 60% de charge est détecté sur une période de 10 secondes. Il sera refermé si l'on tombe en dessous de 10%.

192.168.3.0 demande une connexion de type PAP.

Internet utilise PAP.

Fichier ppp.conf :

```
# profil global
-vj
# profil vers 192.168.2.0 : CHAP  MP+
[0]
+chap
myname routeurA
mp
mpminnchan 1
mpmaxnchan 2
mpbasenchan 1
mpseuilhaut 60
mpseuilbas  10
mpbwsampling 10
# profil vers 192.168.3.0 : PAP
[1]
+pap
ppp_user routeurA
ppp_pwd mypassword
# profil vers internet sur ADSL
[2]
+pap
kap 10 3 1
ppp_user mon_login_isp
ppp_pwd mon_password_isp
```

Pour la connexion CHAP, le mot de passe doit être inscrit dans le fichier secret. Il est associé au nom de l'équipement distant qui, si l'on respect la même norme sera routeurB. Alors le fichier secret contiendra :

 # connexion vers B

routeurB mot-de-passe-cnx-A-B

Il nous reste à configurer ipsec.conf pour le tunnel sécurisé vers le réseau 5.

Fichier ipsec.conf

195.125.15.18
protocole : esp
hash_esp : sha
crypt_esp : 3des_cbc des
life : 10 h
reseau_local : 192.168.1.0/24
reseau_distant : 192.168.5.0/24
group : modp1024
session : 2
timeout : 5
auth_method : psk
psk_str : "ma clef secrete"

V. Configuration avancée

V.1. Utilisation des routages RIP et OSPF

V.1.1. Routage RIP

Nous avons vu le fonctionnement du routage par RIP dans un précédent chapitre, nous allons maintenant voir sa mise en oeuvre. RIP demande l'exécution d'un processus gérant le protocole d'échange et de mise à jour des table de routage. Son nom est **rapd**, pour être lancé de façon automatique au démarrage, il suffit d'insérer son lancement dans le fichier inittab. **Attention :** ce démon doit être lancé après opera. Il faudra donc vérifier dans inittab que cet ordre soit respecté.

Lorsque RIP est en cours d'utilisation, l'exécution de la commande **rapd -l** permet de lister les informations de fonctionnement : les statistiques, les autorisations, le statut des interfaces, les tables de routages.

Le fichier rap.conf permet de configurer RIP sur les équipements Netline. Ce fichier permet de définir des restrictions et des options de fonctionnement.

Les restrictions permettent pour chaque action de rapd de limiter son action à certaines adresses IP sur certaines interfaces. Alors la syntaxe est la suivante :

commande **adresse_ip** intf {**all | lan | wan**}

La liste des commandes possible est la suivante :

- **listen/donotlisten** => accepte/refuse les informations venant de l'adresse indiquée.
- **sendto/donotsendto** => accepte/refuse les envois RIP vers cet adresse.
- **announce/donotannounce** => annonce/n'annonce pas cette adresse.
- **record/donotrecord** => enregistre/n'enregistre pas le routage pour cette adresse.
- **donotage** => ne vieillit pas cette adresse

l'**adresse_ip** indique pour quelle adresse IP s'applique la commande.

l'interface concernée suit le séparateur **intf** ce peut être **lan** pour le réseau local ou **wan** pour le réseau distant. **All** indique à la fois lan et wan.

Attention, si **listen,** par exemple, est utilisé, toutes les adresses à écouter devront être spécifiée, la commande **donotlisten** n'aura plus d'effet : l'ordre est prioritaire sur le contre-ordre.

Le fichier rap.conf permet aussi de configurer les options de fonctionnement. Les possibilité sont :

- **noripin** => Interdit la réception de trames RIP.
- **quiet** => Interdit la transmission de trames RIP.
- **noreflect** => Interdit la rediffusion sur l'interface d'où la route a été connue.
- **debug** => Affichages des log d'erreur via syslog.
- **direct** => Les trames ne seront émises que vers les routeurs indiqués par **ripgateway**.
- **ripgatewayadr**=> Demande l'envoie de trames RIP vers l'équipement d'adresse adr. Cet envoie sera en plus des envoies par diffusion.
- **adv** xxx => Demande la publication de la table de routage, en plus des transmissions normales lors de la connexion, de la déconnexion ou de tout changement d'état (cnx+dcnx). Alors xxx est, respectivement : **onconnection, ondisconnection, onchange.**
- **brec** n => Indique, par n, la taille du tampon de réception des trames.
- **itrame** t => Indique le temps t exprimé en ms séparant l'émission de 2 trames d'annonces. Cette option permet de corriger d'éventuels problème de saturation lié aux annonces.
- **version** v => Sélection de la version de RIP : 1 ou 2.

Note : certaines interfaces comme les liaisons séries ne seront pas annoncées si leur adresse IP reste celle attribuée par défaut : 0.0.0.x. Il est donc nécessaire de leur attribuer une adresse IP correspondant à celle du routeur distant : ceci est effectué par l'utilisation de la variable FR0=Adresse_du_distant**, placée dans le fichier** admin.

V.1.2. Routage OSPF

Tout comme RIP, OSPF utilise un démon et un fichier de configuration associé à ce démon. Le service OSPF est lancé et administré au travers de la commande **ospfconf**. Attention, toutefois, OSPF est un élément optionnel des routeurs Netline nécessitant une activation. Le démon est lancé par la commande **ospfconf -s** qui sera placée dans le fichier inittab pour rendre son exécution automatique.

Cet exécutable peut être utilisé pour connaître l'état du démon et les informations qu'il détient :

- **-A** => Affiche toutes les informations.
- **-d** => Affiche les informations sur les aires (zone OSPF dans laquelle les mises à jour sont diffusées).
- **-e** => Affiche les informations sur les liens externes (routes statiques).
- **-g** => Affiche la configuration globale.
- **-i** => Affiche la liste des interfaces de communication.
- **-l** => Affiche la liste des LSA (ensemble de toutes les informations utilisées pour le routage).
- **-n** => Affiche la liste des entités OSPF voisines par interface (à ajouter à l'option -i).
- **-r** => Affiche la table de routage OSPF.

Enfin, il est possible d'inter-agir sur l'exécution du démon à l'aide d'options :

- **-q** => Stoppe le démon OSPF.
- **-R** => Redémarre le démon OSPF.

Le fichier opsp.conf contient les informations nécessaires à la configuration d'OSPF. Ce fichier est composé de sections contenant des variables associées à une valeur (séparées par le caractère **=**) :

– **[global]** => Doit contenir la variable **router** dont la valeur est une chaîne au même format qu'une adresse IP indiquant un identifiant unique pour l'entité OSPF.

– **[external]** => Optionnelle, elle décrit les routes statiques qui seront utilisées par OSPF. Une section définit 1 seule route statique. Il faudra autant de section que de route a définir.

Les variables sont :

addr = adr_ip => Définit l'adresse du réseau distant.

mask= mask => Définit le masque du réseau distant.

type = val => Indique par **1** que le métrique est de même type que pour l'interface. Si la valeur est **2**, l'unité est différente, alors le coût sera géré de façon spécifique.

metric = val => Coût de la route

Il est possible de remplacer **addr** et **mask** par **name**, qui indiquera alors le nom de l'interface concernée (**il0, il1, sl0, cB0 ...**).

[area] => Doit contenir la variable **area** dont la valeur est une chaîne au même format qu'une adresse IP indiquant l'identifiant commun à tous les routeurs d'une même area. Un réseau OSPF doit toujours intégrer une area backbone dont l'identifiant sera **0.0.0.0.**

La variable **externalroutingcapability** prend pour valeur **0** ou **1** et indique s'il est autorisé de sortir de l'aire dans laquelle se trouve le routeur (Nécessaire pour les routes statiques).

[interface] => Sous-section de l'area elle indique la configuration d'une interface physique.

Les variables sont :

name = val => le nom de l'interface physique (**il0,sl0,cB0...**).

type = => le type de cette interface (**nbma, bcast, ptp, ptm, vlink**).

nbma => lan sans fonctionnement par diffusion.
bcast => lan avec diffusion.

ptp => liaisons point a point (LS).

ptm => liaisons point à multipoint (non supporté).

vlink => lien virtuel reliant l'area au backbone. (non supporté).

cost = val => Coût d'utilisation de l'interface.

hellointerval = v => Indique le temps entre deux envoies de trames Hello (par défaut 10), en secondes.

routerdeadinterval=> Indique le temps, en secondes, après lequel un voisin est considéré comme absent. (par défaut 40).

pollinterval = v => Indique le temps, en seconde, pour les interfaces de type NMBA, au bout duquel les routeurs dit absents sont recontactés.

[neighbor] => Sous section de **[interface]** définissant les voisins. Elle contient les variables :

ip = addr_ip => Adresse Ip du voisin

priority = v => Utile dans le cas d'un lan pour le choix du routeur directeur (centralise les informations). La valeur la plus forte gagne.

- **[security]** => Sous section de **[interface]** indiquant l'authentification des trames par un mot de passe, sans cryptage. La variable **password** indique un mot de passe qui ne doit pas dépasser 8 caractères ; il doit être encadré par le caractère '"'.

  Exemple : **password="mdpass"**

- **[md5key]** => Sous section de **[interface]** incompatible avec **[security]** indiquant le cryptage des trames suivant la methode MD5. Le mot de passe utilisé est indiqué par la variable **password** et doit contenir une chaîne de moins de 16 caractères encadrée par le caractère '"'.

Un exemple de fichier de configuration est :

```
[global]
router=10.0.0.1
[area]
area=0.0.0.0
  [interface]
  name=sl0
  type=ptp
  cost=2
```

V.3. Configuration du filtrage de trafic

Les routeurs Netline peuvent être utilisés pour filtrer les échanges de données. Il est ainsi possible de rejeter des protocoles, des ports, des adresses. Ce filtrage peut être utilisé pour sécuriser un accès ou interdire des destinations voir l'utilisation de certains protocoles.

V.3.1. Utilisation du fichier access

Le routeurs possèdent un fichier de configuration : acces. Il est utilisé pour autoriser ou restreindre les accès à l'équipement. Cette restriction se fait par commande. Ainsi, il est possible d'autoriser ou interdire l'exécution d'un commande dès lors que l'adresse de l'équipement distant nécessitant cette utilisation est identifiée. La syntaxe est la suivante :

commande **:**

[**!**]adresse_machine

La commande est celle dont on souhaite limiter l'utilisation, ce peut être telnet, telnetd, ftpd, printerd, time, route, ping. Elle doit être suivit du caractère '**:**'.

L'adresse_machine indique l'adresse ou le nom de l'équipement auquel on autorise ou interdit l'accès. Le caractère '**!**' indique l'interdiction.

Dans la cas de limitation sur des commandes clientes telles que ping, sont bloqué les trames émises du routeur vers les destinations indiquées, par la commande concernée. Dans le cas de commande serveur, ce sont les connexions de l'extérieur vers le routeur qui sont bloquées.

Le fichier access permet de restreindre, de façon très simple, l'accès de certains services à certains équipement.

V.3.2. Utilisation d'orion

L'outil **orion** permet d'aller bien plus loin que le fichier access en réalisant un filtrage fin des trames au niveau IP. Ce processus peut être exécuté en console ou lancé automatiquement par inittab. Il est associé au fichier de configuration fip qui contiendra les règles à appliquer. Le fichier fip contient une liste de filtre. Chaque trame sera confrontée à ces filtres, un à un. Le premier correspondant sera exécuté. Un filtre est inscrit entre caractères "**{}**", il contient une liste de critères de sélection et un ordre d'orientation de la trame correspondant aux critères. Cet ordre est indiqué par la commande "**vers=**". Il peut y avoir plusieurs critères, alors ils sont séparé par un espace, l'opérateur entre ces critères est toujours de type **ET**. La syntaxe est donc du type :

{ liste_de_critères **vers=**destination,option [**stat=**valeur]**}**

Les critères possibles sont :

- **du=**{**lan**|**wan**} => La trame concernée provient soit du réseau local (**lan**) soit du réseau distant (**wan**). Le N2211 possédant deux ports lan, si besoin, il sera précisé quel lan est concerné : alors la valeur sera **lan0** ou **lan1**.
- **long=**val => La trame concernée devra être exactement de longueur val. Il est possible de choisir des longueurs minimum et maximum avec les commandes **longm** et **longM**. La longueur s'exprime en octets.
- **@src** {**|** | **&**} mask { **==** | **>** | **<** | **<=** | **>=** } val
  => Permet de sélectionner la trame selon son adresse IP source. Cette adresse est tout d'abord modifiée par un masque, l'opération peut être un ET ou un OU logique. Le résultat de cette opération est comparé à la valeur en utilisant l'opérateur indiqué : égalité, supériorité / infériorité stricte ou non. Un exemple peut être, par exemple pour sélectionner les trames venant du réseau 192.168.1.xx :

  @src & 0xFFFFFF00 == 192.168.1.0

  Il est possible d'utiliser, pour le mask et la val des notations hexadécimale ou séparée par des points (dot notation). Il est aussi possible d'utiliser des alias :

  @iladdr pour l'adresse IP de l'interface Ethernet.

  @ilmask pour le masque de l'interface Ethernet.

  @ilner pour l'adresse réseau de l'interface Ethernet.
- **@dst** => Permet la même opération que **@src** mais concerne

l'adresse de destination de la trame.

prot=val => Permet de sélectionner la trame si le protocole est égal à celui indiqué par val. Il est possible d'indiquer une plage de protocole plutôt qu'une égalité stricte, alors les commandes **protm** et **protM** seront utilisées.

- **psrc**=val => Permet de sélectionner une trame si son port source est égal à val. Les ordres **psrcm** et **psrcM** permettent de définir une plage de ports.
- **pdst**=val => S'utilise comme **psrc** mais concerne le port de destination.

La trame, si elle est sélectionnée par les critères décrits si dessus, est dirigées vers une destination indiquée dans le champ du même nom. Les différentes possibilités sont les suivantes :

- **CB** => Autorise l'envoie de la trame vers un canal RNIS de type B si une route permet de joindre la destination sur ce type de support.
- **CD** => Autorise l'envoie de la trame vers un canal RNIS de type D si une route permet de joindre la destination sur ce type de support.
- **LL** => Idem sur une LS synchrone.
- **lan** => Idem sur le réseau local.
- **RE** => Rejette la trame.
- **auto** => Autorise l'envoie de la trame vers le support adapté à sa destination.

Il est possible d'ajouter aux mots clefs indiqué ci-dessus une option :

- **S** => Demande le rejet silencieux de la trame au cas échéant.
- **L** => Demande l'envoie d'un syslog prévenant l'administrateur qu'une telle trame a été reçue.
- **R** => Demande de ralentir les trames décrites : elle seront moins prioritaires.

Enfin, il est possible d'ajouter un modérateur selon le statut du lien d'acheminement. La commande **stat=** suivit d'une valeur permettra de n'acheminer la trame que dans certaines conditions :

- **C** => La trame ne sera acheminée que si la connexion est déjà ouverte.

- **X** => La trame sera acheminée quel que soit l'état du lien. C'est la valeur par défaut.

Il est important de mettre en fin du fichier fip une règle par défaut s'appliquant à toutes les trames non sélectionnée par nos critère. Cette règle peut interdire toutes les trames non reconnue, auquel cas, elle sera de la forme : **{vers=RE,S}**. Elle peut aussi autoriser toutes les trames non filtrées : **{vers=auto}**.

Orion permet le filtrage sur quelques critères supplémentaires, d'utilisation plus rare, vous pourrez les consulter sur la documentation en ligne Netline.

Orion, associé au fichier fip permet de filtrer le trafic IP entrant et sortant selon des critères de protocole, port ou adresse.

V.3.3. Exemples

Utilisation du fichier access :

- Nous souhaitons que seule l'équipement d'adresse IP 192.168.63.13 puisse accéder au routeur par **telnet** :

  access :

  telnetd:

  192.168.63.13

Utilisation d'**orion** :

- Nous souhaitons que seules les trames provenant du réseau 192.168.2.xx soient autorisées à accéder au lan depuis l'extérieur :

  {du=wan @src & 255.255.255.0 == 192.168.2.0 vers=lan}

  {vers=RE}

- Nous souhaitons interdire les communications de type ftp venant du lan :

  {du=lan psrc=ftp vers=RE}

  {vers=auto}

- Seul l'ordinateur d'adresse IP 192.168.1.250, sur le lan est autorisé à faire un telnet sur l'équipement :

  {du=lan psrc=telnet @src == 192.168.1.250 @dst=iladdr vers=auto}

  {psrc=telnet @dst == iladdr vers=RE}

  {vers=auto}

Comme nous le constatons au travers des exemples, il est possible d'utiliser des alias pour les ports, il est aussi possible de le faire pour les protocoles : tcp, udp, icmp... sont reconnus par le système.

V.4. Translation d'adresses IP

Dans le chapitre II.3.8, nous avons vu dans quel cas utiliser la translation d'adresse, rappelons simplement que lors d'une connexion à un réseau publique, vous obtiendrez une et une seule adresse IP publique. Il faut utiliser cette adresse unique pour l'ensemble des échanges qui vont avoir lieu sur le réseau privé ayant lui un plus grand nombre d'adresse privé. La translation repose sur l'attribution de ports différents associés aux adresses.

La translation IP est activée simplement en ajoutant dans numip l'option **t** sur la ligne identifiant la connexion à translater. Dans le cas d'une interface Ethernet, une option de la commande **ifconfig** assurera le même rôle. Un fichier de configuration : trans.conf permet de configurer la translation :

Pour que la translation fonctionne, il lui est nécessaire de connaître l'adresse publique à utiliser, celle-ci peut être connu ou attribuée à la connexion. La variable **externe** adresse_ip est **obligatoire**. L'adresse_ip sera remplacée par **auto** si l'adresse est obtenue par PPP à la connexion.

D'autre paramètres sont optionnels, ils permettent d'autoriser ou non les accès à l'équipement coté lan et coté wan :

- **transicmp** val => Si val est 1, l'équipement répondra aux requêtes ICMP sur sont adresse publique s'ils proviennent du réseau publique. S'il est à 0 non.
- **lanicmp** val => Idem pour les accès à l'adresse privée à partir du réseau publique.
- **transudp** val => Autorise les accès UDP sur l'adresse publique coté réseau publique.

- **lanudp** val => Idem pour l'adresse privée à partir du réseau publique.
- **transtcp** val => Autorise les accès TCP sur l'adresse publique coté réseau publique.
- **lantcp** val => Idem pour l'adresse privée à partir du réseau publique.

Il est possible d'ajouter des options particulières, d'autres non listées ici existent :

- **mcast** => Autorise la réception des trames multicast à partir du wan.
- **vdo** => Permet le traitement sur translation des connexion VDOLive.
- **sustain** sec => Indique, par sec exprimé en seconde, combien de temps une translation non utilisée reste valide. Ensuite, elle sera détruite.

  Une variante permet de demander une validité plus longue pour certains protocoles, alors la syntaxe est : **sustain** sec proto**/**port
- **flush** => Détruit immédiatement les entrées lors de la fermeture du canal support.
- **autoflush** => Idem sur les canaux à attribution d'adresse dynamique uniquement.

Enfin, la translation d'adresse permet la redirection de ports, il s'agit de renvoyer de façon statique les trames arrivant sur un port donné sur l'adresse publique vers un équipement identifié par son adresse IP sur le réseau local :

- **tcport** adrIP**/** port => Renvoie sur l'adresse IP adrIP les données émises en TCP vers le port port. Il est possible d'indiquer une plage de ports par port_min**:**port_max. Il est aussi possible de rediriger les trames vers un port différent sur l'équipement du réseau local. La syntaxe est la suivante : **tcport** adrIP/portLan **@0/**portWan.
- **udport** adrIP/port => Renvoie sur l'adresse IP adrIP les données émises en UDP vers le port udp. Il est possible d'indiquer une plage de ports de la même façon que précédemment.

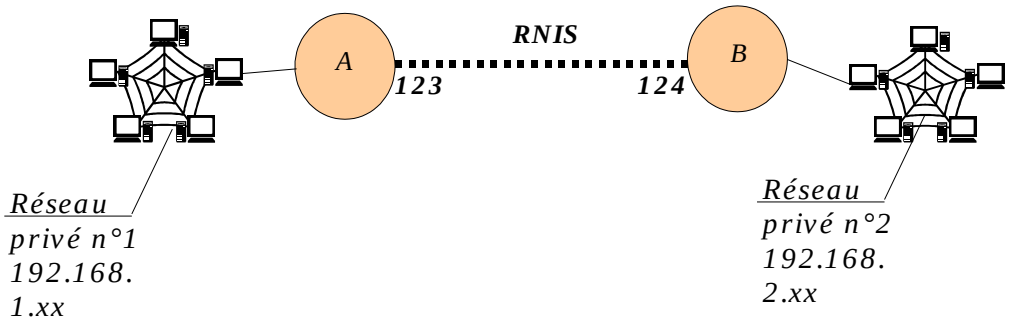
Pour les ports inférieurs à 1024, le nom du port devra être utilisé. Au-delà, le numéro sera accepté. D'autres possibilité sont offerte lors de la redirection de ports, vous les trouverez dans la documentation Netline.

Le protocole gre peut être lui aussi redirigé sur une adresse spécifique par : **greloc** adresse_ip. Plus généralement, les trames d'un protocole donné peuvent être redirigées par : **genrout** adresse_ip/protocole.

La translation IP est activée dans numip et par ifconfig, sa configuration se fait dans le fichier trans.conf. Il est possible de rediriger certains ports sur un équipement du réseau local. Il est possible de filtrer une partie des protocoles à destination de l'adresse publique.

V.5. Exemple de configuration du service de rétro-appel

Nous avons plusieurs fois évoqué l'utilisation du service de rétro-appel, nous allons revenir sur la configuration à utiliser pour connecter deux équipements de cette façon :



L'équipement **B** est le serveur de rétro-appel, son numéro de téléphone est **124**. L'équipement **A** est l'utilisateur du call-back, son numéro est **123**.

La configuration des fichiers numip sera donc la suivante :

Équipement A	Équipement B
numip : 192.168.2.0 = 124:60:CB	numip : 192.168.1.0 = 123:60:c

Lors de l'appel, l'équipement A devra s'identifier suivant la méthode CHAP, le login utilisé pour l'équipement **A** sera reseauA, celui de **B** réseauB. La configuration PPP sera donc la suivante :

Équipement A	Équipement B
ppp.conf : +chap myname reseauA secret : reseauB lemotdepasse	ppp.conf : +chap myname reseauB callwait 200 secret : reseauA lemotdepasse

Le serveur de rétro-appel doit faire le lien entre le nom de login et le réseau à joindre, alors une entrée dans le fichier hosts établit ce lien :

Équipement A	Équipement B
	hosts 192.168.1.0 reseauA

V.6. Utilisation de RNIS en secours de LS

Il arrive malheureusement que les ligne louées soient défaillantes de temps à autre, l'erreur humaine ou la panne technique peuvent à tout instant couper la communication. Les réparations sont généralement rapide, toutefois, la perte d'une LS durant quelques heures peut bloquer la production de l'entreprise. Il est alors possible d'utiliser un accès RNIS comme liaison de secours.

Pour cela, il est possible d'utiliser le routage RIP que nous avons précédemment étudié ; il est aussi possible d'utiliser un système de test de la liaison : lite.

V.6.1. Utilisation du routage RIP

RIP peut logiquement être utilisé pour détecter la coupure d'une ligne et gérer le re-routage des trames. Le seul inconvénient de cette méthode vient du temps de réponse, fixé par le protocole à quelques 3 minutes.

Pour sa mise en oeuvre, l'utilisation d'une ligne série devant être annoncée par RIP nous oblige à positionner la variable FR0 dans admin à l'adresse du routeur distant. Nous avions vu ce point en remarque dans le chapitre consacré à RIP.

admin :

FR0=Adresse_routeur_distant

Il est ensuite nécessaire de donner une route permettant de joindre cette destination, sans quoi le trafic RIP utiliserait la liaison RNIS indiquée en secours. Numip contiendra donc ces deux entrées : la route pour joindre le distant par la LS et la destination par le lien RNIS en secours.

numip :

Adresse_routeur_distant **= C0:0**

Adresse_reseau_distant **=** num_tel**:30**

Enfin, le démon RIP doit être exécuté au démarrage :

inittab :

radp

V.6.2. Utilisation de lite

Lite est un outil qui va scruter automatiquement l'état de la liaison série : il vérifie à intervalles réguliers et paramétrables si des trames ont été reçues, si ce n'est pas le cas, il envoie automatiquement une requête ICMP-ECHO au distant. Celle-ci devrait alors entraîner une réponse, et le nombre de trames reçues devrait augmenter. Si au bout de trois essais, aucune nouvelle trame n'est arrivée, la liaison peut être considérée comme hors service.

Le fichier numip contiendra une entrée pour le réseau distant avec les deux moyens possibles et une entrée permettant de joindre le routeur distant, utilisée pour vérifier l'état de la ligne :

numip :

routeur_distant **= C0:0**

reseau_distant **= C0:0,** num_tel**:30**

Le démon lite sera lancé automatiquement au démarrage : la ligne suivante sera insérée dans le fichier inittab : **lite -s** routeur_distant **&.**

V.6.3. Remarque

Les méthodes évoquées précédemment peuvent s'utiliser dans le cas de LS, mais aussi dans le cas de tunnels par exemple. C'est alors une solution fiable pour une interconnexion de site au travers de l'ADSL, sécurisé par une liaison RNIS.

V.7. Configuration de Radius sur les équipements Netline

Radius est un service permettant la gestion par un serveur de l'authentification. Les routeurs Netline intègrent un client Radius. Nous ne verrons pas ici comment configurer le coté serveur, notons simplement que ce type d'authentification peut être utilisé autant pour l'ouverture d'une console sur le routeur que pour l'établissement d'une communication du wan vers le lan par exemple. Radius se configure au travers d'un fichier de configuration : radius.conf.

Ce fichier comporte deux types de sections, pour les deux types de services rendus par Radius : une première section sert à l'enregistrement, c'est à dire à l'établissement de communications entre le Wan et le Lan, elle est indiqué dans radius.conf par **[accounting]**. Un second type de section sert à l'authentification, il est noté **[authentification]**.

Dans chaque section il est indispensable de trouver la déclaration d'un serveur Radius. La forme est la suivante : nom_du_serveur clef**.** Le nom_du_serveur indique comment joindre le serveur, ce peut être un nom correspondant à une entrée dans le fichier hosts, une adresse IP... La clef est une sorte de mot de passe partagé entre le client et le serveur Radius. Il est possible de proposer plusieurs couple serveur/clef différents, alors ils seront mis ligne après ligne et sélectionner de haut en bas lors de l'établissement de connexion. Le second ne sera pris que si le premier ne répond pas et ainsi de suite.

Les sections de type **[accounting]** acceptent certaines options :

- **.chap** => Indique que l'enregistrement de la section ne se fera qu'après l'identification chap.
- **.pap** => Indique que l'enregistrement de la section ne se fera qu'après l'identification pap.
- **.clid** => Indique que l'enregistrement de la section se fera dès la réception de l'appel.
- **.port** n => Permet le changement de port UDP à utiliser pour la communication avec le serveur.

Les sections de type **[authentification]** acceptent certaines options :

- **.chap** => Indique que si la paire nom_distant/mot-de-passe n'est pas trouvée dans le fichier secret alors, il faut demander au serveur Radius. Attention, certaine fois, le challenge chap doit être limité à 16 pour que cela fonctionne.
- **.pap** => Indique que si la paire n'est pas trouvée dans le fichier passwd, il faut faire appel au serveur Radius.

– **.clid** => Indique que si le numéro du distant n'est pas trouvé dans le fichier numip, il faut faire appel au serveur Radius. Le mot de passe Radius sera alors obligatoirement "Netline-CLID"

– **.port** n => Permet le changement de port UDP à utiliser pour la communication avec le serveur.

Des exemples de configuration sont disponibles sur le site de Netline.

Note : Dans le cas d'une authentification par Radius pour un accès à la console, la variable **PASSWD** aura pour valeur ***radius**. Cette variable est initialisée dans le fichier admin.

V.8. Sauvegarde et restauration des configurations

L'accès aux fichiers de configuration présents sur le routeur se fait simplement au travers du service FTP. Celui-ci s'exécute simplement sur le routeur en lançant le démon qui lui est associé par la commande suivante :

ftpd &

Il est alors possible de transférer des fichiers entre le routeur et un poste utilisant un client FTP. La connexion ne nécessite pas de nom d'utilisateur particulier : tout choix est valable. Le mot de passe est lui le même que celui attribué à l'accès en console.

Le service FTP sera stoppé en exécutant la commande **kill** avec comme paramètre le numéro de processus associé au service FTP. Ce dernier sera obtenu avec la commande **ps**.

Voyons l'exemple ci-dessous de lancement et arrêt du processus ftpd:

```
192.168.1.254>ftpd &
[8]
192.168.1.254>ps
idp dev     nom       etat  prio   position pile          taille pile      sem      message  idu
--- --- ------------ ----- ---- -------------    -----------  --- -------- ---
  0   1   prnull   pret     0 3FF400-3FFFFC    166/ 3072          -        -        0/0
  1   1   named            atten   20 3F7400-3F83FC   1116/ 4096        6        -        0/0
  7   1   kal2     cdga    19 3F4400-3F4FFC    236/ 3072          -        -        0/0
  8   1            ftpd    atten   20 3FC800-3FD3FC    652/ 3072        8        -        0/0
  9   1            I-Dpc   recep   30 3EA000-3EABFC    160/ 3072        -        -        0/0
 10   1            psam    atten   40 3EAC00-3EB7FC    284/ 3072        11       -        0/0
 11   1            plp     atten   40 3EB800-3EC3FC    280/ 3072        13       -        0/0
 12   1            isdn    atten   60 3EC400-3ECFFC    292/ 3072        18       -        0/0
 13   1            ccm     atten   30 3ED000-3EDBFC    276/ 3072        22       -        0/0
 14   1            plm     atten   30 3EDC00-3EE7FC    280/ 3072        25       -        0/0
 15   1            Dpc     recep   30 3EE800-3EF3FC    164/ 3072        -        -        0/0
192.168.1.254>kill 8
```

Il est à noter que le numéro de processus est retourné lors du lancement du démon. Dans la table retournée par **ps** il est indiqué dans la colonne idp.

Il est aussi possible de fonctionner en sens inverse : alors si vous avez un serveur ftp à disposition, vous pouvez vous y connecter de l'équipement au travers de l'utilisation de la commande **ftp**. Cette solution peut éviter d'ouvrir des services tels que FTP sur les routeurs eux-mêmes.

La transmission des fichiers se fait à l'aide des commandes FTP classiques : get et put.
Exemple de récupération et d'envoi du fichier numip :

Lancement du service sur le routeur :

```
                N2211 (N2211>)
Real Time Monitor 'n Shell  Version 1.43   (16:24 24/07/02)   [? pour l'aide]
N2211> ftpd &
[111]
N2211> ftpd: attention, acces sans mot-de-passe!
```

Ouverture de la connexion FTP depuis un poste client :

```
disk@disk:~/projets/Formation> ftp 192.168.63.143
Connected to 192.168.63.143.
220 N2211> serveur FTP (Version 1.43) ready.
Name (192.168.63.143:disk):
331 Password required for disk.
Password:
230 disk login ok.
Remote system type is Netline.
ftp>
```

Récupération du fichier numip :

```
ftp> get numip
local: numip remote: numip
200 Commande okay.
150 Connexion de donnees pour commande.
226 Fin de transfert.
141 bytes received in 0.000518 secs (2.7e+02 Kbytes/sec)
ftp>
```

Envoi d'un fichier numip vers le routeur :

```
ftp> put numip
local: numip remote: numip
200 Commande okay.
150 Connexion de donnees pour commande.
226 Fin de transfert.
157 bytes sent in 0.000101 secs (1.5e+03 Kbytes/sec)
ftp>
```

Certains noms de fichiers sont particuliers et interprétés de façon spécifique par les routeurs Netline, c'est le cas du fichier **backup**. Ce nom n'indique pas un fichier particulier, mais l'ensemble des fichiers de configuration du routeur. S'il est utilisé en récupération ou en envoie, alors un fichier s'appelant **backup** sera créé, il contiendra l'ensemble des fichiers de configuration.

Cette méthode permet la sauvegarde et la restauration de la configuration complète d'un équipement pour qu'elle fonctionne correctement, il faut passer le client ftp en mode binaire :

Exemple de sauvegarde de la configuration :

```
disk@disk:~/projets/Formation> ftp 192.168.63.143
Connected to 192.168.63.143.
220 N2211> serveur FTP (Version 1.43) ready.
Name (192.168.63.143:disk):
331 Password required for disk.
Password:
230 disk login ok.
Remote system type is Netline.
ftp> bin
200 Type set to I.
ftp> get backup
local: backup remote: backup
200 Commande okay.
150 Connexion de donnees pour commande.
226 Fin de transfert.
4360 bytes received in 0.00858 secs (5e+02 Kbytes/sec)
ftp>
```

La commande FTP et le service FTPD permettent la sauvegarde et la restauration de la configuration. Il est possible d'accéder aux fichiers un à un, mais il est aussi possible de lancer des procédures de sauvegarde et restauration globales en utilisant l'allias backup.

Note : Les routeurs Netline intègrent une fonction permettant de replacer l'équipement dans sa configuration d'usine. Cette option peut être intéressante si suite à diverse manipulations hasardeuses, vous avez finit par placer l'équipement dans une configuration instable, ou si vous souhaitez effacer toutes trace de votre ancienne configuration. Il faut pour cela exécuter la commande **usine** sur la console. L'équipement vous demandera, par mesure de sécurité, son numéro de série, indiqué au dos de l'appareil.

Note : Il est possible d'utiliser un mot de passe différent et un login particulier pour l'accès FTP. Les variables **FTP_PWD** et **FTP_USER** peuvent pour cela être initialisées dans le fichier admin.

V.9. Mise à jour du logiciel Netline

Les routeurs sont des éléments qui nécessitent une mise à jour régulière, celle-ci permet la correction de problèmes logiciel, apporte un meilleure interopérabilité entre les équipements de différentes marques. Enfin la mise à jour permet l'ajout à un routeur ancien de nouvelles fonctionnalités apparues récemment.

La mise à jour des routeurs Netline se passe de la même façon que la restauration de fichiers : Il faut simplement envoyer au routeur un fichier dont le nom sera **update**. Le routeur sait alors qu'il s'agit d'une mise à jour, qu'il effectuera de façon automatique. Il est nécessaire que le fichier soit transmis en mode binaire (commande FTP **bin**). La mise à jour débutera une fois la session ftp fermée. Les fichier **update** peuvent être téléchargés à partir du site de Netline.

Il est aussi possible de mettre à jour les routeurs directement à partir du site de Netline, dès lors que l'équipement possède un accès à Internet et à un DNS permettant de résoudre l'adresse www.netline.fr. Alors une commande spécifique est employée :

hload -h www.netline.fr /sload/N2211 update

VI. Administration des routeurs Netline

Pour l'administration des routeurs Netline, nous avons généralement utilisé la console, accessible soit au travers de la liaison série, soit au travers de la commande telnet. Nous ne reviendrons pas trop dessus, si ce n'est pour voir comment l'utiliser dans des scripts à l'aide de la commande **rsh**. Nous avons aussi évoqué l'utilisation du service **httpd** pour la configuration de l'équipement, nous n'avons pas beaucoup plus de chose à ajouter puisque ce service offre un accès plus simple aux fichiers de configuration, vous devriez pouvoir vous en servir sans plus de détails.

Nous verrons donc dans ce chapitre des moyens plus évolué d'administrer les routeurs, et surtout des moyens permettant de les surveiller, une fois configuré, jour après jour. Nous examinerons donc entre autre l'utilisation de **syslog** et de **SNMP** et celle de commandes d'administration utilisées en consoles.

VI.1. Administration, surveillance de l'équipement en console

La console intègre des outils permettant de diagnostiquer l'état de l'équipement. Nous avons vu les commandes **route**, **ifconfig** qui nous apportent des informations sur le routage et la configuration des interfaces de l'équipement. Il existe d'autres commandes, très utiles pour connaître son état :

VI.1.1. La commande numstat

La commande **numstat** affiche les informations concernant la gestion des connexions vers les réseaux distants. Une option placée à la suite de la commande indique quelle information est souhaitée :

- **numstat -a** => Affiche l'état des différents liens.
- **numstat -C** => Affiche les statistiques de compression.
- **numstat -j** => Affiche le journal des connexions.
  La variable **JOURL** placée dans admin permet de définir le nombre de ligne du journal.
- **numstat -t** => Affiche les informations de translation d'adresse IP.

Il est possible d'ajouter à la suite une seconde option :

- **c** => Demande l'affichage en continu, rafraîchi chaque seconde.
- **z** => Demande la remise à zéro des informations indiquées dans l'option demandée.
- **n** => Demande l'affichage de la durée des communication et non l'heure de fin.

VI.1.2. La commande netstat

La commande **netstat** permet d'interroger les statistiques réseau. Ainsi, en fonction des options passées à la commande, différentes informations vont être transmises à l'utilisateur :

- **netstat -I** => Affiche la bande passante utilisée sur les interfaces sous forme semi-graphique.
- **netstat -b** => Affiche la bande passante utilisée par interface, sous forme de compteurs.
- **netstat -a** => Affiche l'état des sockets IP.
- **netstat -d** => Affiche les informations relatives à l'agrégation de canaux.
- **netstat -p** => Affiche les paramètres PPP en cours d'utilisation.
- **netstat -r** => Affiche la table de routage.
- **netstat -s** => Affiche les statistiques par protocole.

Tout comme pour **numstat**, d'autres options comme '**c**' permettent un rafraîchissement automatique des résultat toutes les secondes, '**z**' permet la remise à zéro des compteurs affichés.

VI.1.3. Exécution de commandes à distance

Lorsque l'on souhaite automatiser certains traitements d'administration par exemple, il est intéressant de pouvoir exécuter des commandes à distance, celles-ci peuvent alors être lancés depuis un équipement informatique distant, par un script. Ceci est possible grâce au service **rshd** qui peut être lancé sur les routeurs Netline. **rshd** se lance soit en console, soit automatiquement en plaçant la commande dans la fichier inittab. L'options **-l** permet de suivre via un syslog les commandes exécuté. Cette option permet une surveillance de ce démon, dont un mauvais usage peut être dangereux pour la sécurité de votre réseau.

Le fichier rcmds limite l'utilisation qu'il est possible de faire de ce service. Il permet de limiter l'utilisation de chaque commande à un groupe restreint d'utilisateurs. Son indiqué dans ce fichier, par équipement et par utilisateur les commandes autorisées à l'exécution. La syntaxe est la suivante :

{*|adresse|nom}**:**{*|utilisateur}**:**{*|commande1[,commande2...]}

La première partie indique quels sont les équipements ayant l'autorisation pour cette ligne. Cette autorisation peut concerner tout le monde, auquel cas, '*****' est utilisé, ou un seul équipement, auquel cas est indiqué son adresse IP ou son nom s'il possède une entrée dans le fichier hosts.

La seconde partir identifie le ou les utilisateurs autorisés. '*' Indique une autorisation globale, sinon le nom de l'utilisateur concerné sera ici indiqué.

La dernière partie liste les commandes qu'il est possible d'exécuter à distance, '*' indiquera que toutes les commandes sont exécutables.

Si l'on souhaite autoriser l'exécution de **numstat** à l'utilisateur disk depuis l'équipement administration, d'adresse IP 192.168.63.13, la configuration sera la suivante :

hosts :

192.168.63.13 administration

rcmds :

administration:disk:numstat

Il faut ensuite exécuter le démon rshd :

rshd &

Alors, il est possible d'exécuter la commande **numstat** depuis l'équipement administration :

```
disk@disk:~/projets/Formation> rsh -l disk 192.168.63.143 "numstat -j"
Jeu 29 Aout 2002 16:22:09
 C:      Adresse:  Numero distant:  Recus:  Trans: Errs: Debut:   Fin:    Lib:
---      --------  ---------------  ------  ------ -
T3   193.56.124.0 >U_62.4.18.229        0       0     1 12:24:06 12:24:40   0
O2       0.0.0.0 >O_0.              14164   13706     0 12:24:32
T3   193.56.124.0 >U_62.4.18.229        0       0     1 12:24:45 12:25:20   0
T3   193.56.124.0 >U_62.4.18.229      464     173     1 12:25:37
disk@disk:~/projets/Formation>
```

VI.1.4. Configuration du journal des connexions

Le journal des connexions, consultable par **numstat -j** peut aussi être publié de façon automatique lorsqu'une communication de termine. La variable **JOURD** permet d'initialiser cette fonctionnalité. Cette variable sera définie dans le fichier admin. Il s'agit d'un champ de bits, indiquant vers quelle destination ces log seront envoyés :

- 1 et 2 indiquent la console ; 2 spécifie qu'une session devra être ouverte.
- 4 indique une emission syslog

Ainsi, si l'on souhaite un affichage en console et syslog, il faudra choisir comme valeur 1+4 = **5.**

D'autres bits sont utilisés pour ajouter des options, ainsi, l'ajout de 8 permet la sortie de la durée plutôt que la date de fin....

La console est un bon outil d'administration, elle permet aussi la surveillance de l'état de l'équipement, au travers de commandes comme netstat et numstat. Il est possible d'y accéder simplement au travers du réseau par l'utilisation du service rshd dont l'utilisation nécessite une configuration au travers du fichier rcmds.

VI.2. Utilisation de syslog

Syslog est un système permettant de remonter des alertes vers un système distant. Il permet la surveillance d'un équipement et la mémorisation sur le long terme d'événements passés. En effet, un routeur ne mémorise qu'un nombre limité d'événements et les perds de façon certaine à chaque redémarrage.

L'utilisation de syslog est activée sur les routeurs au travers de la commande **syslogd** qui sera exécuté en console ou automatiquement au démarrage en étant placé dans le fichier inittab. La commande doit être suivit du nom de l'équipement devant mémoriser les événements syslog. Il faudra donc veiller à ce qu'une entrée correspondante existe dans le fichier hosts. Voici un exemple de configuration pour syslog envoyés vers une machine d'administration :

hosts :

192.168.63.13 administration

inittab :

syslogd -f administration **&**

Il est possible de tester le bon fonctionnement de ce service à l'aide de la commande **syslog -t essai** par exemple, qui enverra la chaîne "essai" à l'équipement distant.

La configuration du serveur syslog recevant les messages sur l'équipement d'administration devra être effective pour que les messages soient enregistrés convenablement.

Syslog est un outil permettant de sauver les événements de façon durable sur un équipement distant. Il permet une surveillance efficace du routeur. Son utilisation passe par l'exécution d'un processus syslogd sur le routeur et la configuration d'un serveur syslog sur l'équipement d'administration.

VI.3. Utilisation de SNMP

SNMP est un protocole standard utilisé pour la surveillance des équipements réseau. Le routeur contient en quelque sorte une base de donnée de sa configuration. La structure de cette base est décrite dans un fichier appelé MIB. Il est possible de télécharger la MIB des routeurs sur le site de Netline. Un client, installé sur un poste d'administration, va, au travers de SNMP interroger le routeur sur son état ou sa configuration.

L'activation de SNMP sur le routeur se fait par l'exécution du démon **snmpd**. Son lancement se fait en console ou au démarrage de l'équipement en plaçant la commande dans le fichier inittab.

La syntaxe est la suivante :

snmpd [**-p** port] [**-c** communauté] **&**

Le port est un argument facultatif indiquant le port UDP qui sera utilisé pour recevoir les requêtes SNMP.

La communauté est une chaîne de caractères servant en quelque sorte de clef. Elle doit être connue du serveur et du client. Par défaut, la communauté est : public.

Attention, **snmpd** doit être exécuté après le lancement d'**opéra**.

L'utilisation de SNMP amène à l'émission par le routeur d'alarme SNMP appelées SnmpTrap. La variable **SNMPTRAP** qui peut être définie dans admin permet d'indiquer à destination de quelle adresse ces alarmes doivent être émise. Il est possible d'interdire l'émission de ces alarmes en initialisant convenablement la variable **SNMPTNOUT** dans le fichier admin. Nous vous recommandons de vous référer à la documentation en ligne pour plus de détails.

Les variables **SYSADM** et **SYSLOC,** initialisées dans le fichier admin permettent de donner le nom de l'administrateur et la localisation de l'équipement. Ces informations peuvent être, par la suite, demandées à l'aide d'une requête SNMP.

SNMP est un service permettant l'interrogation des équipements réseaux à distance, il est activé sur les routeurs Netline par la commande snmpd. SNMP génère des alarmes qui seront émises vers un serveur dont l'adresse est indiqué par la variable SNMPTRAP. La variable SNMPTNOUT permet de configurer la liste des alarmes à émettre.

VII. Autres sources d'informations

L'administration d'un routeur est proche de l'administration d'un système informatique. Elle nécessite donc la maîtrise de nombreux concepts réseaux et télécom. Nous avons essayé dans ce documents de regrouper de façon synthétique les connaissances minimum permettant une administration confortable des routeurs Netline. Toutefois, il pourra être nécessaire de compléter ces informations par d'autres :

- L'ensemble des RFC, documents décrivant les protocoles, est facilement accessible sur Internet à l'aide de mots clefs tels que RFC et le nom du protocoles.

- La documentation Netline, accessible par www.netline.fr, présente dans le dossier "Le Manuel" donne une description exacte de chaque commande, de chaque variable. Vous trouverez aussi à cette adresse des exemple de configuration et une foire aux questions.

- Il est aussi possible d'envoyer un e-mail au support Netline dont l'adresse est support@netline.fr en précisant, un numéro de téléphone ou vous joindre, le type du matériel concerné (N2211 par exemple), le numéro de série, son numéro RNIS (s'il est joignable de cette façon) et bien sure un descriptif des problèmes rencontrés.