Sujet :

Contrôleur de trafic

Elaboré par :
CHOKRI HARBAOUI

Rapport de Projet de Fin d'Etudes

Présenté en vue d'obtention du titre

Licence Appliquée en Sciences et Techniques de l'information et de Communications

Encadré par :
Mr. Hattab Ayari (Encadreur)
Mr. Kamel Khediri (Rapporteur)
Société d'accueil : ISCI

UNIVERSITE VIRTUELLE DE TUNIS

Année Universitaire 2010 / 2011

بِسْمِ اللَّهِ الرَّحْمَٰنِ الرَّحِيمِ

وَأَنزَلَ اللَّهُ عَلَيْكَ الْكِتَابَ وَالْحِكْمَةَ وَعَلَّمَكَ
مَا لَمْ تَكُن تَعْلَمُ ۚ وَكَانَ فَضْلُ اللَّهِ عَلَيْكَ عَظِيمًا

Dédicaces

Tout qui commence bien finis bien, après un an de travail acharné dans le cadre de ma formation, il est bien le temps de savourer mon succès.

Je dédie mon projet de fin d'études à :

- La mémoire de mon oncle Salah que je n'oublierais jamais.
- Mes chères parents Mohamed Akremi et Naima, mon frère Nabil, mes deux sœurs Marwa et Hanane, mon oncle Ahmed, mon oncle Mohsen et toutes les membres de notre précieuse famille.
- Mes amis : Ramzi, Aymen, Lassaad, Noureddine, Abdersattar, Manssour et toutes les personnes qui me connaissent.
- Mon encadreur Mr Hattab Ayari pour son grand soutient.
- Mes professeurs qui ont veillé sur ce que je suis maintenant, surtout Mr Kamel KHEDHIRI mon professeur de Réseau mobile.

Harbaoui Chokri

Remerciements

C'est avec la plus chaleureuse sensation que je profite de cette occasion pour remercier d'abord tous mes professeurs d'UVT de Tunis qui m'ont supporté tout au long de ma formation et qui m'ont permis d'avoir toutes ces multiples connaissances lors de ma formation ce qui m'ont donné les ailes d'aborder mon projet de fin d'étude avec confiance et sincérité.

Je tiens toute ma gratitude à Monsieur le directeur de l'ISCI Mr Mohamed Chakroun qui ma accepté comme stagiaire dans le cadre de mon projet de fin d'étude et qui ma consacré ses équipements et la disponibilité de ses employés.

Je tiens à remercier également Monsieur Mohamed Thouraya Secrétaire Général de l'ISCI de Tunis qui ma bien encadré, dirigé et aidé pour réaliser mon projet pendant la durée de mon stage et à qui j'adresse ma vifs remerciements.

J'exprime ma gratitude et mon sentiment de respect profond envers Madame Marwa Dhriwa pour son aide, et ses conseils.

Enfin, je présente tout mes respect et pour ceux tous qui m'ont aidé de réaliser mon projet de fin d'étude et de réaliser ce rapport dans les bonnes conditions.

Table des matières

Table des figures

Introduction générale

Ce rapport a été rédigé suite au stage de fin d'étude effectué au sein de l'Institut Supérieur de la Civilisation Islamique (ISCI) dans le cadre de mon formation de **Licence Appliquée en Sciences et Techniques de l'information et de Communications.**

Durant les trois mois de ce stage, j'ai pu réaliser mon projet dans de bonnes conditions grâce au soutien apporté tant au niveau professionnel qu'universitaire.

A cause de mon occupation dans l'ISCI tant que technicien Supérieur responsable sur maintenance du réseau et mise à jour de site web, j'ai réservé le premier chapitre de ce rapport pour la présentation de l'ISCI.

Pour s'intégrer dans ce travail, je dois avoir un minimum de connaissance sur l'informatique en général, sur le réseau en particulier et sur son environnement, c'est pour cette raison que le deuxième chapitre rassemble quelques informations sur l'architecture des réseaux informatiques.

Pour savoir plus sur la catégorie du produit développé, j'ai consacré le troisième chapitre pour expliquer et présenter quelques produits testés et étudié juste avant la réalisation de mon projet.

Le quatrième chapitre n'est qu'une présentation de la problématique concernant la technique de communication entre les ordinateurs de même réseau ainsi que le choix d'outil de développement utilisé.

Le fruit de mon travail ou l'application réalisé est le thème du siscième et dernier chapitre de ce rapport et tous les détails techniques sur la réalisation du projet se trouvent au niveau de ce chapitre. Mais avant de le réaliser j'ai préparé la conception pour le bon déroulement des besoins annoncés par la spécification et cela dans le cinquième chapitre.

Présentation du projet
(CAHIER DES CHARGES)

Introduction :

Les réseaux à l'heure actuelle sont des composantes très importantes de l'informatique. Chaque interconnexion permet, certes, de partager réciproquement diverses informations, mais offre aussi l'ouverture de son réseau à d'autres utilisateurs. Cet accès doit être limité et sécurisé afin de ne partager que les données ce que l'on autorise.

Aujourd'hui, la tendance dans l'évolution de cette technologie, est de rendre l'information disponible, n'importe où et n'importe comment. De plus les réseaux deviennent domestiques, il n'est plus rare de rencontrer plusieurs ordinateurs au sein d'un même foyer.

Il se trouve que le monde utilise directement les réseaux informatiques, et possède une partie de sa vie sous forme de bits. On ne peut pas donc plus se permettre d'ignorer les risques de sécurité qui encourent ces donnés. Et pour réaliser la sécurité des informations on utilise plusieurs technologies de sécurité comme le firewall, et la cryptographie. Et il se trouve justement qu'une petite partie de la sécurité informatique passe par le contrôleur de trafic.

Un contrôleur de trafic est un dispositif permettant d'écouter le trafic du réseau, c à d capturer les informations qui y circulent.
En effet dans un réseau non commuté, les données sont envoyées à toutes les machines du réseau. Toutefois, dans une utilisation normale les machines ignorent les paquets qui ne leur sont pas destinés. Ainsi en utilisant l'interface réseau dans un mode spécifique (mode promiscues), il est possible d'écouter tout le trafic passant par un adaptateur réseau (une carte réseaux Ethernet, une carte réseaux sans fil,…).

Le contrôleur de trafic est le formidable outil permettant d'étudier le trafic du réseau. Il sert également aux administrateurs pour diagnostiquer les problèmes sur leur réseau ainsi que pour connaître la nature des données qui y circule.

L'objet de ce projet de fin d'étude consiste à mettre en place une application de sécurité au sein de l'ISCI, qui se base sur un contrôleur de trafic.

Sujet du projet :

Mon projet de fin d'étude a comme titre « contrôleur de trafic », le nom que nous avons attribués pour ce contrôleur est « **HC_ScanNetwork** » (les lettres H et C désignent Harbaoui et Chokri) et comme son nom indique c'est un petit sniffer. Ce projet est développé et testé au sein de l'ISCI.

Mon contrôleur de trafic permet à un administrateur réseau de recevoir tout le trafic circulant en capturant tous les paquets, les analyser afin d'avoir une vision globale du trafic et détecter les sources de problèmes qui peuvent exister.

Objectifs du projet :

Le but du projet est de développer un outil d'administration d'un réseau (contrôleur de trafic). Plusieurs fonctionnalités sont requises dans le sujet de ce travail à savoir la capture des paquets, la mise en œuvre des statistiques (graphique) des paquets selon le protocole, le teste des adresses IP (Ping), la recherche des hôtes, la génération des rapports et d'autres informations supplémentaires.

La plate-forme :

Mon projet est développé sur deux machines (ordinateurs) équipées de système d'exploitation Windows XP, les deux postes sont liés par un câble croisé formant un petit réseau, ça m'aide de tester le fonctionnement de mon application.

Pour développer un contrôleur de trafic, je dois faire appel à un outil qui m'aide au niveau de la programmation réseau, mon choix est reposé enfin sur le Microsoft Visual Basic 6.0, plus d'informations sur ce langage de programmation, ses avantages et d'autres détailles se trouvent dans le chapitre 4 de ce rapport.

Il faut également mentionner que l'utilisation d'UML permet d'analyser des solutions techniques qui se trouvent dans le chapitre 5 de ce rapport.

CHAPITRE 1

Présentation de l'ISCI

« Nous ne somme pas les seuls mais nous sommes les meilleurs »

1.Présentation générale :

L'institut supérieur de la civilisation islamique **(ISCI)** de Tunis est considéré comme l'un des établissements les plus renommées mondialement dans la recherche scientifique spécialisée dans les études islamiques, il reçoit chaque année un ensemble d'étudiants étrangers provenant du monde musulman à qui il offre en plus du programme d'enseignement, un ensemble d'activités culturelles, sportives et informatique complétant leur formation académique, créant ainsi un équilibre entre les études et la culture.

2. L'activité de l'ISCI :

Les activités sont regroupées organiquement au sein de ***deux salles spécialisées.***

1. Salle informatique :

- PC: Acer Veriton 7700
- Imprimante: HP Laser jet 1000Séries.
- Logiciels et outils bureautiques: Microsoft, Linux.

2. Salle Réseaux LAN & WAN:

- Equipments LAN (Hub Switch): **Switch DLinkDES - 3225G; Switch DLink DSG 3426.**
- Gestionnaire de réseaux: **MICROSOFT NT SERVER**, **LINUX**.
- Onduleurs: Pulsar3000VA.
- Serveur: IBMx35000
- Routeurs : **CISCO**. (Pour l'application ADAB)
- Modems analogiques réseaux: **CXR**.
- Multiplexeurs : **CXR**, **SAGEM**.
- Modems Internet: **PLANET**.
- Commutateurs **X25, FRAD** et **Switch ATM: CXR** et **CISCO**.
- Modems HDSL et accessoires de communication: **CXR**.
- Anti-virus: **Office scan, Avast 4.8.**

CHAPITRE 2

Réseaux Informatiques

« Toute l'imagination et tout le savoir faire de l'analyste seront sans emploie s'ils ne peuvent s'exercer sur une base réaliste »

A. Collongue

I. Introduction générale :

Depuis sa création il y a presque quinze ans, la micro-informatique n'a cessé de s'améliorer, de se démocratiser. Elle n'est plus réservée à une élite de savants. Mais pour atteindre cette maturité, elle a du s'installer dans les milieux professionnels. Dans l'industrie, la micro-informatique a et aura de plus en plus d'importance. L'informatique dans une entreprise permet d'archiver, de créer des produits, de tester aussi, mais elle permet surtout de communiquer. Pour cela il faut avoir ce que l'on appelle, un réseau informatique. Ce réseau doit être fiable, rapide et surtout facile à gérer. Le type de réseau peut-être déterminant pour la bonne marche de celui-ci et de l'entreprise. Il devra être adapté aux besoins de l'entreprise.

Dans un sondage réalisé il y a quelques années déjà aux Etats-Unis, il était dit que si une entreprise était brutalement privée de son informatique, suite à un incendie ou un tremblement de terre par exemple, il y avait 95% de risque que cette entreprise fasse faillite. C'est pourquoi aujourd'hui, la bonne marche d'une entreprise dépend avant tout de son informatique, et son implémentation est une étape relativement difficile et minutieuse.

II. Réseau informatique :

1. Que signifie réseau ?

Un réseau en général est le résultat de la connexion de plusieurs machines entre elles, afin que les utilisateurs et les applications qui fonctionnent sur ces dernières puissent échanger des informations. Le terme réseau en fonction de son contexte peut désigner plusieurs choses. Il peut désigner l'ensemble des machines ou l'infrastructure informatique d'une organisation avec les protocoles qui sont utilisés, ce qui 'est le cas lorsque l'on parle de Internet.

Le terme réseau peut également être utilisé pour décrire la façon dont les machines d'un site sont interconnectées. C'est le cas lorsque l'on dit que les machines d'un site (sur un réseau local) sont sur un réseau Ethernet, Token Ring (voir architecture Ethernet et token Ring de la page 27 et 28), réseau en étoile, réseau en bus (voir topologie de la page 28). Le terme réseau peut également être utilisé pour spécifier le protocole qui est utilisé pour que les machines communiquent.

2. Pourquoi des réseaux ?

Les réseaux sont nés d'un besoin d'échanger des informations de manière simple et rapide entre des machines. Lorsque l'on travaillait sur une même machine, toutes les informations nécessaires au travail étaient centralisées sur la même machine. Presque tous les utilisateurs et les programmes avaient accès à ces informations.

Pour des raisons de coûts ou de performances, on est venu à multiplier le nombre de machines. Les informations devaient alors être dupliquées sur les différentes machines du même site. Cette duplication était plus ou moins facile et ne permettait pas toujours d'avoir des informations cohérentes sur les machines. Un réseau est donc un dispositif qui permet d'interconnecter différents matériels informatiques dont l'organisation des machines qui l'autour possède les avantages suivants :

- Partage de ressources communes (critère économique) :
- Partage d'équipements : imprimantes, mémoire, …
- Partage de logiciels
- Partage de données

III. Les protocoles :

1. Introduction :

La notion de protocole de communication est assez vague : en générale un protocole est tout ensemble de spécifications précisant les moyens logiciels et physiques (les équipements nécessaires pour partager des informations).

Je différencierais les protocoles de bas niveau (correspondant aux couches basses, comme le protocole Ethernet) des protocoles de haut niveau. Les protocoles de haut niveau permettent de structurer l'acheminement de l'information à travers le réseau (notamment en permettant l'adressage logique), la gestion des erreurs et le contrôle de flux ainsi que l'interaction avec les couches applications.

Pour que des paquets de données puissent se rendre d'un ordinateur source à un ordinateur de destination sur un réseau, il est important que toutes les unités du réseau communiquent dans la même langue ou *protocole*. Un *protocole* consiste en un ensemble de règles qui rehaussent l'efficacité des communications au sein d'un réseau. Voici quelques exemples:

- ✓ Au Parlement provincial et fédéral, une forme de droit de parole permet aux centaines de députés, qui désirent tous parler, de s'exprimer à tour de rôle et de communiquer leurs idées de manière ordonnée.
- ✓ En conduite automobile, il faut indiquer, à l'aide de son clignotant, que l'on désire tourner à gauche, sinon ce serait le chaos sur les routes.
- ✓ Lorsqu'ils pilotent un avion, les pilotes obéissent à des règles très précises pour communiquer d'un appareil à l'autre ou d'un appareil à la tour de contrôle.

2. Pourquoi une normalisation ?

Au cours des deux dernières décennies, le nombre et la taille des réseaux ont augmenté considérablement. Cependant, bon nombre de réseaux ont été mis sur pied à l'aide de plates-formes matérielles et logicielles différentes. Il en a résulté une incompatibilité entre de nombreux réseaux et il est devenu difficile d'établir des communications entre des réseaux fondés sur des spécifications différentes. Pour

résoudre le problème de l'incompatibilité des réseaux et leur incapacité de communiquer entre eux, l'*Organisation internationale de normalisation* (*ISO*) a étudié des structures de réseau telles que DECNET, SNA et TCP/IP afin d'en dégager un ensemble de règles. À la suite de ces recherches, l'ISO a mis au point un modèle de réseau pour aider les fournisseurs à créer des réseaux compatibles avec d'autres réseaux.

Le *modèle de référence OSI* (Open System Interconnexion - interconnexion de systèmes ouverts), diffusé en 1984, a ainsi été créé comme architecture descriptive. Ce modèle a apporté aux fournisseurs un ensemble de normes assurant une compatibilité et une interopérabilité accrues entre les divers types de technologies de réseau produites par de nombreuses entreprises partout dans le monde.

3. Structuration en couches de protocoles :

Un protocole de communication de données est un ensemble de règles ou convention qui détermine le format et la transmission des données. La couche n d'un ordinateur communique avec la couche n d'un autre ordinateur. Les règles et conventions utilisées lors de cette communication sont collectivement appelées *protocole de couche n* (voir figure 1)

Les protocoles sont des normes qui spécifient la représentation des données transférées d'une machine à une autre. Ils définissent la façon dont la transmission se déroule, dont les erreurs sont détectées et dont les accusés de réception sont transmis. Pour simplifier la conception et la réalisation des protocoles, les problèmes de communication sont décomposés en sous problèmes qui peuvent être traités séparément. Chaque protocole est dédié à la résolution d'un sous problème particulier.

Le principe de structuration en couches, fondamental pour la conception des protocoles, constitue une méthodologie de conception de protocoles de communication. Dans le modèle, chaque couche gère une partie du problème de communication et correspond généralement à un protocole. Les protocoles respectent le principe de structuration en couches qui stipule que les logiciels réalisant la couche n sur la machine destination reçoivent des informations identiques à celles émises par la couche n de la machine source. (figure 1)

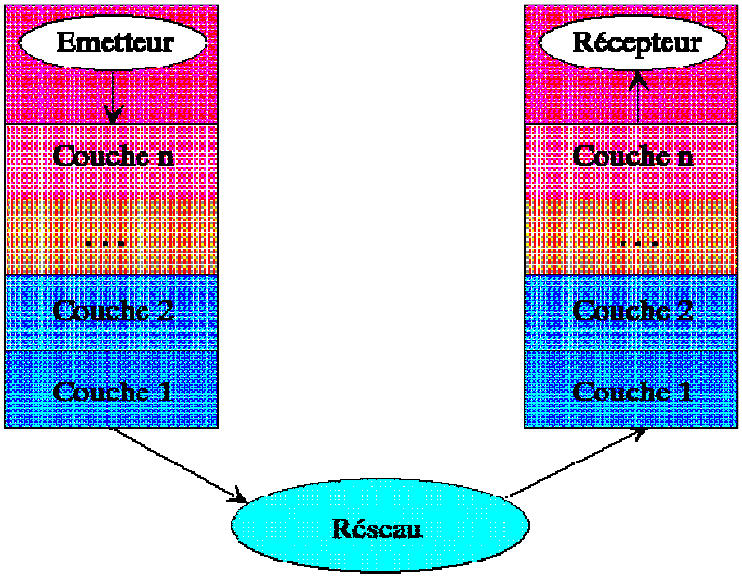


Figure 1 : Les couches conceptuelles de logiciels de communication

4. Le modèle de référence OSI :

a) Introduction :

Le modèle de référence OSI est le principal modèle des communications en réseau. Bien qu'il existe d'autres modèles, la majorité des fournisseurs de réseaux relient aujourd'hui leurs produits à ce modèle de référence, particulièrement lorsqu'ils désirent donner aux utilisateurs la formation sur l'utilisation de leurs produits. Ils le considèrent comme le meilleur outil offert pour décrire l'envoi et la réception de données dans un réseau.

Le modèle de référence OSI me permet de voir les fonctions réseau exécutées à chaque couche. Plus important encore, ce modèle de référence constitue un cadre que me pourrez utiliser pour comprendre comment l'information circule dans un réseau. En outre, je peux servir du modèle de référence OSI pour visualiser comment l'information ou les données, circulent à partir des programmes d'application (ex. : tableurs, documents, etc.), en passant par un média réseau (exemple : fils, etc.), jusqu'à un autre programme d'application se trouvant dans un autre ordinateur en réseau, même si l'expéditeur et le destinataire utilisent des types de réseau différents.

Le modèle de référence OSI comporte sept couches numérotées, chacune illustrant une fonction réseau précise. Cette répartition des fonctions réseau est appelée *organisation en couches*.

b) Les 7 couches du modèle OSI :

Le problème consistant à déplacer de l'information entre des ordinateurs est divisée en sept problèmes plus petits et plus faciles à gérer dans le modèle de référence OSI. Chacun des sept petits problèmes est représenté par une couche particulière du modèle. Voici les sept couches du modèle de référence OSI :

Couche 7 : la couche application

Couche 6 : la couche de présentation
Couche 5 : la couche session
Couche 4 : la couche de transport
Couche 3 : la couche réseau
Couche 2 : la couche liaison de données
Couche 1 : la couche physique

Chaque couche du modèle OSI doit exécuter une série de fonctions pour que les paquets de données puissent circuler d'un ordinateur source à un ordinateur de destination sur un réseau. On trouve à la suite une brève description de chaque couche du modèle de référence OSI qui est illustré dans la figure 2.

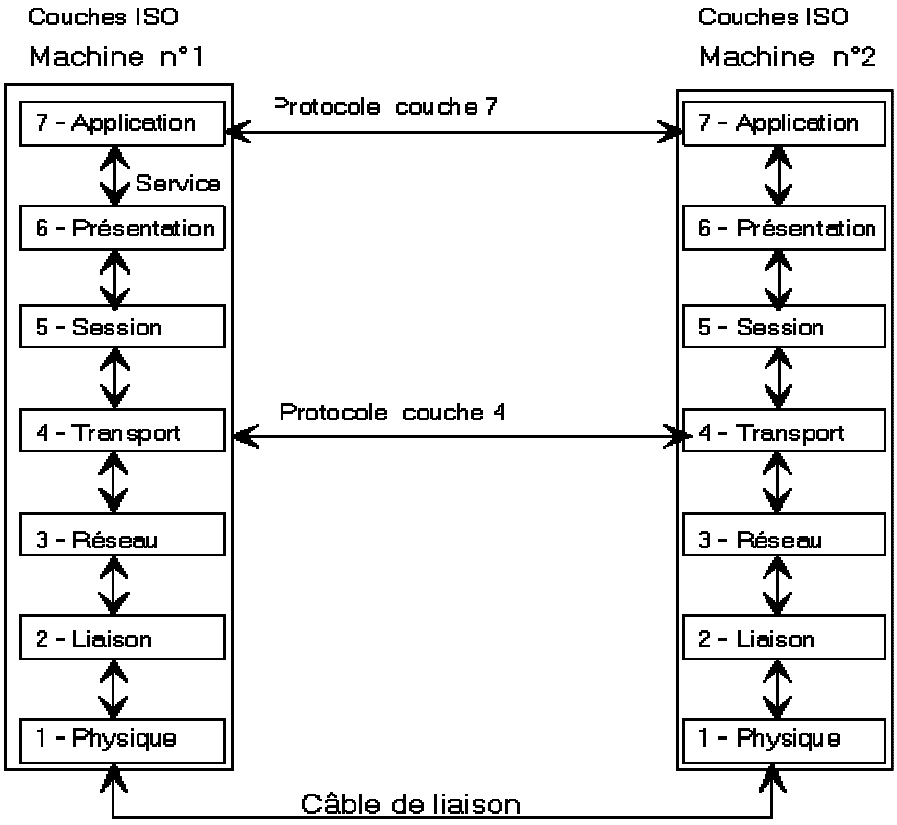


Figure 2 : Les couches du modèle OSI

Couche 7 : La couche application

La couche application est la couche OSI la plus près de l'utilisateur; elle fournit des services réseau aux applications de l'utilisateur. Elle se distingue des autres couches en ce qu'elle ne fournit pas de services aux autres couches OSI, mais seulement aux applications à l'extérieur du modèle OSI. Voici des exemples de ce type d'application : tableurs, traitement de texte et logiciels de terminaux bancaires. La couche application détermine la disponibilité des partenaires de communication voulus, assure la synchronisation et établit une entente sur les procédures de reprise sur incident et de contrôle de l'intégrité des données.

Couche 6 : La couche de présentation

La couche de présentation s'assure que l'information envoyée par la couche application d'un système est lisible par la couche application d'un autre système. Au besoin, la couche de présentation traduit différents formats de représentation des données en utilisant un format commun.

Couche 5 : La couche session

Comme son nom l'indique, la couche session ouvre, gère et ferme les sessions entre deux systèmes hôtes en communication. Cette couche fournit des services à la couche de présentation. Elle synchronise également le dialogue entre les couches de présentation des deux hôtes et gère l'échange des données. En plus de la régulation de la session, la couche session assure également le transfert efficace des données et la classe de service, ainsi que la signalisation des écarts de la couche session, de la couche de présentation et de la couche application.

Couche 4 : La couche de transport
La couche de transport segmente les données envoyées par l'hôte émetteur et les rassemble en flot de données à l'hôte récepteur. La frontière entre la couche session et la couche de transport peut être vue comme la frontière entre les protocoles de couche média et les protocoles de couche hôte. Alors que les couches application, de présentation et de transport se rapportent aux applications, les trois couches qui les suivent se rapportent au transport des données.

La couche de transport tente de fournir un service de transport des données qui protège les couches supérieures des détails d'implantation du transport. Plus particulièrement, les questions comme la façon d'assurer la fiabilité du transport entre deux systèmes hôtes relèvent de la couche de transport. En fournissant un service de communication, la couche de transport établit et raccorde les circuits virtuels, en plus d'en assurer la maintenance. En fournissant un service fiable, elle fait appel à des contrôles de détection des erreurs de transport, de reprise sur incident et de flux d'information.

Couche 3 : La couche réseau
La couche réseau est une couche complexe qui assure la connectivité et la sélection du trajet entre deux systèmes hôte pouvant être situés sur des réseaux géographiquement éloignés.

Couche 2 : La couche liaison de données
La couche liaison de données assure un transit fiable des données sur une liaison physique. Ainsi, la couche liaison de données s'occupe de l'adressage physique (plutôt que logique), de la topologie du réseau, de l'accès au réseau, de la notification des erreurs, de la livraison ordonnée des trames et du contrôle de flux.

Couche 1 : La couche physique

La couche physique définit les spécifications électriques, mécaniques, procédurales et fonctionnelles pour activer, maintenir et désactiver la liaison physique entre les systèmes d'extrémité. Les caractéristiques comme les niveaux de tension, la synchronisation des changements de tension, les débits physiques, les distances maximales de transmission, les connecteurs physiques et autres attributs semblables sont définies par la couche physique.

5. Le Modèle TCP/IP :

a) Introduction :

Même si le modèle de référence OSI est universellement reconnu, historiquement et techniquement, la norme ouverte d'Internet est le *protocole TCP/IP (pour Transmission Control Protocol/Internet Protocol).* (Voir l'annexe A pour plus d'informations sur la comparaison entre ces deux modèles). Le *modèle de référence TCP/IP* et la *pile de protocoles TCP/IP* rendent possible l'échange de données entre deux ordinateurs (voir la figure 3), partout dans le monde, à une vitesse quasi équivalente à celle de la lumière. Le modèle TCP/IP présente une importance historique tout comme les normes qui ont permis l'essor des industries du téléphone, de l'électricité, du chemin de fer, de la télévision et de la bande vidéo.

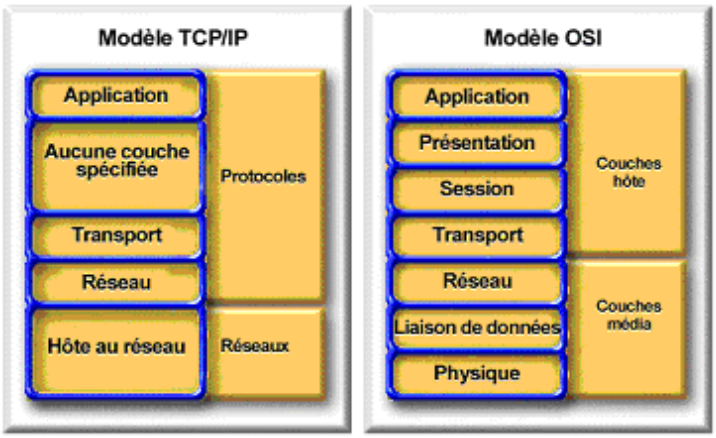


Figure 3 : Présentation des modèles OSI et TCP/IP

En première approximation, le modèle TCP/IP est structuré en quatre couches conceptuelles distinctes, construites au-dessus d'un cinquième matériel. Le schéma de la figure 4 montre les couches conceptuelles ainsi que la structure des données échangées par chaque niveau. La couche nommée interface réseau est parfois appelée couche liaison de données.

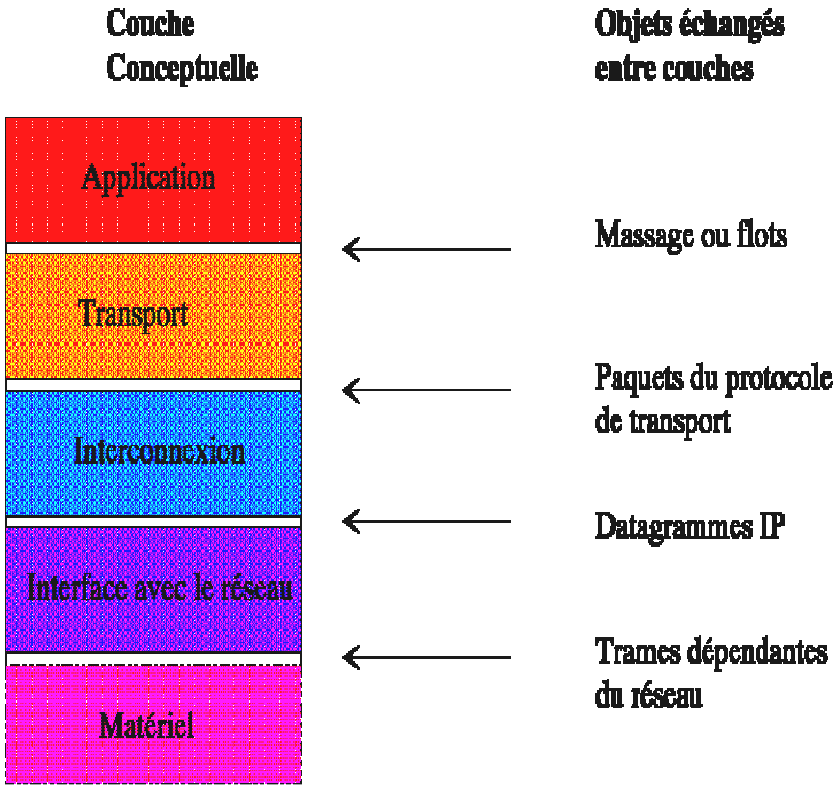


Figure 4 : Le modèle TCP/IP

b) Les 4 couches du modèle TCP/IP :

- **La couche application :**

Les concepteurs de TCP/IP estimaient que les protocoles de niveau supérieur devaient inclure les détails des couches session et présentation. Ils ont donc simplement créé une couche application qui gère les protocoles de haut niveau, les questions de représentation, le code et le contrôle du dialogue. Le modèle TCP/IP regroupe en une seule couche tous les aspects liés aux applications et suppose que les données sont préparées de manière adéquate pour la couche suivante.

La couche application supporte la gestion de réseaux comme la figure 5 le montre. Elle comporte des protocoles pour le transfert de fichiers, le courrier électronique et la connexion à distance.

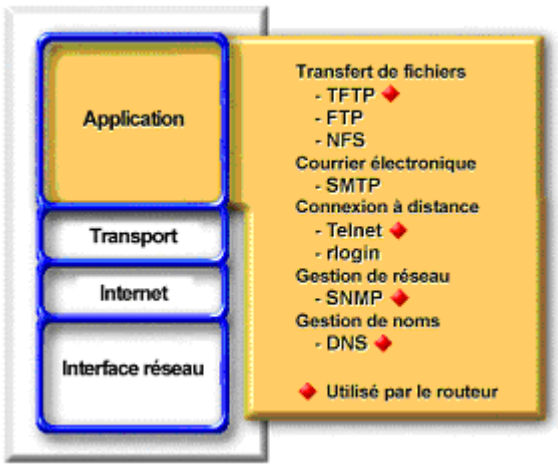


Figure 5 : La couche application

- **La couche de transport :**

La couche de transport est chargée des questions de qualité de service touchant la fiabilité, le contrôle de flux et la correction des erreurs. L'un de ses protocoles, TCP (Transmission Control Protocol - protocole de contrôle de transmission), fournit d'excellentes façons de créer en souplesse des communications réseau fiables, circulant bien et présentant un taux d'erreurs peu élevé.

Le protocole TCP est orienté connexion. Il établit un dialogue entre l'ordinateur source et l'ordinateur de destination pendant qu'il prépare l'information de couche application en unités appelées segments. Un protocole orienté connexion ne signifie pas qu'il existe un circuit entre les ordinateurs en communication (ce qui correspondrait à la commutation de circuits). Ce type de fonctionnement indique qu'il y a un échange de segments de couche 4 entre les deux ordinateurs hôtes afin de confirmer l'existence logique de la connexion pendant un certain temps. C'est ce qu'on appelle la commutation de paquets.

La couche de transport offre également deux protocoles :

- Le protocole TCP, un protocole fiable orienté connexion, assure le contrôle du flux au moyen de fenêtres coulissantes, et la fiabilité, en fournissant des numéros de séquence et des accusés de réception. Le protocole TCP retransmet toute information non reçue et fournit un circuit virtuel entre les applications des utilisateurs finals. L'avantage du protocole TCP est qu'il offre une livraison garantie des segments.

- Le protocole UDP est un protocole non fiable sans confirmation. Bien que chargé de la transmission des messages, ce protocole n'exécute aucune vérification logicielle de la livraison des segments au sein de cette couche.

- L'avantage du protocole UDP est sa vitesse. Puisque le protocole UDP ne fournit pas d'accusés de réception, le trafic sur le réseau est plus faible ce qui assure des transferts plus rapides.

Selon la figure 6, la couche transport est basée sur deux protocoles de base qui sont : les protocoles TCP et UDP.

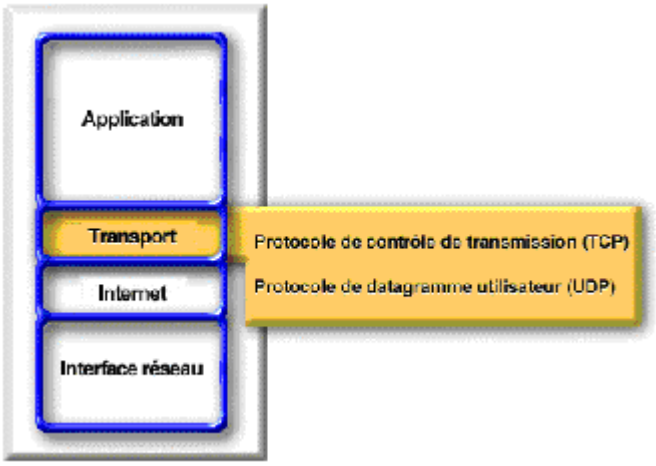


Figure 6 : La couche transport

- **La couche Internet :**

Le rôle de la *couche Internet* consiste à envoyer des paquets source à partir d'un réseau quelconque de l'inter réseau et à les acheminer à destination, indépendamment du trajet et des réseaux traversés pour y arriver. Le protocole qui régit cette couche s'appelle IP (Internet Protocol - protocole Internet) (voir la figure 7)

L'identification du meilleur trajet et la commutation de paquets ont lieu à cette couche. On pense au système postal, lorsqu'on poste une lettre, on ne sait pas comment elle arrive à destination (il existe plusieurs routes possibles), tout ce qui vous importe c'est qu'elle se rende. De nombreux protocoles sont exécutés à la couche Internet de modèle TCP/IP qui correspond à la couche réseau OSI.

- Le protocole IP assure un routage sans confirmation des datagrammes fondé sur le principe de la remise au mieux. Il ne se préoccupe pas du contenu des datagrammes; il cherche simplement une voie afin d'acheminer les datagrammes à destination.
- Le *protocole ICMP* offre des fonctions de contrôle et de transmission de messages.
- Le *protocole ARP* détermine l'adresse de couche liaison de données pour les adresses IP connues.
- Le *protocole RARP* détermine les adresses réseau lorsque les adresses de couche liaison de données sont connues.

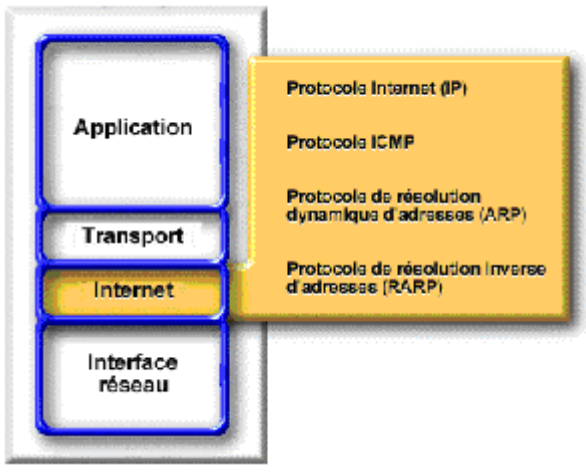


Figure 7 : La couche internet

❖ **Le schéma d'un datagramme IP :**

La structure de datagramme IP est selon la figure 8 :

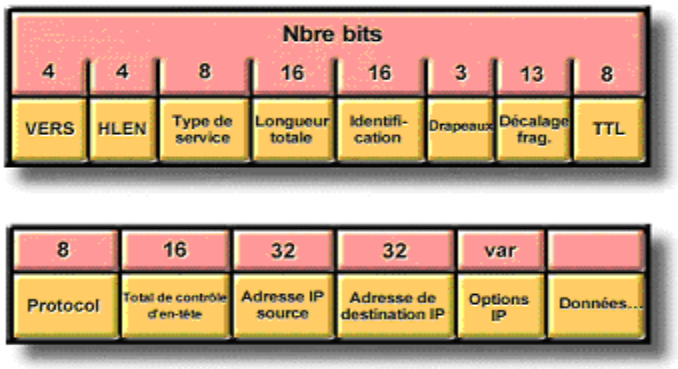


Figure 8 : Le datagramme IP

- *VER :* Numéro de version
- *HLEN*: Longueur de l'en-tête, en mots de 32 bits
- *Type de service :* Mode de traitement du datagramme
- *Longueur totale :* Longueur totale (en-tête + données)
- *Identification, repères, compensation de fragmentation :* Permet de fragmenter les datagrammes pour accommoder divers types de MTU au sein de l'inter réseau.
- *TTL :* Durée de vie minimum
- *Protocole :* Protocole de couche supérieure (couche 4) qui envoie le datagramme.
- *Total de contrôle de l'en-tête :* Contrôle d'intégrité de l'en-tête.
- *Adresse IP source et adresse IP de destination :* Adresses IP de 32 bits
- *Options VIP :* Options de vérification de réseau, de débogage, de sécurité et autres options.

❖ **Le rôle du champ de protocole dans le datagramme IP :**

Le champ de protocole détermine le protocole de couche interface réseau transporté dans un datagramme IP. La majeure partie du trafic IP utilise le protocole TCP, mais d'autres protocoles peuvent aussi faire appel au protocole IP. Chaque en-tête IP doit déterminer le protocole de la couche 4 (interface réseau) de destination pour le datagramme (voir la figure 9). Les protocoles de couche de transport sont numérotés, de façon similaire aux numéros de port. Le protocole IP indique le numéro de protocole dans le champ du protocole.

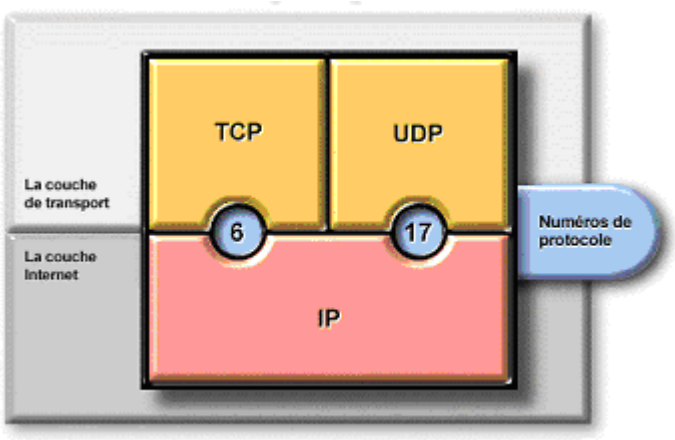


Figure 9 : Le champ de protocole

- **La couche d'accès réseau :**

Le nom de cette couche a un sens très large et peut parfois prêter à confusion. On l'appelle également la couche hôte réseau. Cette couche se charge de tout ce dont un paquet IP a besoin pour établir une liaison physique, puis une autre liaison physique. Cela comprend les détails sur les technologies de réseau local et de réseau long distance, ainsi que tous les détails dans les couches physiques et liaison de données du modèle OSI.

c. La structure de segment TCP :

D'après la figure 10, le segment TCP comprend les champs suivants :

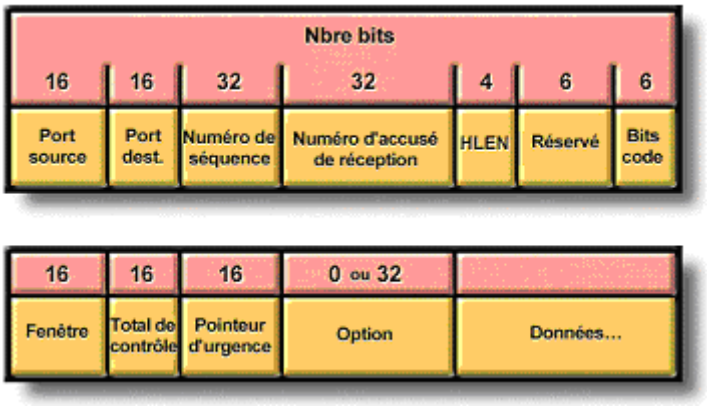


Figure 10 : La structure de segment TCP

- Port source : Numéro du port demandeur.
- Port de destination : Numéro du port demandé.
- Numéro de séquence : Numéro utilisé pour assurer la bonne séquence des données entrantes.
- Numéro d'accusé de réception : Prochain octet TCP attendu.
- HLEN : Nombre de mots de 32 bits contenus dans un en-tête.
- Réservé : Réglé à zéro.
- Bits de code : Fonctions de contrôle (telles l'ouverture et la fermeture de session)
- Fenêtre : Nombre d'octets que l'émetteur est prêt à accepter.

- Total de contrôle : Total de contrôle calculé des champs d'en-tête et de données.
- Pointeur d'urgence : Indique la fin des données urgentes.
- Option : Une seule option est présentement définie : la taille maximale d'un segment TCP.
- Données : Données du protocole de couche supérieure.

d. La structure de segment UDP :

D'après la figure 11, le segment TCP comprend les champs suivants :

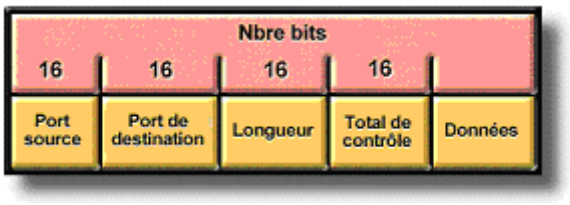


Figure 11 : La structure du segment UDP

Les protocoles de la couche application doivent assurer la fiabilité au besoin. Le protocole UDP n'offre pas de fenêtrage, ni d'accusés de réception. Il est conçu pour les applications qui n'ont pas à assembler des séquences de segments. Voici quelques protocoles qui utilisent le protocole UDP :

- Protocole TFTP
- Protocole SNMP
- Système de fichiers en réseau (NFS)
- Serveur de noms de domaine (DNS)

6. Le Protocole TCP/IP :

a) Introduction :

L'ensemble de protocoles TCP/IP (protocole de contrôle de transmission et protocole Internet) a été développé dans le cadre des recherches de la *DARPA (Defense Advanced Research Projects Agency)*. L'ensemble de protocoles définit des caractéristiques pour les couches 3 et 4 (IP et TCP, par exemple) ainsi que des caractéristiques pour des applications générales telles que le courrier électronique, la connexion à distance, l'émulation de terminal et le transfert de fichiers. Ultérieurement, le protocole TCP/IP a été inclus dans la version Berkeley du système d'exploitation *UNIX*.

Les protocoles Internet permettent de communiquer au sein de tout ensemble de réseaux interconnectés. Ils conviennent aussi très bien à la communication au sein de réseaux locaux et de réseaux longs distance.

b) Fonctionnalités du protocole TCP/IP :

TCP/IP est un système de protocole qui doit assurer les fonctionnalités suivantes :

- Fractionnement des messages en segments faciles à mettre sur la ligne de communication.
- Interface avec la carte réseau.
- Adressage : les messages doivent être envoyés vers la bonne adresse.
- Routage : le système doit acheminer les données sur le sous Réseau auquel appartient le destinataire.
- Contrôle d'erreurs et des flots de données.
- Transfert de données d'une application réseau.

Pour accomplir toutes ces fonctionnalités, TCP/IP adopte une approche modulaire dont chaque module est chargé d'une tache précise dans le fonctionnement du réseau de communication.

c) Utilitaires TCP/IP :

La suite de protocole TCP/IP met à disposition des services (applications) comme Telnet (terminal à distance), FTP (File Transfert Protocol, pour l'échange de fichiers), NFS (Network File System, de SUN, pour permettre le partage de fichiers dans un environnement de type 'Bureautique'), LP (Line Printer, pour les impressions), SMTP (Simple Mail Transfer Protocol, pour la messagerie), HTTP (HyperText Transfer Protocol - Worl-Wide Web) etc.

Utilitaires de Transfert de données :

- ***FTP*** (File Transfert Protocol) : Assure le transfert bidirectionnel de fichiers entre deux hôtes TCP/IP même faisant partis d'environnements hétérogènes. Il utilise le protocole TCP qui est sécurisé. Attention, pour que cela fonctionne il faut qu'au moins l'un des deux hôtes exécute le logiciel serveur. C'est notamment l'utilitaire qui permet d'échanger des fichiers entre NT et Unix.

- ***TFTP*** (Trivial File Transfert Protocol) : Il est de même que FTP mais se base sur le protocole UDP qui lui n'est pas sécurisé. Attention, pour que cela fonctionne il faut qu'au moins l'un des deux hôtes exécute le logiciel serveur.

- ***RCP*** (Remote Copy Control) : Copie des fichiers entre un hôte (machine) NT et un hôte Unix.

Utilitaires d'exécution à distance :

- ***TELNET:*** Assure l'émulation de terminal pour un hôte TCP/IP exécutant le logiciel serveur Telnet. Ne fonctionne pas sur un serveur NT. Telnet est utilisé par exemple, pour visualiser des données sur un ordinateur Unix à partir d'un ordinateur NT. Il faut cependant qu'il puisse utiliser un serveur DNS ou un fichier HOST.

- ***RSH*** (Remote Shell) : Exécute des commandes sur un hôte Unix.

- ***REXEC*** (Remote Execution) : Exécute une procédure sur un ordinateur distant.

Utilitaires de diagnostiques :

- ***PING*** *(Packet Internet Groper) :* Vérifie que TCP/IP est correctement configuré et qu'un hôte est disponible.

- ***IPCONFIG:*** Vérifie la configuration TCP/IP, l'adresse des serveurs DHCP, WINS, DNS.

- ***HOSTNAME:*** Renvoie le nom d'hôte de l'ordinateur local.

- ***NETSTAT:*** Affiche les statistiques du protocole ainsi que l'état courant des connexions TCP/IP. Sur un serveur NT, Netstat affiche aussi la table de routage. De plus, si l'on désire afficher les statistiques Ethernet et TCP/IP, c'est Netstat qu'il faut utiliser.

- ***NBTSTAT:*** Vérifie l'état actuel de NetBios sur les connexions TCP/IP et met à jour le cache **LMHOSTS :** Permet de mettre à jour le cache Lmhosts ou de déterminer le nom et l'identificateur d'étendue.

- ***ROUTE:*** Affiche ou modifie la table de routage locale.

- ***TRACERT:*** Vérifie les routes empruntées entre l'hôte local et l'hôte distant.

- ***ARP*** (Address Resolution Protocol) : Affiche un cache d'adresses IP résolues en adresses MAC (Media Access Control).

Utilitaires de Configuration :

Parmi les fichiers de résolutions de nom on distingue:

- **HOSTS:** Permet de résoudre les noms d'hôtes en adresse IP.

- **LMHOSTS:** Permet de résoudre les noms NetBios en adresse IP.

- **NETWORKS:** Permet de résoudre les noms réseaux en identificateur de réseau IP.

IV. Les réseaux locaux :

1. Les différents types de réseaux :

On distingue différents types de réseaux (privés) selon leur taille (en terme de nombre de machine), leur vitesse de transfert des données ainsi que leur étendue. Les réseaux privés sont des réseaux appartenant à une même organisation. Il existe trois catégories de réseaux (voir figure 12):

- LAN (local area network)
- MAN (metropolitan area network)
- WAN (wide area network)

a) Les réseaux locaux LAN (Local Area Network) :

Ces réseaux sont en général circonscrits à un bâtiment ou à un groupe de bâtiment pas trop éloignés les uns des autres (site universitaire, usine ou campus). L'infrastructure est privée et est gérée localement par le personnel informatique. De tels réseaux offrent en général une bande passante comprise entre 4Mbit/s et 100 Mbits/s.

b) Les réseaux métropolitains MAN (Metropolitan Area Network) :

Ce type de réseau est apparu relativement récemment et peut regrouper un petit nombre de réseaux locaux au niveau d'une ville ou d'une région. L'infrastructure peut être privée ou publique. Par exemple, une ville peut décider de créer un 'MAN' pour relier ses différents services disséminés sur un rayon de quelques kilomètres et en profiter pour louer cette infrastructure à d'autres utilisateurs. La bande passante peut être de quelques centaines de kbits/s à quelquesMbits/s.

c) Les réseaux distants WAN (Wide Area Network) :

Ce type de réseau permet l'interconnexion de réseaux locaux et métropolitains à l'échelle de la planète, d'un pays, d'une région ou d'une ville. L'infrastructure est en général publique (PTT, Télécom etc.) et l'utilisation est facturée en fonction du trafic et/ou en fonction de la bande passante réservée, pour les lignes louées (une ligne louée est réservée exclusivement au locataire, 24h sur 24, pour la durée du contrat).

Les modems sont un des éléments de base des WANs et la bande passante va de quelques kbits/s à quelques Mbit/s. Une valeur typique pour une ligne louée est de 64kbits/s (en fonction des services offerts).

Au cours de mon projet je vais s'intéresser plus aux réseaux locaux LAN, dont me pouvons tirer quelques spécificités par rapport aux autres réseaux.

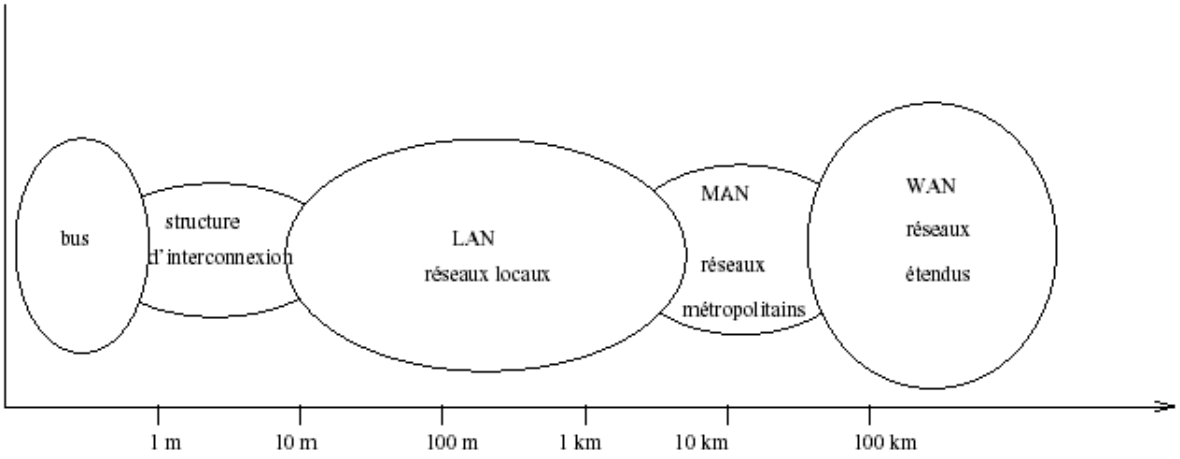


Figure 12 : La Classification des réseaux informatiques selon leur taille

2. Objectifs des réseaux locaux :

Un réseau est un dispositif qui permet d'interconnecter différents matériels informatiques. Les réseaux Locaux appelés aussi LAN (Local Area Network) diffèrent des MAN (Metropolitan Area Network) et des WAN (World Area Network) par leurs objectifs et caractéristiques.

Bien que pour les LAN ce soit la technique qui ait précédé les besoins (au départ on ne savait pas que faire d'une telle vitesse de transmission et on ne pensait vraiment pas pouvoir un jour les saturer) l'objectif était clair dès le départ. En effet le but premier d'un réseau local est économique : il permet de partager les ressources matérielles et logicielles entre les différents postes et utilisateurs.

L'avantage du partage des ressources matérielles est évident, les besoins en imprimantes, disques et mêmes processeurs sont amoindris. Celui des ressources logicielles est moins flagrant sans être pour autant négligeable : on peut partager un fichier de configuration et ainsi gagner du temps d'administration, on peut aussi partager les logiciels afin de ne pas acheter autant de licences que de postes.

Les autres objectifs, communiquer à hauts débits, répartir les traitements, connecter tout le monde, etc. découlent de cet objectif principal: le partage pour l'économie. L'amélioration du rendement est en fait une conséquence de cet objectif.

3. Architecture d'un réseau local :

a) Introduction :

L'architecture de réseau comprend sa structure globale ainsi que tous les éléments qui permettent son fonctionnement, notamment le matériel et le logiciel système. Pour définir une architecture réseau, on doit définir les spécifications suivantes :

- Les principales fonctionnalités de conception.
- Les paramètres de performances.
- Les configurations matérielles et logicielles
- Les projets de mise en œuvre d'un réseau.

Les principales architectures réseaux connues sont : Ethernet, Token Ring.

b) L'architecture Ethernet :

Ethernet est actuellement l'architecture de réseau la plus répondue. Cette architecture en bande de base utilise une topologie en bus (voir figure 18) qui est caractérisé par :

- Débit de 10 Mbit/s à 1 Gbit/s.
- Transmission en bande de base, codage Manchester.
- Topologie logique en bus et physique en étoile.
- Méthode d'accès suivant la norme IEEE 802.3 : CSMA/CD.
- La longueur de trame est comprise entre 64 et 1518 octets.
- Le support est de type câble coaxial, paire torsadée ou fibre optique.

c) L'architecture Token Ring :

Chaque station est reliée à sa suivante et à sa précédente par un support unidirectionnel.
Un réseau Token Ring pose des caractéristiques suivantes :

- Débit de 4 Mbit/s à 16 Mbit/s
- Transmission en bande de base.
- Topologie de l'anneau en étoile.
- Passage du jeton comme méthode d'accès.
- Le support est la paire torsadée blindée et non blindée.

V. Topologie des réseaux :

1. Introduction :

Un réseau informatique est constitué d'ordinateurs reliés entre eux grâce au matériel (câblage, cartes réseau, ainsi que d'autres équipements permettant d'assurer la bonne circulation des données). L'arrangement physique de ces éléments est appelé topologie physique. Il en existe trois:

- La topologie en bus
- La topologie en étoile
- La topologie en anneau

On distingue la topologie physique (la configuration spatiale, visible, du réseau) de la topologie logique. La topologie logique représente la façon de laquelle les données transitent dans les câbles. Les topologies logiques les plus courantes sont Ethernet et Token Ring.

2. La topologie en bus :

Tous les équipements sont branchés en série sur le serveur. Chaque poste reçoit l'information mais seul le poste pour lequel le message est adressé traite l'information. On utilise un câble coaxial pour ce type de topologie. (Voir figure 13)

L'avantage du bus est sa simplicité de mise en œuvre et sa bonne immunité aux perturbations électromagnétiques. Par contre, si le câble est interrompu, toute communication sur le réseau est impossible.

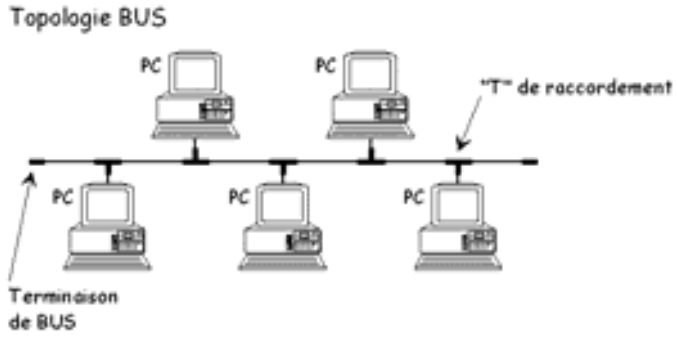


Figure 13 : La topologie en bus

3. La topologie en étoile :

Dans cette topologie, toutes les liaisons sont issues d'un point central. C'est une liaison dite « point à point », c'est à dire que les équipements sont reliés individuellement au nœud central et ne peuvent communiquer qu'à travers lui. On utilise les câbles en paires torsadées ou en fibre optique pour ce type de topologie. (Voir figure 14)

L'avantage est que les connexions sont centralisées et facilement modifiables en cas de défectuosité. Si un câble est interrompu, le reste du réseau n'est pas perturbé.

L'inconvénient de cette topologie est l'importante quantité de câbles nécessaire.

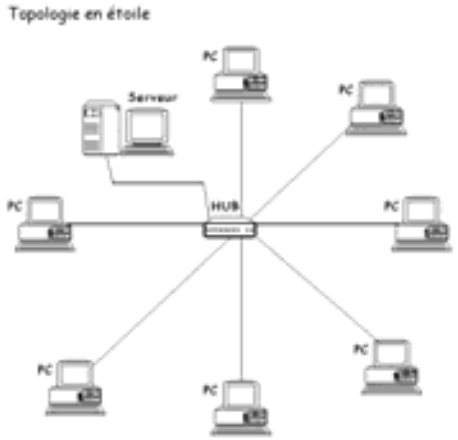


Figure 14 : La topologie en étoile

4. La topologie en anneau :

Les équipements sont reliés entre eux en formant une boucle. La liaison entre chaque équipement est point à point. L'information est gérée comme dans la topologie bus. Chaque station reçoit le message, mais seule la station à qui le message est adressé le traite. Pour le câblage, on utilise un câble en paires torsadées ou de fibre optique. (Voir figure 15)

L'avantage est que l'anneau offre deux chemins pour aller d'un point à l'autre. Ceci permet à l'information de passer malgré une coupure sur le câble. On utilise cette topologie pour les réseaux de type Token Ring. Pour augmenter la sécurité, on peut utiliser un double anneau (si le premier anneau est interrompu, les données passent sur l'anneau secondaire, le temps de réparer le premier anneau).

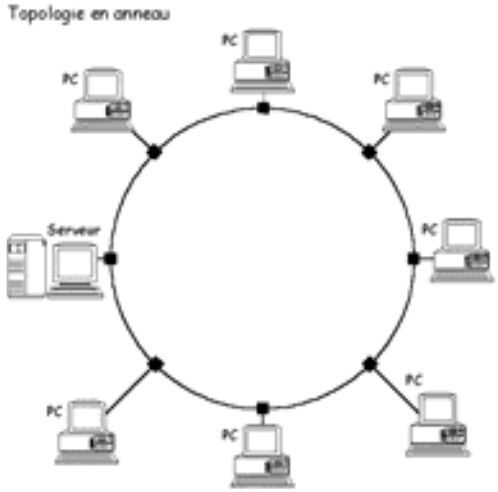


Figure 15 : La topologie en anneau

VI. Supports de transmission :

Pour relier les diverses entités d'un réseau, plusieurs supports physiques de transmission de données peuvent être utilisés. Une de ces possibilités est l'utilisation de câbles. Il existe de nombreux types de supports, mais on distingue généralement:

- **Le câble coaxial :** Proche du câble qui relie le téléviseur à son antenne, le câble coaxial est composé d'un câble central entouré d'un isolant, lui-même recouvert d'une tresse métallique, elle-même recouverte d'un isolant. Il permet des vitesses de transmission bien plus élevées que la paire torsadée et des connexions à plus grande distance. Il reste néanmoins assez coûteux.

- **La Paire torsadée :** C'est le même câble utilisé pour les téléphones. Il existe des câbles à 2 ou 4 paires mais aussi des câbles blindés (STP) ou non blindés (UTP), ce type de câbles est utilisé pour du câblage dit universel mais aussi pour les réseaux Token Ring (anneau à jeton) ou Ethernet. C'est une solution économique mais limitée. La paire torsadée ne permet pas une grande vitesse de transmission de l'information et elle est en outre très sensible à l'environnement électromagnétique.

- **La fibre optique :** C'est le nec plus ultra des médias télématiques véhiculant des impulsions lumineuses (et non pas électromagnétiques), elle n'est absolument pas sensible aux perturbations pouvant affecter les autres supports. La fibre optique permet d'aller jusqu'au plusieurs kilomètres avant que l'information ne subisse de graves détériorations et nécessite d'être restaurée (tous les km pour le câble coaxial).

- **Les ondes hertziennes :** Elles supportent de grande distance et de grandes capacités, pour une propagation en visibilité directe (entre 50 et 80 km). Elles prolongent et remplacent les câbles, pour une plus grande souplesse mais aussi une plus grande sensibilité au bruit.

VII. Les équipements d'interconnexion :

1. Introduction :

Un réseau local sert à interconnecter les ordinateurs d'une organisation, toutefois une organisation comporte généralement plusieurs réseaux locaux, il est donc parfois indispensable de les relier entre eux. Dans ce cas, des équipements spécifiques sont nécessaires.

Lorsqu'il s'agit de deux réseaux de même type, il suffit de faire passer les trames de l'un sur l'autre. Dans le cas contraire, c'est-à-dire lorsque les deux réseaux utilisent des protocoles différents, il est indispensable de procéder à une conversion de protocole avant de transférer les trames. Ainsi, les équipements à mettre en oeuvre sont différents selon la configuration face à laquelle on se trouve.

L'évolution du réseau local vers le réseau d'entreprise implique différentes techniques fondamentales d'interconnexion pour faire évoluer la taille et l'architecture d'un réseau, augmenter le flux de communications, interconnecter plusieurs réseaux locaux situés localement ou sur des sites distants.

2. Les répéteurs :

Ces systèmes permettent l'interconnexion de médias similaires ou différents pour une méthode d'accès donnée et assurent ainsi une continuité de la topologie physique pour constituer un réseau local unique (voir figure 16).

Le répéteur est l'équipement clé du réseau Ethernet pour tout ce qui concerne la taille du réseau ou le changement de média. C'est le matériel de plus bas niveau sur le réseau local. Il n'interprète pas les trames qu'il reçoit et se contente de les retransmettre bit à bit sur les autres segments.

La principale fonction du répéteur est régénération du signal, en effet le signal subit une atténuation et une distorsion tout au long de sa propagation dans le câble. Le répéteur émet les signaux reçus en les remettant en forme.

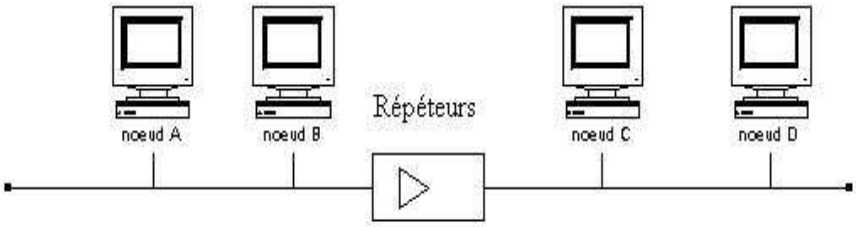


Figure 16 : Interconnexion par répéteur

3. Les ponts (Bridge) :

Ils réalisent la connexion entre deux réseaux locaux de type différent. Ils permettent de gérer des liaisons locales ou distantes pour réaliser l'interconnexion des réseaux (voir figure 17) et optimiser les flux de communications en plus ils offrent une possibilité d'extension au-delà des limites imposées par la norme Ethernet ou Token Ring.

Au niveau de sécurité les ponts sont plus puissants que le répéteur car ils peuvent éviter la propagation de certains défauts, en plus ils filtrent les trames entre deux réseaux.

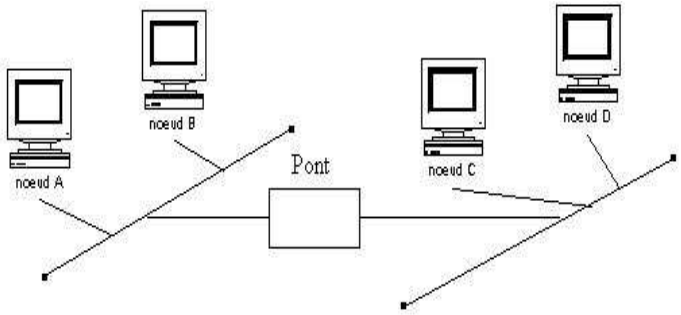
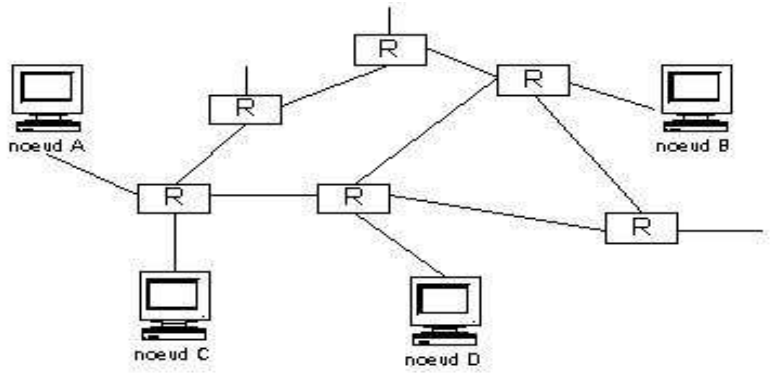


Figure 17 : Interconnexion par pont

4. Les routeurs :

Les routeurs sont des éléments actifs qui permettent d'interconnecter localement ou à distance des réseaux entre eux. Agissant au niveau 3 de l'OSI, ils proposent un certain nombre de mécanismes évolués permettant de déterminer le meilleur chemin pour assurer l'acheminement de l'information (voir figure 18).

Le mécanisme d'un routeur est lié aux protocoles supportés par les réseaux dont il réalise l'interconnexion puisque les fonctionnalités de routage des paquets interviennent au niveau de la couche réseau et mettent en œuvre des algorithmes de routage pour assurer l'acheminement des paquets à leur destination finale.



5. Les Hubs (concentrateurs) :

Les Hubs (concentrateurs) permettent la connexion de plusieurs noeuds sur un même point d'accès sur le réseau, en se partageant la bande passante totale.

La structure physique qui s'en dégage est une étoile, mais la topologie logique reste un bus (pour Ethernet).

6. Le commutateur (ou switch) :

Le commutateur (switch) est un système assurant l'interconnexion de stations ou de segments d'un réseau local en leur attribuant l'intégralité de la bande passante, à l'inverse du concentrateur qui la partage.

Les commutateurs ont donc été introduits pour augmenter la bande passante globale d'un réseau d'entreprise et sont une évolution des concentrateurs Ethernet (hubs). Ils ne mettent en oeuvre aucune fonctionnalité de sécurité, hormis l'amélioration de la disponibilité.

7. La passerelle (Gateway) :

C'est un système complet du point de vue de la connexion puisqu'elle rassemble au routeur qui assure l'interconnexion des réseaux hétérogènes, elle fait de la conversion de protocole. C'est la seule qui travaille jusqu'à la
7ème couche du modèle OSI (couche application).

VIII. Conclusion :

Ce chapitre a une grande importance dans mon travail. J'ai essayé de présenter aux lecteurs informaticiens et non informaticiens quelques informations de base sur le thème informatique, qui sont utile à la suite pour comprendre le concept de réalisation de mon projet.

CHAPITRE 3

Contrôleur de trafic

« La seule façon de renforcer notre intelligence est de n'avoir d'idées sur rien, de laisser l'esprit accueillir toutes les pensées »

Jhon Keats

I. Définition :

Les logiciels de contrôleur de trafic sont nés avec les réseaux. C'est une catégorie de produits mature dont nombreuses propriétés sont disponibles avec la plupart de ces logiciels. L'ensemble de ces outils aide en mettant en application l'intelligence incorporée qui informe l'utilisateur de l'état du réseau et de tout problème pouvant émerger. Ces fonctions éliminent le besoin de naviguer parmi des milliers de paquets, par exemple localiser les ordinateurs qui envoient d'importantes quantités de données ou qui ont d'autres problèmes de communication.

L'analyseur est dit « de protocole » parce que pour intercepter, décoder et analyser une trame, il faut savoir de quel protocole elle relève. Il permet à un administrateur de réseau d'examiner les trames échangées entre deux dispositifs de réseau à des fins d'investigation (en cas d'affaiblissement des débits, notamment).

Il y a plusieurs tests que je peux effectuer avec les outils d'analyse de réseaux :

- Localisation des problèmes.
- Détection des nœuds sur le réseau.
- Détermination de ceux générant le plus de trafic.
- Détermination de la structure habituelle du trafic et des écarts par rapport à cette structure.
- Génération des rapports.

III. Exemples d'analyseurs :

1. Introduction :

Pour avoir une idée plus claire et précise sur les contrôleurs de trafic, leurs fonctionnalités et ses opérations standard, j'ai testé quelques produits connus comme : Analyzer, Ethereal, CommView et RadCom (le contrôleur utilisé de l'Institut) sur mon petit réseau représenté en un segment isolé de deux ordinateurs.

2. Analyzer :

Analyzer est un outil de la capture des paquets, il capture des paquets de réseau et il les affiche à travers une interface graphique. L'utilisateur peut choisir l'adaptateur du réseau, spécifie un filtre approprié, sélectionne, copie de paquets de la pâte (voir figure 19). L'inquiétude des traits avancée les deux la possibilité de faire (et intrigue) quelques

statistiques avancées sur les paquets des captures et comploter des statistiques dans le vrai temps (nombre de paquets qui coulent à travers le réseau et ainsi de suite).

Voici l'interface Analyzer :

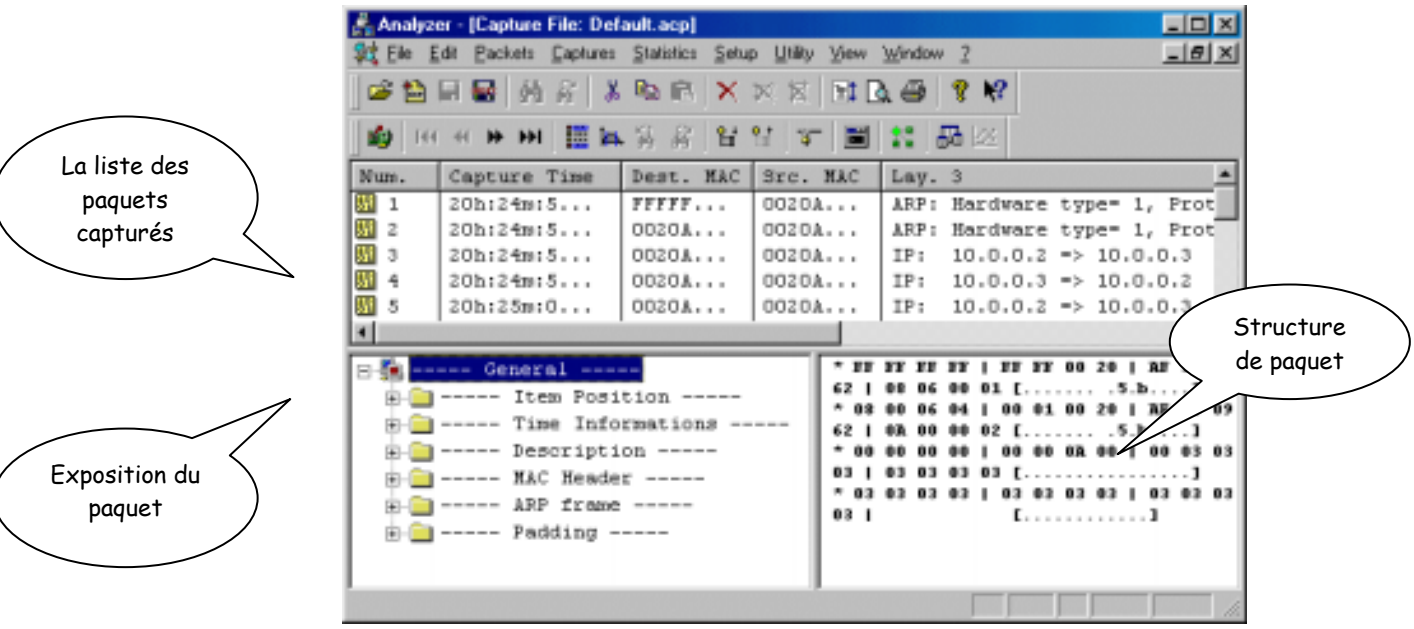


Figure 19 : Description d'Analyzer

- La première fenêtre montre l'index du paquet: c'est la liste des paquets qui appartiennent au document de la capture.
- La fenêtre sur les expositions gauches le paquet sélectionné dans l'index du paquet; ce paquet est montré à travers une vue de l'arbre.
- La fenêtre montre l'hexadécimal à droite (une vue du paquet sélectionné).

3. Ethereal :

a) Introduction : (voir site http://www.ethereal.com)

Il existe un outil sous licence GNU qui permet de faire cela et qui permet d'interpréter la structure des paquets, cela de façon graphique, cet outil s'appelle **Ethereal**.

L'Ethereal comporte plusieurs phases (voir figure 22) on peut citer les trois principaux phases qui sont :

- Phase de capture.
- Phase d'affichage.
- Phase de filtrage.

b) Phase de capture :

Le principe est simple, on doit lancer une session de capture à l'aide du menu **Capture**. Cette session peut être interactive ou pas. En d'autres termes, les paquets capturés peuvent être affichés au fur et à mesure ou à la fin de la capture (voir figure 20). Pour lancer une session de capture, il faut accéder au menu **Capture** puis cliquer sur l'option **Start**...

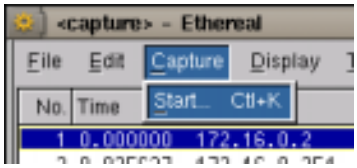


Figure 20 : Lancement de capture

Apparaît la boîte de dialogue (voir figure 21) qui permet de spécifier ce qui doit être analysé.

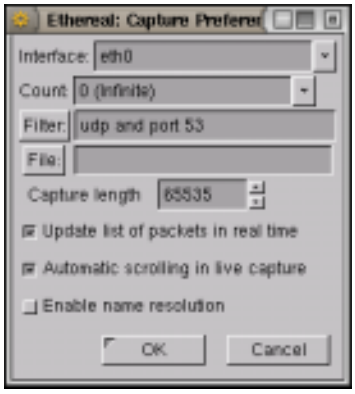


Figure 21 : Paramètres de capture

c) Phase d'affichage :

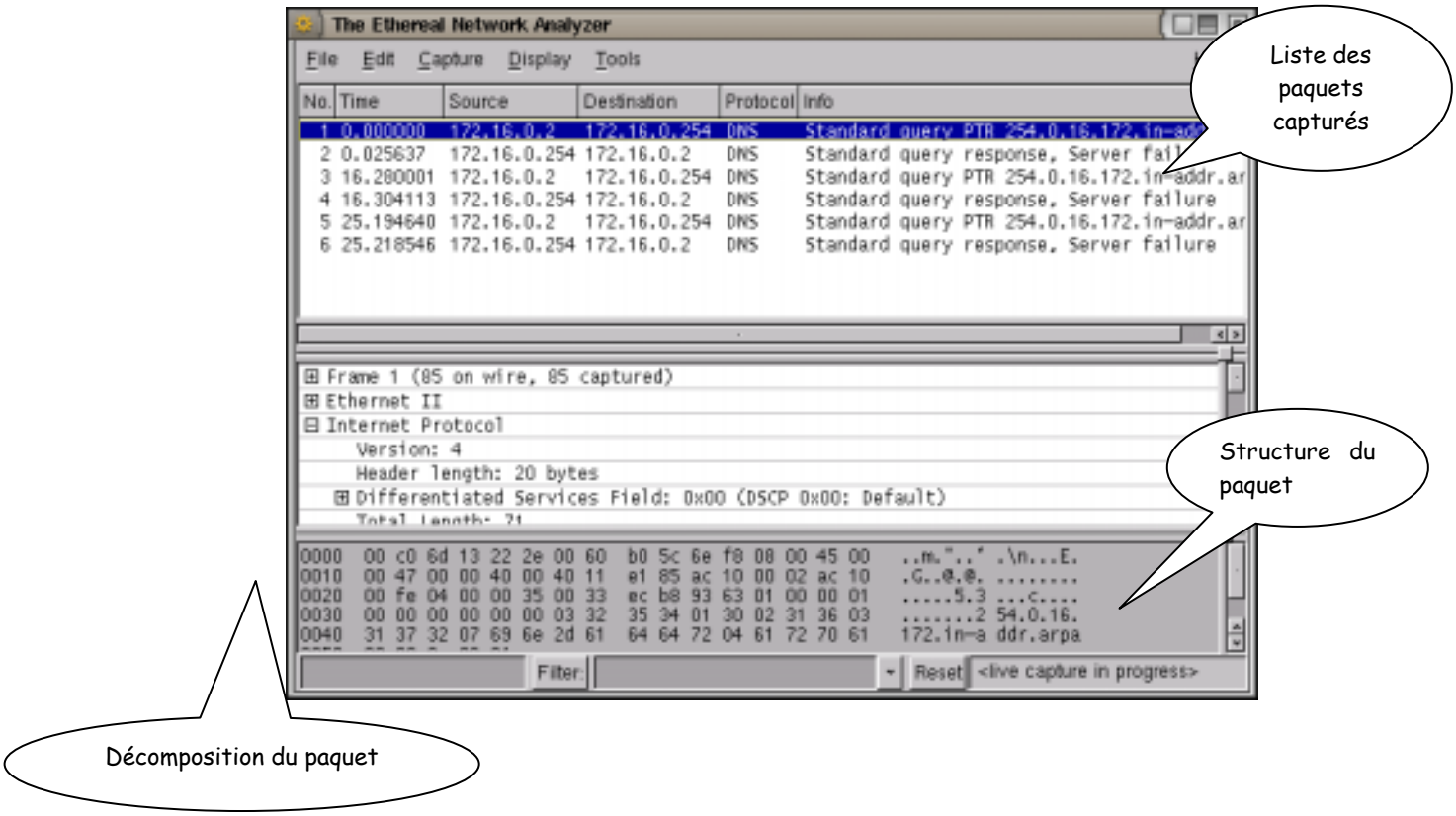


Figure 22 : Description du Ethereal :

L'affichage des résultats se décompose en trois parties :

- La liste des paquets capturés disponibles en dessous de la barre de menu avec un affichage synthétique du contenu de chaque paquet.
- La décomposition exacte du paquet actuellement sélectionné dans la liste. Cette décomposition permet de visualiser les champs des entêtes des protocoles ainsi que l'imbrication des différentes couches de protocoles connus.
- La troisième zone contienne le paquet (le début s'il est trop gros) affiché en hexadécimal et en ASCII.

Tout en bas du programme se trouve un champ **filter**. Il permet de n'afficher que les paquets qui correspondent aux critères spécifiés dans ce champ **filter**. C'est un filtre qui permet de temporairement cacher une partie des paquets.

d) Phase de filtrage :

Il y a deux sortes de filtres, les filtres à la capture et les filtres à l'affichage. Ces filtres n'ont pas la même syntaxe.

i. Filtres de capture :

Ne seront gardés que les paquets pour lesquels le filtre est vrai. Les filtres se décomposent en 3 parties :

- Le **protocole** qui peut être **ether**, **ip**, **arp**, **rarp**, **tcp, udp**, etc..
- La **direction** qui peut être **src (source)** ou **dst (destination)**

Un **champ** qui peut être **host**, **net** ou **port** suivi d'une valeur.

ii. Filtres d'affichage :

Les filtres d'affichage sont un peu plus fins que ceux de la capture. Seuls les paquets pour lesquels l'expression du filtre est vraie seront gardés. Les expressions sont basées sur les champs disponibles dans un paquet. Le simple ajout d'un champ veut dire que l'on garde le paquet si ce champ est disponible.

4. CommView :

a) Introduction :

CommView est un programme conçu pour contrôler les activités des réseaux Internet et des réseaux locaux, capable de capturer et d'analyser les paquets de réseau. CommView réunit les informations sur les données qui passent par une connexion modem à accès commuté ou une carte Ethernet, puis décode les données analysées(voir figure 23).

Avec CommView, on peut visionner la liste des connexions réseau, ainsi que les statistiques IP vitales, puis examiner les paquets individuellement. Les paquets sont décodés au niveau le plus bas, accompagnés d'une analyse complète des protocoles les plus communs. Un accès entier aux données non traitées est aussi fourni. Les paquets capturés peuvent être enregistrés, afin d'enregistrer les fichiers pour fins d'analyse future. Un système flexible de filtres permet l'abandon de paquets dont on n'a pas besoin ou de la capture de seulement ceux qu'on souhaite capturer. Des alarmes configurables peuvent vous notifier à propos d'événements importants, tels que des paquets suspects, une utilisation élevée de la bande passante ou des adresses inconnues.

CommView est un outil utile pour les administrateurs de réseau local les professionnels en sécurité, les administrateurs de réseau ou toute personne souhaitant avoir une image nette du trafic circulant sur un ordinateur ou un segment de réseau local. Cette application est conçue pour les utilisateurs d'Internet, ainsi que pour les petits et moyens réseaux.

b) Le programme :

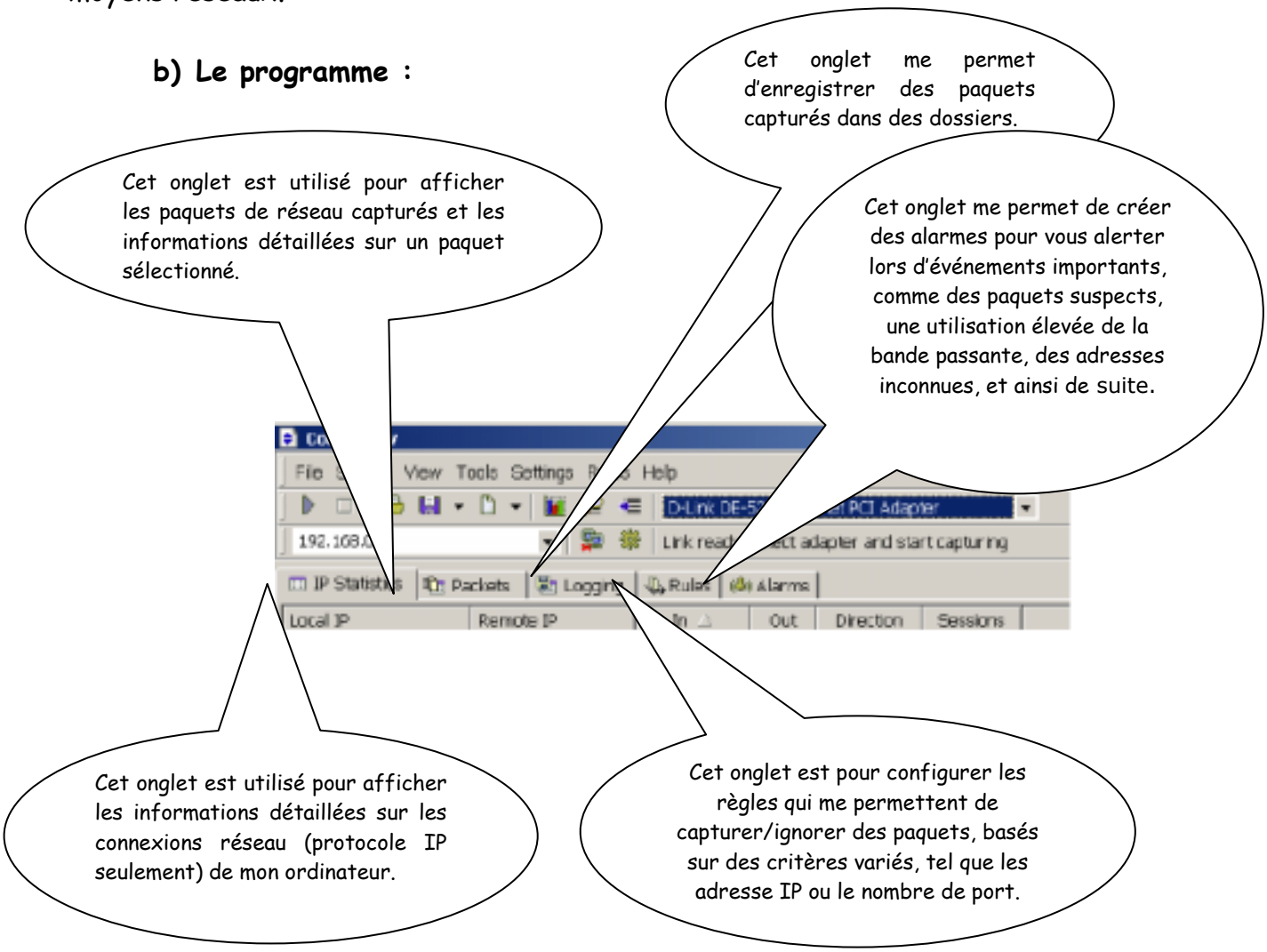


Figure 23 : Description de Commview

L'interface du programme consiste en cinq onglets qui me permettent de visionner les données et d'exécuter des actions variées avec les paquets capturés (voir figure). Pour commencer à capturer les paquets, on doit sélectionner un adaptateur de réseau à partir de la liste du menu déroulant sur la barre d'outils, puis on clique sur le bouton **Démarrer la capture** ou on sélectionne **Fichier = > Démarrer la capture** à partir du menu. Si le trafic

du réseau passé par l'adaptateur de réseau est sélectionné, CommView va commencer à afficher les informations.

c) Phase de capture et d'affichage :

Pour commencer à capturer des paquets, on sélectionne **Fichier => Démarrer la capture** à partir du menu ou on clique le bout j'ai approprié sur la barre d'outils.

Cet onglet (voir figure 24) est utilisé pour répertorier tous les paquets réseau capturés et afficher les informations détaillées à propos d'un paquet sélectionné.

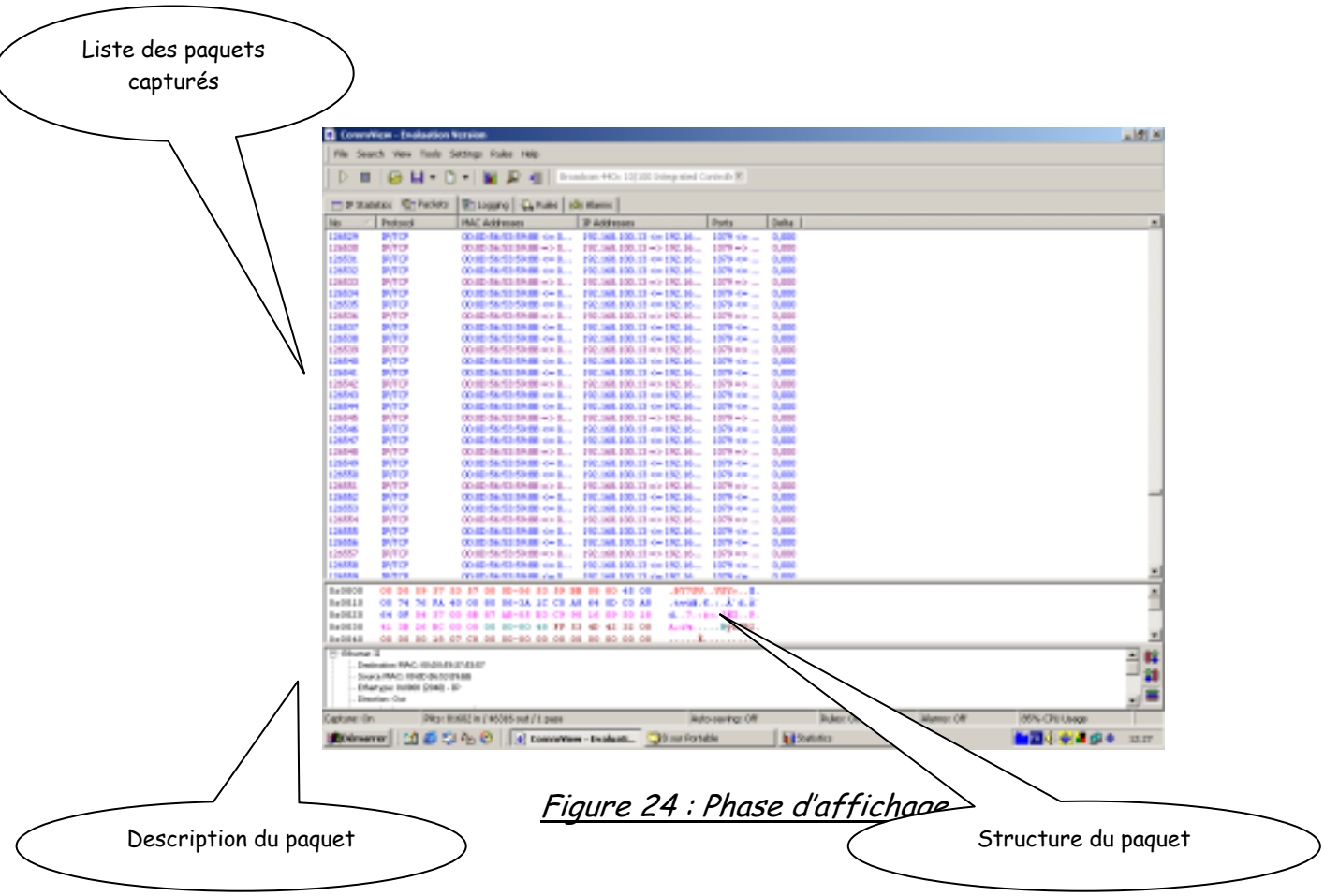


Figure 24 : Phase d'affichage

d) Phase de statistique :

La fenêtre de la figure (**Affichage => Statistiques**) affiche les statistiques vitales réseau de notre ordinateur ou segment de réseau local (voir les figures 25 et 26), telles que le taux de paquets par seconde, le taux d'octets par seconde, et les graphiques de distribution de protocoles et sous protocoles IP.

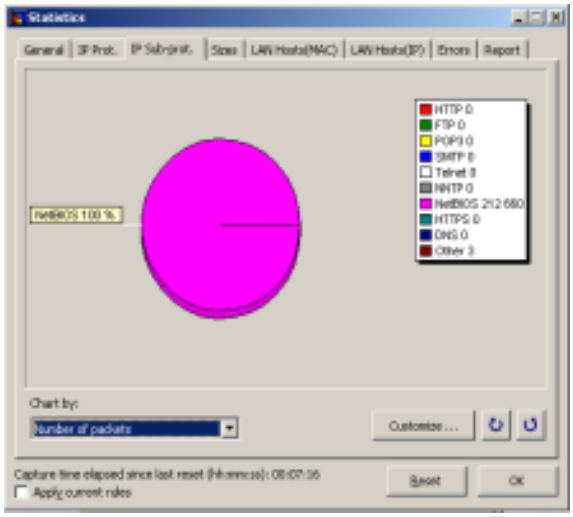


Figure 25 : Statistique selon les protocoles

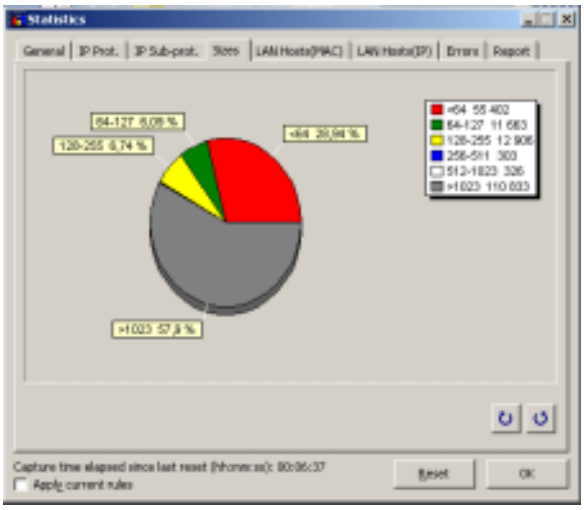


Figure 26 : Statistique selon la taille des paquets

Les statistiques réseau peuvent être collectionnées en utilisant toutes les données transitant sur notre adaptateur réseau ou en utilisant les règles qui sont actuellement configurées.

5. RadCom (RCW-100 FL):

RadCom, c'est un type d'analyseur qui est utilisé et testé au sein de l'ISCI, ce produit permet de descendre aux plus basses couches d'un réseau, une sonde que l'on place entre les deux dispositifs dont on veut contrôler le dialogue et qui procéder au décodage des trames, pour examen ultérieur à l'aide du logiciel approprié.

L'avantage énorme pour ce produit par rapport à tous les autres produits qu'j'ai testé qu'il peut analyser les réseaux étendues WAN, il a la possibilité de détecter les nœuds externes du réseau comme les routeurs (voir figure 27).

Pour des raisons de **sécurité de l'Institut**, j'ai effacé les adresses IP sources et destinations.

Figure 27 : Journal des adresses IP sources et destinations inclus dans le réseau

Comme les autres produits testés, RadCom permet de visualiser la structure de trame à voir la figure 28 suivante :

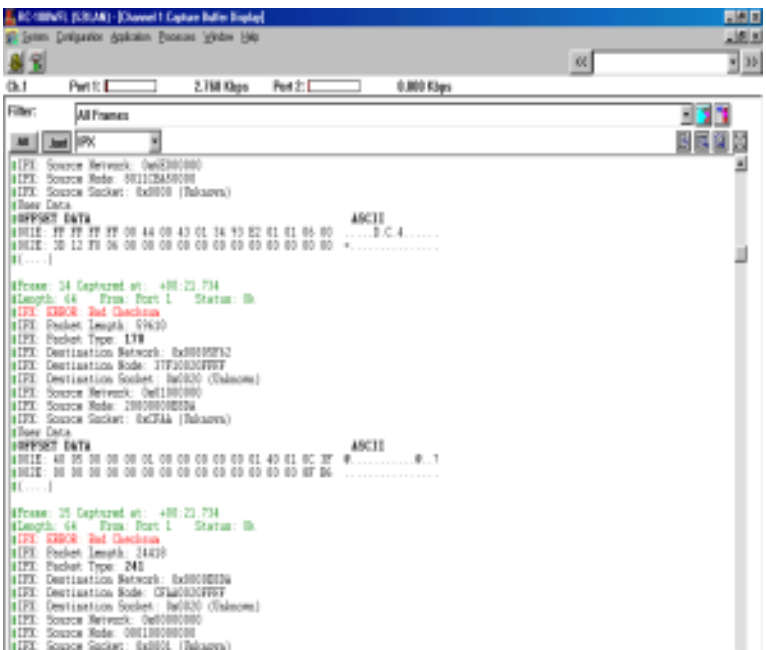


Figure 28 : Structure de trame

Parmis les points fort de ce produit (RadCom) sont les représentations graphiques selon les adresses (sources et destinations), les protocoles (TCP, UDP, IPX, etc..) et les ports (sources et destinations).

La figure 29 ci-dessous est une représentation suite à une analyse du réseau selon les protocoles échangés.

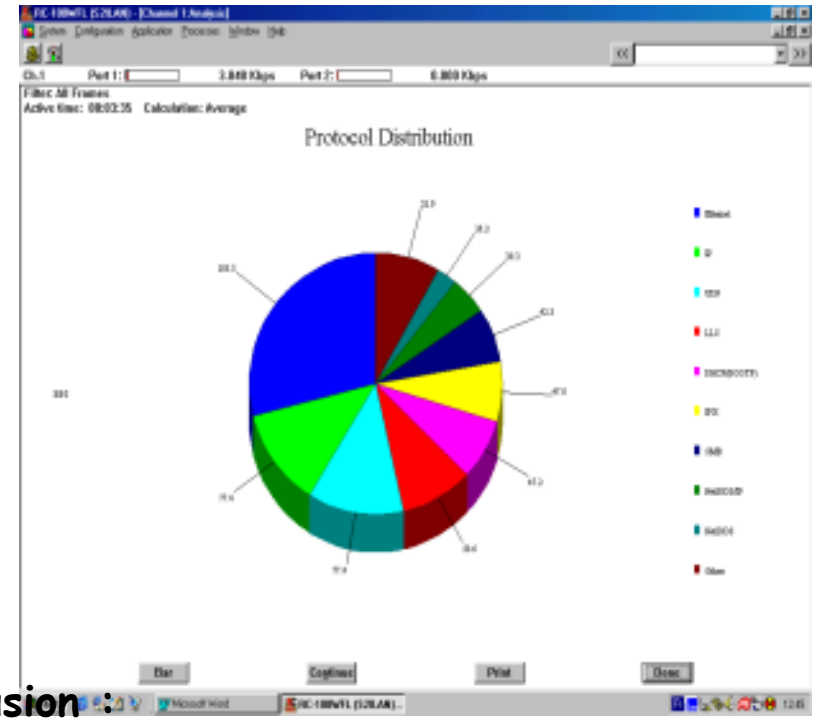


Figure 29 : Statistique graphique selon le protocole

IV. Conclusion :

Ce chapitre est assez important pour la réalisation de mon projet, j'ai bien constaté que chacun de ces analyseurs a des propres spécifications et d'autres qui sont standards. Puisque que je n'ai pas des vrai développeurs ou des professionnels, j'ai pris les principales fonctions que je dois l'intégrer au niveau de mon application comme par exemple la détection des adresses IP (sources et destinations), aussi le type de graphique (en secteur ou en bar) et encore d'autres.

CHAPITRE 4

Problématique

« Penser ne suffit pas; il faut penser à quelque chose »
Jules Renard

I. Introduction :

Lorsqu'un utilisateur envoie un paquet vers une autre machine, une connexion est établit et le paquet va se propager à travers tout le réseau (voir figure 30 ci dessous), les problèmes qui se posent sont :

- Comment je vais capturer ces paquets sur le réseau même s'ils ne sont pas destinés à mon machine ?
- Comment je vais choisir l'outil de développement ?
- Comment je vais implémenter les fonctions nécessaires au niveau d'outil choisit ?

Pour répondre à ces questions je dois comprendre d'abord la manière particulière dont fonctionnent les applications sur les réseaux en générale. Ces applications, lorsqu'elles doivent communiquer, utilisent un paramètre supplémentaire en plus de l'adresse IP de l'expéditeur et du destinataire : le port de l'application (généralement, chaque application travaille sur un port spécifique).

Ensuite, je dois choisir l'outil de développement de telle sorte qu'elle m'aide d'aboutir mes objectifs. Puis étudier la manière spécifique permettant l'implémentation des fonctions lors du développement.

Etablir une connexion entre deux postes n'est vrai que lorsqu'on assure une association des sockets serveur et client plus le type de transport (TCP/UDP). Ces derniers ne sont que des couples d'adresses IP et les numéros de ports (les identifications locales de communication).

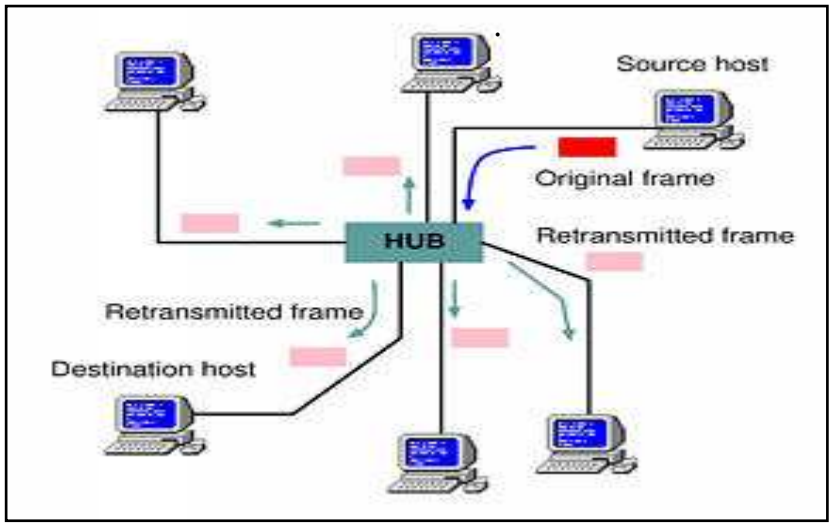


Figure 30 : Propagation des paquets sur un réseau Ethernet partagé

II. La technique de communication :

1. Les sockets :

a) Définition :

Lorsqu'un ordinateur transmet des données sur le réseau, les paquets circulant contiennent les informations suivantes :

- L'adresse IP du hôte destinataire.
- Le socket de l'application serveur.
- L'adresse IP de l'expéditeur.
- Le socket de l'application client.

Je peux modéliser une « socket » comme un téléphone, c'est à dire comme l'extrémité d'un canal de transmission bidirectionnel. En connectant deux ordinateurs au même canal de transmission, on peut faire passer des informations d'un programme à un autre sur deux ordinateurs différents.

b) Les Winsock :

« Winsock » est l'abréviation des Windows Sockets, passage obligé de la programmation réseaux, même si certains outils, comme certains composants pour les outils Borland, permettent de ne pas se préoccuper de cet aspect reboutant au premier abord.

Winsock est utilisé dans les applications Windows comme support pour la programmation orientée réseaux. C'est l'interface entre le « software » et le « hardware », c'est ce qui permet de programmer l'envoi et la réception de trames.

2. Le numéro de port :

a) La notion de ports d'écoute :

Une bonne approche des ports pourrait être de les comparer au fonctionnement des lignes téléphonique numériques : on sais que ces téléphones peuvent recevoir plusieurs appels sur un seul numéro (par exemple, téléphoner avec un ami, surfer sur Internet et recevoir un fax, le tout simultanément). On peut faire l'analogie avec le TCP/IP :

- Le numéro de téléphone correspond à l'adresse IP.
- Les différentes lignes de téléphone correspondent aux différents sockets

La comparaison s'arrête là, car cette ligne téléphonique peut recevoir simultanément au plus 8 lignes, autant que la connexion à Internet par exemple comporte plusieurs milliers de ports d'écoute.

D'une manière générale, on peut dire qu'une grande majorité d'applications serveurs tournent sur un port inférieur à 1024 (http : 80, ftp : 21, telnet : 23, pop3 : 110, etc.). Avec des exceptions toutefois, par exemple des serveurs FTP "pirates" (tournant sur des ports exotiques) ou certains serveurs web (port 8080 par exemple).

D'une manière générale, les applications clientes font leur requête sur un port supérieur à 1024.

b) Le concept de numéro de port :

Le protocole TCP et le protocole UDP utilisent des numéros de port pour transmettre de l'information aux couches supérieures. Les numéros de port servent à distinguer les différentes conversations qui circulent simultanément sur le réseau. (Voir figure 31)

Les développeurs d'applications ont convenu d'utiliser les numéros de port bien connus qui sont définis dans la requête pour commentaires 1700. Par exemple, toute conversation acheminée à une application du protocole FTP utilise le numéro de port standard 21. Les conversations auxquelles participent des applications qui n'ont pas de numéros de port définis se voient attribuer des numéros de port sélectionnés de manière aléatoire à l'intérieur d'une plage précise. Ces numéros de port sont utilisés comme adresses source et de destination dans le segment TCP.

Certains ports sont réservés au sein des protocoles TCP et UDP, bien que les applications ne soient pas nécessairement codées pour les supporter. Les plages attribuées aux numéros de port sont les suivantes :

- Les numéros inférieurs à 255 sont réservés aux applications publiques.
- Les numéros de 255 à 1023 sont attribués aux entreprises pour les applications à commercialiser.
- Les numéros supérieurs à 1023 ne sont pas attribués.

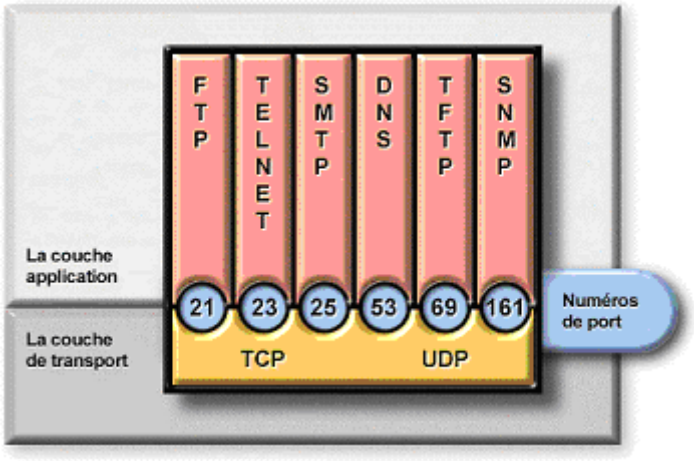


Figure 31 : Le numéro de port

Les systèmes d'extrémité se servent des numéros de port afin de sélectionner les applications appropriées. Les numéros de port source d'origine, généralement des numéros supérieurs à 1023, sont attribués de façon dynamique par l'hôte. (Voir figure 32)

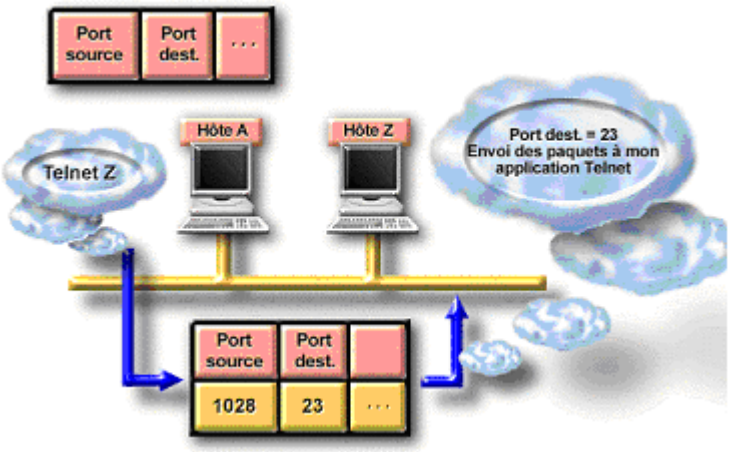


Figure 32 : Les Numéros de port TCP/UDP

c) Le diagramme d'une connexion TCP ouverte :

Les deux extrémités d'une connexion sont synchronisées au moyen d'une séquence de connexion ouverte comportant un échange en trois étapes, voir la figure 33 ci-dessous. L'échange de numéros de séquence d'ouverture au cours de la connexion permet d'assurer que les données perdues pourront être récupérées si des problèmes surviennent ultérieurement.

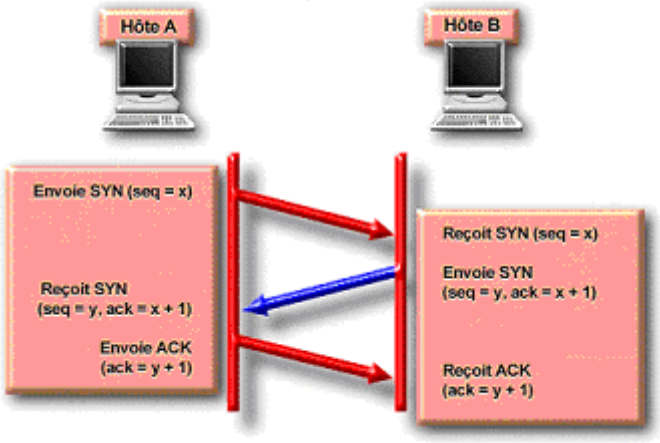


Figure 33 : Etablissement d'une connexion : Echange en 3 étapes TCP

d) Les numéros de séquence et d'accusé de réception TCP :

D'après le schéma de la figure 34, le protocole TCP assure l'ordonnancement des segments grâce à des accusés de réception vers l'avant. Chaque datagramme est numéroté avant la transmission. Au poste récepteur, le protocole TCP assemble le segment en message complet. Si un numéro de séquence est absent de la série, le segment correspondant est transmis de nouveau. Si l'accusé de réception pour un segment n'est pas reçu dans un délai déterminé, ce segment est transmis de nouveau.

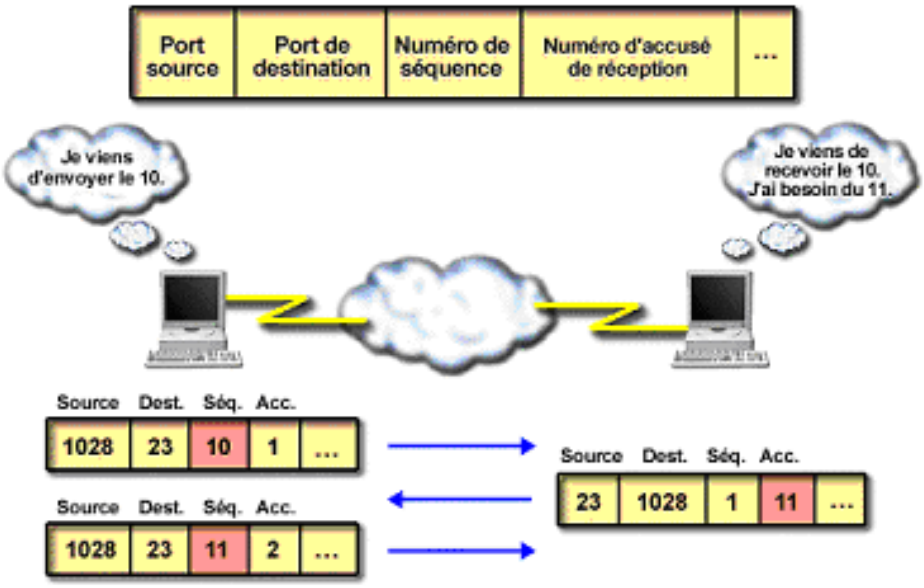


Figure 34 : Numéros de séquence et d'accusé de réception TCP

3. Conclusion :

Toutes ces notions m'aide à comprendre la technique de communication entre les ordinateurs au sein d'un réseau local, le travail à réaliser est donc d'implémenter les fonctions nécessaires lors de programmation selon le langage choisit et qui m'aide à atteindre mes objectifs.

III. L'outil de développement :

1. Le choix d'outil :

Outre la facilité d'utilisation et d'apprentissage, un outil doit répondre aux besoins de l'application à mettre en place. Il est donc primordial d'étudier les caractéristiques de l'application afin de choisir l'outil de développement :

- L'application est-elle client - serveur, mono - client ou de type Web ?
- Quelle est la plate-forme cible pour la partie serveur : Windows, Unix... ?
- Quelle est la plate-forme cible pour la partie cliente : Windows, navigateur HTML... ?
- L'application doit-elle être multilingue ?
- La conception sera-t-elle orientée objets ou composants ?
- Sur quel support seront stockées les données : fichiers, BD Oracle... ?

2. Introduction :

Microsoft's Visual Basic, communément appelé VB, est un atelier de génie logiciel (AGL) de création d'applications Microsoft Windows. La première version de visual basic sortie en 1991 avait pour vocation d'étendre le langage de programmation BASIC avec des

fonctionnalités graphiques et de fournir un environnement convivial de développement d'applications dédiées Windows basé sur ces extensions. Avec l'arrivée d'Internet, le visual basic a agrandi son registre en intégrant dans son offre des fonctionnalités Web avec par exemple un éditeur HTML.

Le visual basic est un outil en perpétuelle évolution. Ainsi, la version 6.0 (octobre 1998) commence à introduire des concepts objets dans le langage BASIC et fournit tout un ensemble de fonctionnalités Web. Le visual basic est aussi le premier à intégrer les nouveaux standards Microsoft, il est même le biais par lequel sont imposés ces nouveaux standards tels que ActiveX.

Plus d'autres informations sur le visual Basic se trouvent dans l'annexe B.

3. Caractéristiques générales :

a) Un éditeur graphique puissant et simple :

L'éditeur graphique de VB est WYSIWYG (what you see is what you get), ce qui signifie que la vision de l'interface donnée par l'éditeur lors de la conception est exactement celle obtenue lors de l'exécution. Ce concept facilite la mise au point de l'interface qui ne nécessite pas l'exécution du programme pour être visualisée. De plus, le processus de création d'une interface a été optimisé en utilisant, entre autres, les notions de glisser, déplacer ou de dimensionnement au moyen de la souris.

b) Un langage interprété :

Le langage BASIC est à la base un langage interprété. À la compilation du projet, du code appelé p-code est généré, il est ensuite traduit en code natif par la librairie dynamique MSVBVM60.dll au moment de l'exécution. C'est pourquoi il reste beaucoup moins performant que d'autres langages compilés tels que Visual C++ ou Turbo Pascal. Il convient donc parfaitement à la conception d'interface mais n'est pas approprié aux traitements lourds.

De plus, cette caractéristique empêche l'utilisation de traitements développés en VB par d'autres langages compilés, alors que l'inverse est possible. En effet, toute fonction C++ peut être utilisée dans VB par l'intermédiaire d'une interface d'appel écrite en BASIC. Cette technique permet notamment d'accéder à l'ensemble des bibliothèques systèmes de Windows.

c) Une programmation événementielle :

La conception d'une application VB sort un peu du cadre standard de programmation. En effet, un programme traditionnel repose sur une procédure principale qui appelle des traitements en chaîne afin de remplir une tâche donnée. Une fois la tâche achevée, le programme s'arrête. Le point de départ d'une application VB est généralement une fenêtre qui s'affiche à son lancement. Par la suite, des événements sont envoyés à la fenêtre par le système opératoire ou l'utilisateur via le clavier et la souris. Le travail de programmation

consiste alors à coder les traitements à exécuter en réponse à chacun de ces événements, le programme s'arrêtant lorsque la fenêtre principale de l'application est fermée. Ce mode de fonctionnement n'est pas propre à VB et se retrouve dans d'autres outils de développement d'applications graphiques et ce, indépendamment du système opératoire.

d) Un outil orienté composant :

Outre la possibilité de s'interfacer avec des traitements externes, VB est un outil très ouvert qui suit une approche composant via le concept de composants ActiveX. L'avantage de tels composants est la réutilisation de code dont le comportement est conditionné par des propriétés définies au niveau de ces composants. Les composants ActiveX sont facilement distribuables et ont l'avantage de se comporter comme de véritables boîtes noires en masquant entièrement leur implémentation. Cette technique est aussi le moyen par lequel VB peut partager et diffuser certains de ses traitements dans d'autres logiciels de développement.

e) Un outil de développement complet mais sans support de conception :

Le visual basic n'est pas un outil de conception, sa pauvreté actuelle au niveau des concepts objets rendant difficile toute transposition automatique entre un langage de conception tel qu'UML et le langage BASIC et malgré ça j'ai essayé de faire une partie d'analyse et conception pour la réalisation de mon projet.

En contrepartie, VB possède tout ce qu'il faut pour mettre en œuvre une application Windows complète. En effet, il intègre l'aspect donné, en fournissant notamment des fonctionnalités de gestion de base de donnée, d'aide en ligne et de génération de rapport.

f) Un outil 100 % Microsoft :

Le visual basic intègre l'ensemble des standards Windows. On retrouve ainsi les menus Fichier, Édition, Fenêtre ou l'aide, ainsi que les raccourcis claviers habituels (Ctrl-C pour copier ou Alt-F pour le menu Fichier). Cela ne rend que plus facile son apprentissage aux personnes déjà familières avec des applications telles que Microsoft Word ou Visual C++. VB offre aussi un ensemble de barres d'outils que l'utilisateur peut choisir d'afficher ou pas.

4. Implémentation de Winsock :

Pour installer le canal de transmission, je commence par prévenir la compagnie qui s'occupe des téléphones. Je commence donc par prévenir l'ordinateur. J'initialise « Winsock » en appelant la fonction **WSAStartup ()**, et en donnant la version que je vais utiliser.

De même, lorsque j'en ai fini avec les transmissions de données par les réseaux, j'appelle la fonction **WSACeanup ()** pour désinstaller proprement le canal de transmission.

Il s'agit ensuite d'installer le canal de transmission. Je fais appel à la fonction **socket ()** pour spécifier le type d'adresse que j'utilise (typiquement les adresses IP sur Internet : la structure **AF_INET** contenant une adresse IP et un numéro de port), ainsi que la façon dont je vais envoyer les données : soit sur TCP (Transfert Control Protocol, paramètre **SOCK_STREAM**), soit sur UDP (User Datagram Protocol, paramètre **SOCK_DGRAM**).

Il faut par la suite tester une adresse à écouter pour lier le canal de transmission à une adresse et un numéro de port. A cet instant, le canal est prêt à fonctionner, on peut envoyer et recevoir des données avec les fonctions **send** et **recv**. Juste à la fin de la connexion, je fais appel à la fonction **closesocket** qui termine cet acheminement des données.

5. Conclusion :

Le Visual Basic reste un outil de prototypage très efficace ; sa prise en main rapide ainsi que sa richesse en termes de composants de développement en font l'un des meilleurs de sa catégorie.

Ses extensions Web le mettent maintenant au premier plan pour le développement d'applications Internet et intranet. En effet, la convivialité et la simplicité de l'outil jumelées à son approche de composants via ActiveX poussent de plus en plus l'institut à le choisir pour le développement de leur système intranet dont ils maîtrisent les environnements clients.

IV. Conclusion :

Au cours de ce chapitre, j'ai essayé de répondre aux quelques questions posé dés le début, mais le travail le plus important sera la résolution de problème sur le plan pratique c'est à dire au niveau de programmation.

CHAPITRE 5

Analyse et conception

«Il faut aimer travailler en solitaire »
Jéromes

I-Introduction

Le sujet de mon projet est la réalisation d'un contrôleur de trafic pour analyser le trafic dont les caractéristiques ont été mentionnées dans le cahier des charges. Le but de ce chapitre est d'établir une première description du futur système. Ses données sont les résultats de l'analyse des besoins ainsi que des considérations techniques et de faisabilité informatique. Son résultat est une description de ce que doit faire le contrôleur de trafic en évitant des décisions prématurées de réalisation (on dit quoi, on ne dit pas comment).

Ce chapitre représente une première approche pour se mettre dans le cadre du sujet. Une description des majeures fonctionnalités du système, son utilité.

II- Conception

II-1 Introduction

Cette étape est délicate car elle consiste à détailler le comment faire pour assurer le bon déroulement des besoins annoncés par la spécification. De plus, tout le travail à effectuer par la suite repose sur cette phase une fois achevée…

Qu'est-ce qu'un prototype ?

Le prototypage est un concept très important dans le développement des logiciels de qualité. C'est une approche basée sur une vue évolutive du développement logiciel qui consiste en la création des versions préliminaires d'un logiciel ou partie pour validation. Il sert entre autre pour la communication entre utilisateurs et développeurs. Un prototype est tout simplement un modèle de l'application.

Pourquoi l'approche objet ?

⇒ Conception

Les avantages de l'approche objet sont la stabilité de la modélisation par rapport aux entités du monde réel, la construction itérative facilitée par le couplage faible entre composants et la possibilité d'utiliser des éléments d'un développement à un autre. L'approche objet repose à la fois sur la rationalisation d'une démarche cartésienne et sur une démarche systémique qui considère un système comme une totalité organisée, dont les éléments solitaires ne peuvent définis que les uns par rapport aux autres. Elle propose une méthode de décomposition, non pas basée uniquement sur ce que le système fait, mais plutôt sur l'intégration de ce que le système est et fait.

L'approche objet a pour but une modélisation des propriétés statiques et dynamiques de l'environnement dans lequel sont définis les besoins.
Les objets

L'objet est une entité atomique formée de l'union d'un état et d'un comportement. Il fournit une relation d'encapsulation qui assure à la fois une cohésion interne très forte et un faible couplage interne entre le dit objet et l'extérieur.

L'objet révèle son vrai rôle et sa vraie responsabilité lorsque, par l'intermédiaire de l'envoi d'un message, il s'insère dans un scénario de communication.

Pour atteindre mon besoin de projet, et les demandes de cahier de charge, j'ai choisis le langage de modélisation UML car il répond à la démarche de mon application qui est une solution technique, et UML traite ce type de solutions à contraire de Merise qui est orienté vers les solutions de base de données.

II-2 Définition d'UML

UML (Unified Modeling Langage) est un langage de modélisation objet et non pas une méthode. Le travail sur ce nouveau langage a continué et il a été adopté par les grands acteurs industriels. En novembre 1997 UML, a été adopté par l'OMG (Objet Management Group).
Les constituants d'UML sont les éléments de modélisation et les diagrammes. Les éléments de modélisation représentent toutes les propriétés du langage. Ces éléments ne se limitent pas au symbolisme graphique utilisé.
UML définit neuf diagrammes pour représenter les différents points de vue de modélisation.
Les diagrammes d'activité : représentation du comportement en termes d'action.
Les diagrammes de cas d'utilisation : représentation des fonctions du système du point de vue de l'utilisateur.
Les diagrammes de classe : représentation de la structure statique en termes de classes et de relations.
Les diagrammes de collaboration : représentation spatiale des objets, des liens et des interactions.
Les diagrammes de déploiement : représentation du déploiement des composants sur les dispositifs matériels.
Les diagrammes d'états transitions : représentation du comportement d'une classe en terme d'états.
Les diagrammes d'objets : représentation des objets et de leurs relations.
Les diagrammes de séquence : représentation temporelle des objets et de leurs interactions.
Toutefois, pour éviter de surcharger le rapport et d'entrer dans les détails techniques, je ne présenterais que quelques diagrammes qui sont utiles pour comprendre le projet à savoir le diagramme des cas d'utilisation et le diagramme de séquence.
Maintenant que j'ai vu la partie 'théorique', je peux comprendre avec plus de facilitée comment fonctionne un contrôleur de trafic, et l'intérêt qu'un tel outil présente, autant dans le but de surveiller son réseau, que d'attaquer un réseau.

II-3 Démarche de conception

II-3-a Modèle des cas d'utilisation

L'approche consiste à regarder le système à construire de l'extérieur, du point de vue de l'utilisateur et des fonctionnalités qu'il en attend. Les cas d'utilisation sont par conséquent très utiles en phase d'analyse des besoins.

Le but de ce modèle est de présenter les différents besoins et facettes de l'application d'une façon formelle. Je vais dans un premier temps représenter les différents acteurs du système. Les acteurs de l'application se répartissent comme suit :

L'administrateur : c'est l'acteur principal, son rôle réside dans la configuration d'un analyseur de trafic sur le réseau.

La Carte réseau : représente l'intermédiaire entre le réseau et l'application.

Le Réseau : c'est l'acteur qui agit sur l'application en générant les paquets.

Système : celui qui exécute les différentes taches de scénario.

Compte tenu de ces acteurs je vais maintenant approfondir mon analyse davantage pour dégager les différents cas d'utilisation. Je vais donner par la suite le diagramme des cas d'utilisation de tout le système ainsi que son diagramme de séquence.

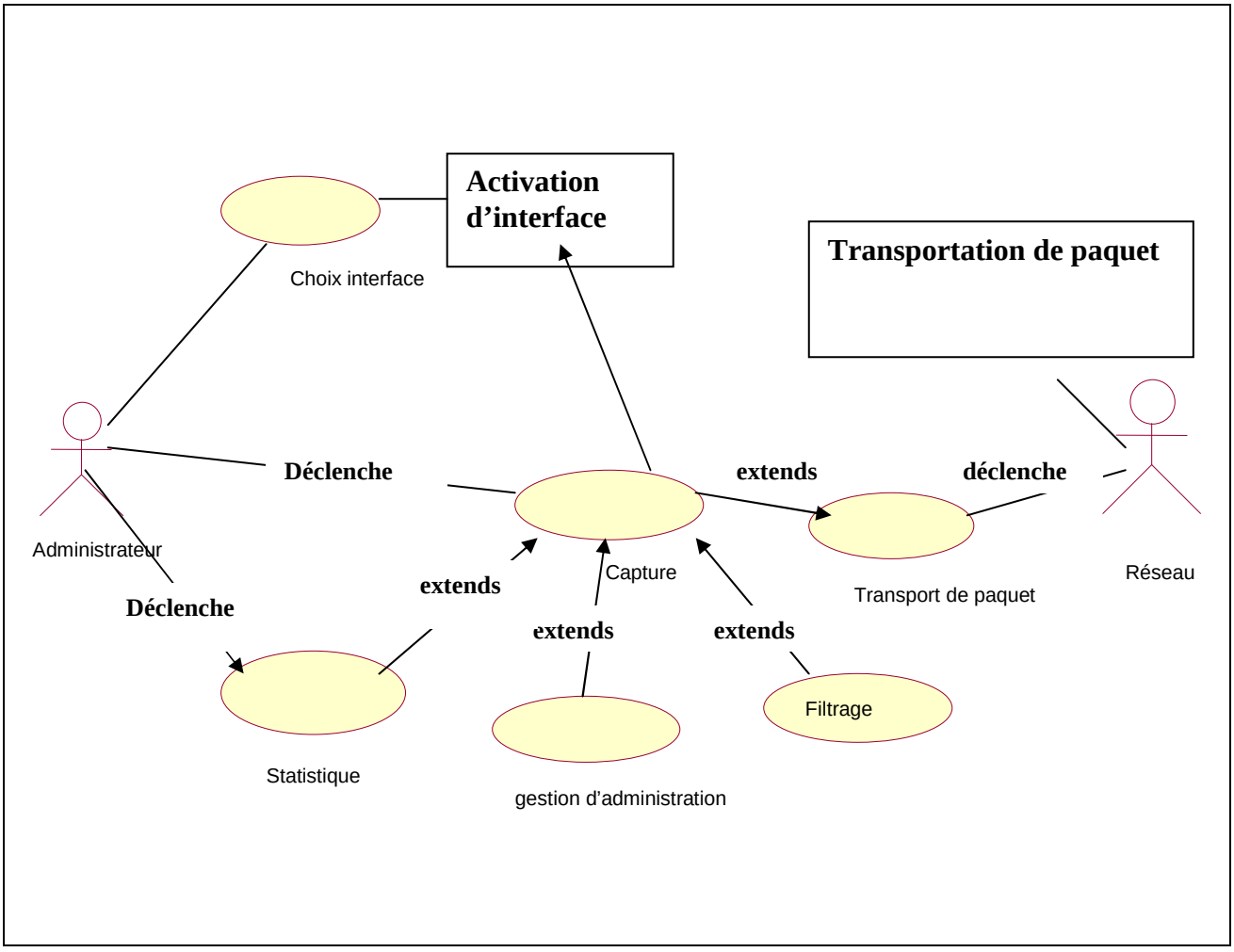


Figure 4.1 : Diagramme de cas d'utilisation

EXPLICATION

L'administrateur choisit l'interface de capture (s'il en existe plusieurs) et lance la capture. Si le réseau génère des paquets, ils seront captés par la carte réseau, traités par le programme. On pourra appliquer le filtrage selon des règles précises, visualiser la partie statistique qui fonctionne en temps réel, et faire un ping sur le réseau.

II-3-b Diagramme de Séquences

Le diagramme de séquence représente les interactions entre les objets dans un enchaînement temporel. Il montre les objets et les classes impliqués dans un scénario, ainsi que la succession des messages échangés entre les objets pour réaliser la fonctionnalité du scénario.

Configuration du système

L'utilisation du système est en fonction des besoins de l'utilisateur et de l'orientation adoptée. Une configuration est donc nécessaire. L'utilisateur doit tout d'abord choisir ces paramètres globaux (capter le trafic par interprétation de leurs propres informations, filtrer les trafics, consulter les statistiques, et faire un ping sur le réseau).

Voici maintenant les différentes possibilités de séquence des actions

Supposons qu'on est à zéro risque d'échec de mon application comment le contrôleur de trafic agit il ?

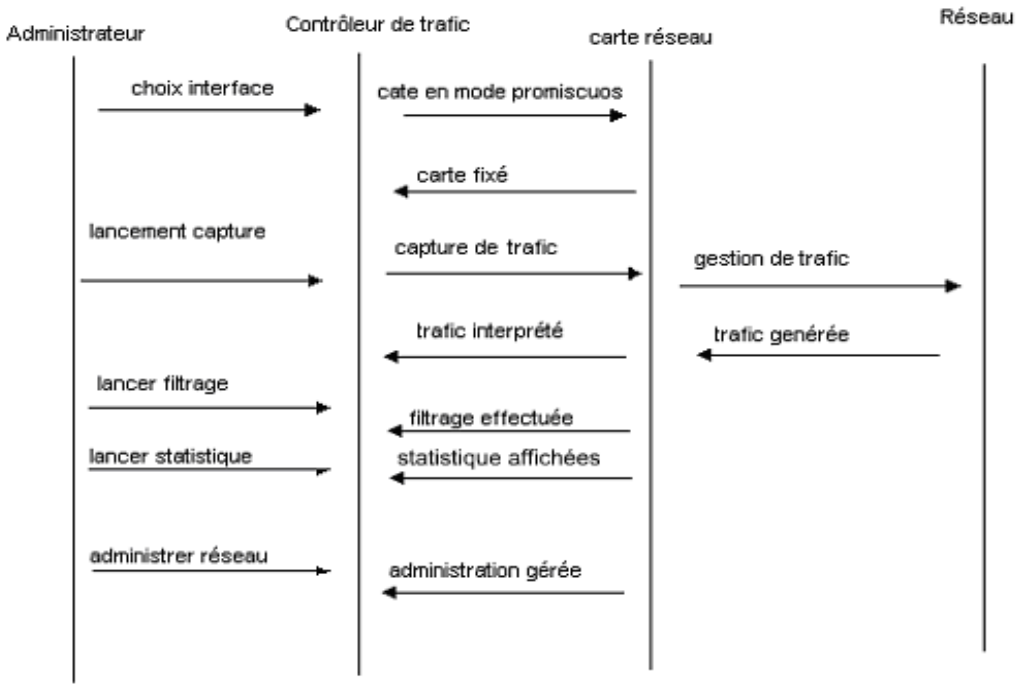


Figure 4.2: Diagramme de séquence

Le réseau est mon deuxième acteur qui représente la raison pour laquelle on a développé mon application. Mon application affiche un résultat si le réseau génère des paquets

II-3-c Diagramme de déploiement

Les diagrammes de déploiement montrent la disposition physique des différents matériels (les nœuds) qui entrent dans la composition d'un système et la répartition des programmes exécutables sur les matériels.

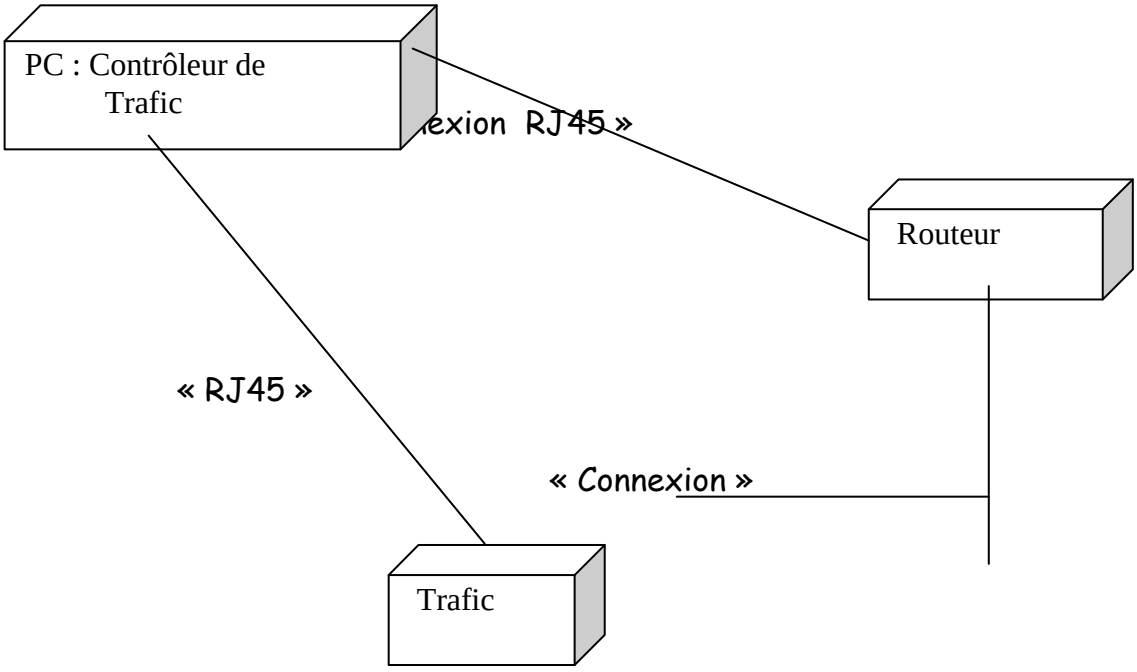


Figure 4.3 : Diagramme de déploiement

CHAPITRE 6

Réalisation

« Tout ce qu'un homme peut imaginer, un jour d'autres hommes le réalisent »
Jules Verne

I. Introduction :

Mon projet est comme déjà prédéfinie ne sort pas de l'environnement de la programmation réseau en Visual Basic. Comme tous les outils de développements, le VB a des avantages et des inconvénients. En plus qu'il est un langage orienté réseau, il est aussi orienté interface mais pour programmer un long programme (au niveau de code) et qui possède plusieurs fonctionnalités, quelques inconvénients appariaient comme :

- Le temps d'exécution qui est assez long (surtout lorsque le programme doit accéder à une base de donnée).
- La modification du code n'est pas facile (pour l'ajout ou la suppression de quelques fonctions ou procédures)

C'est pour ces raisons que j'ai passé de la programmation classique à une programmation modulée d'où la décomposition du programme (principal) a développé en sous programmes, de cette manière j'assure un travail plus :

- Organisé
- Rapide lors d'exécution
- Facile à modifier

La programmation modulée exige que l'application ou le programme principal doive être divisé en modules qui seront appelé par la suite. Sous VB, un sous-programme est considéré comme un projet.

II. Création du projet :

Au cours de ce chapitre, j'ai introduit mon travail juste après la présentation des principales étapes de création d'un projet VB sont les suivantes :

1) Création de l'interface : feuilles, contrôles, propriétés.
2) Codage : événements, procédures.
3) Débogage et test.
4) Création d'un exécutable.

La programmation des feuilles et modules est la même puisqu'ils ont en commun les interfaces et les codes.

1. Interface :

Pour créer l'interface d'une application, il est nécessaire de construire des feuilles, d'y ajouter des contrôles et de modifier leurs propriétés.

a) Création des feuilles :

Une feuille est l'élément principal de création d'une interface, elle contient les contrôles et deviendra une fenêtre Windows une fois compilée. Une application peut contenir une ou plusieurs fenêtres, chaque fenêtre pouvant à son tour contenir des fenêtres filles. Ce concept s'appelle MDI **(Multi Document Interface)**. Microsoft Word est l'exemple type d'une application MDI car il est possible d'ouvrir plusieurs documents en même temps.

b) Ajout des contrôles :

Une fois la feuille créée, il faut assembler les contrôles nécessaires à l'accomplissement des actions de l'application. Les contrôles sont sélectionnés dans la boîte à outils et placés sur la feuille.

c) Affectation des propriétés :

Une fois les contrôles positionnés, les propriétés de la feuille et des contrôles peuvent être modifiées, à commencer par leurs noms. VB recommande un ensemble de règles de nommage pour chaque contrôle et feuille.

d) Ajout des menus :

Une interface utilisateur nécessite souvent l'introduction de menus. L'ajout d'un menu personnalisé à une feuille se fait au moyen de l'option du menu principal. Les barres de menu ainsi créées prennent en charge les notions de sous-menus, de raccourcis clavier ou encore d'options désactivées.

2. Écriture du code :

La programmation VB répartit le code entre deux types de modules : les feuilles et les modules standards.

Les feuilles se composent de :

- Un ensemble d'événements à coder, relatifs à la fenêtre et à ses contrôles.
- Déclarations de variables.
- Procédures nécessaires à l'implantation des événements de la feuille.

Les modules standards contiennent les traitements applications et s'articulent autour de :

- Déclarations de variables.
- Fonctions.
- Instructions (Sub).

a) Code des feuilles (Événements) :

Le code écrit dans les procédures d'événements détermine le comportement de l'application en réponse à des actions d'utilisateurs ou à des messages provenant du système, d'autres applications ou de l'application elle-même.

Le nom associé à une procédure d'événement d'un contrôle combine le nom effectif du contrôle (spécifié dans la propriété Name), un trait de soulignement (_) et le nom de l'événement. La liste des événements disponibles pour le contrôle sélectionné est visible dans la zone des procédures/événements de la fenêtre de code.

b) Code des modules (Procédures externes) :

VB fournit un moyen de créer des procédures externes qui peuvent être appelées depuis n'importe quel événement. Les procédures définies par l'utilisateur ne sont donc pas déclenchées par un événement spécifique du système mais par d'autres procédures dans l'application. Contrairement aux événements, aucune procédure ne possède un nom prédéfini.

Sans aucun doute que cette partie est la plus importante au niveau de programmation (au moins dans le cadre de mon projet), c'est pour cette raison qu'on profite de cette partie pour présenter quelques codes assez important dans mon travail.

3. Débogage du code et procédure de test :

Il existe un ensemble d'outils d'aide à la mise au point d'une application. Les techniques de débogage de VB prennent en charge les points d'arrêt, les expressions d'arrêt, les expressions espionnes, l'exécution du code instruction par instruction ou procédure par procédure, ainsi que l'affichage des valeurs des variables et des propriétés. VB comporte aussi des options spéciales de débogage, notamment la possibilité de modifier la valeur de variables en cours d'exécution, de redéfinir la prochaine instruction à exécuter ou encore de vérifier des procédures alors que l'application est en mode arrêt.

a) Entrer en mode débogage :

Pour pouvoir entrer dans le mode de débogage pendant l'exécution de l'application, un point d'arrêt doit être défini sur un ensemble de lignes de code. Un point d'arrêt demande à VB d'interrompre l'exécution de l'application au niveau de la ligne de code concernée : l'application est alors en mode arrêt.

b) Examiner l'application :

Lorsqu'un point d'arrêt est atteint et que l'application est interrompue, un suivi pas à pas de l'exécution de l'application peut être mis en œuvre. Une flèche dans la marge de la fenêtre code et une ligne colorée indiquent la prochaine ligne de code à exécuter. Les options de menu Pas à pas détaillé ou Pas à pas principal permettent d'avancer dans le code ligne après ligne.

En mode arrêt, il est également possible d'examiner la valeur de variables ou de propriétés en utilisant la fenêtre Variables locales, la boîte de dialogue Espion express, les expressions espionnes ou la fenêtre Exécution.

4. Création d'un exécutable :

La dernière étape d'un projet concerne la génération d'un exécutable juste après la compilation du code.

III. Développement :

Mon travail consiste en premier lieu de créer les interfaces pour les sous programmes ainsi que le programme principal. Ensuite, c'est la partie du code que j'ai attaqué, j'ai implémenté quelques parties du code juste après chaque interface.

J'ai pratiqué le même principe sur tous les sous programmes au niveau d'interface. En plus des menus, chaque projet possède une barre d'outils qui résume ses propres fonctionnalités (voir les figures 35, 36, 37, 38, 39 et 40 ci-dessous).

Le programme principal (voir figure 35) est : **HC_ScanNetwork** où il fait appel aux sous programmes qui sont :

Grâce au guide d'utilisation, il est très simple d'utiliser mon l'application (voir annexe C).
Le programme principal : **HC_ScanNetwork**

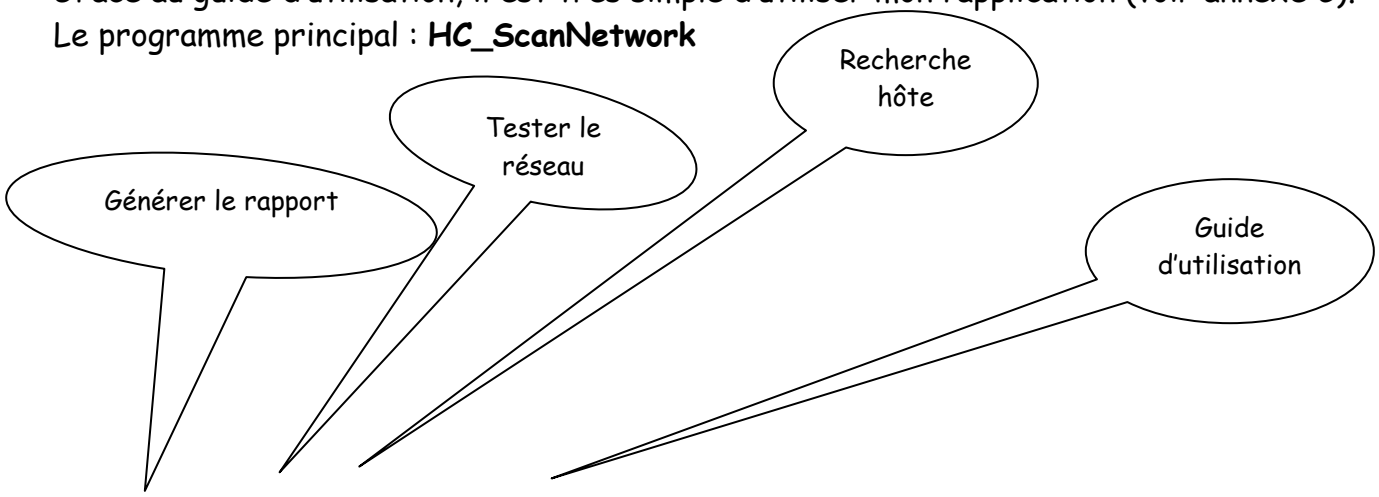


Figure 35 : Fenêtre principal de HC_ScanNetwork

HC_ScanNetwork offre en plus de ces outils :

- Un guide d'utilisation : Pour simplifier et guider l'utilisation de l'application.
- Info système : Pour plus savoir sur le système utilisé sur une machine locale.

Les interfaces crées pour le programme principal et les sous programme sont :

1. **HC_ScanNetwork Capture** : Assure la détection des paquets échangés sur un réseau local, leurs adresses IP sources et destinations, les numéros de ports sources et destinations et le nombre de bit transmis (voir figure 36).

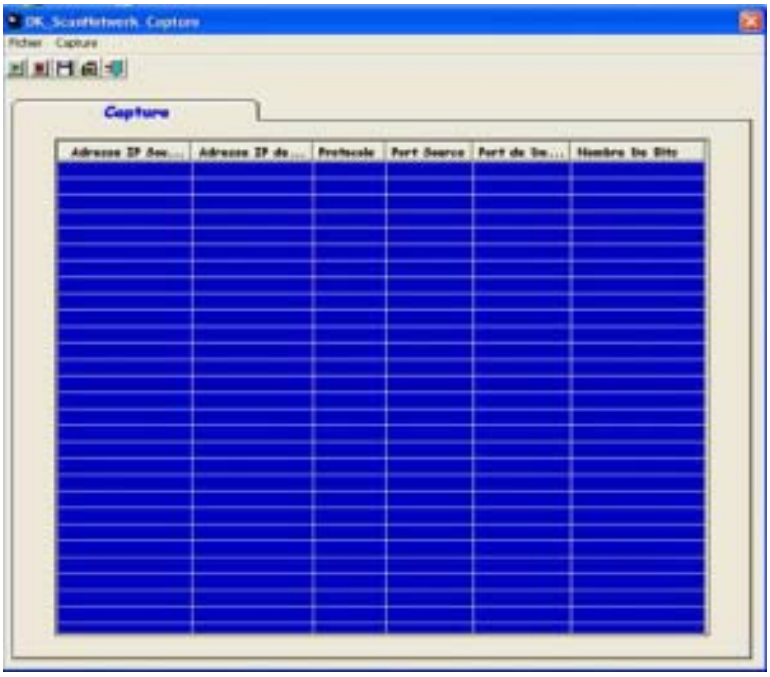


Figure 36 : Fenêtre de HC_ScanNetwork Capture

Code pour démarrer socket:

```
Public Function StartWinsock(sDescription As String) As Boolean
   Dim StartupData As WSADataType
   If Not WSAStartedUp Then
      If Not WSAStartup(&H101, StartupData) Then
         WSAStartedUp = True
         sDescription = StartupData.szDescription
      Else
         WSAStartedUp = False

End If
End If  StartWinsock = WSAStartedUp
End Function
```

Code pour arrêter socket:

```
Sub EndWinsock()
   Dim ret&
   If WSAIsBlocking() Then
      ret = WSACancelBlockingCall()
   End If
   ret = WSACleanup()
   WSAStartedUp = False
End Sub
```

2. **HC_ScanNetwork Graphique :** Assure une représentation en temps réel des paquets selon différents types de graphe : Aire, Barre, Combinaison, Secteur et Escalier (voir figure 37).

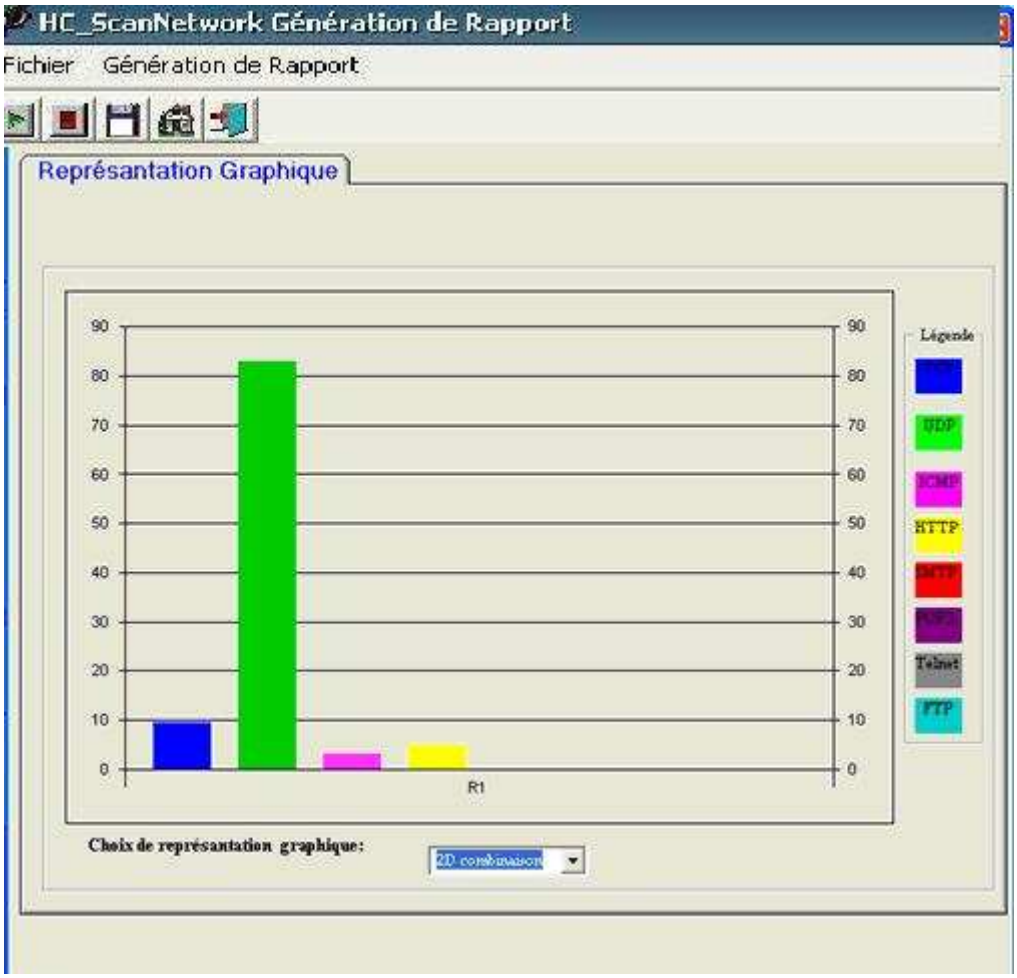


Figure 37 : Fenêtre de HC_ScanNetwork Graphique

Code concernant la statistique temporelle (timer stat) :

```
total = stcp + sicmp + sudp + http + smtp + pop3 + telnet + ftp
If stcp <> 0 Or sicmp <> o Or sudp <> 0 Then
arrPrices(1) = (stcp / total) * 100
arrPrices(2) = (sudp / total) * 100
arrPrices(3) = (sicmp / total) * 100
arrPrices(4) = (http / total) * 100
arrPrices(5) = (smtp / total) * 100
arrPrices(6) = (pop3 / total) * 100
arrPrices(7) = (telnet / total) * 100
arrPrices(8) = (ftp / total) * 100
MSChart1.ChartData = arrPrices
```

3. **HC_ScanNetwork Test réseau :** Assure les tests des hôtes en réseau grâce aux ping, trace route et informations hôte (voir figure 38).

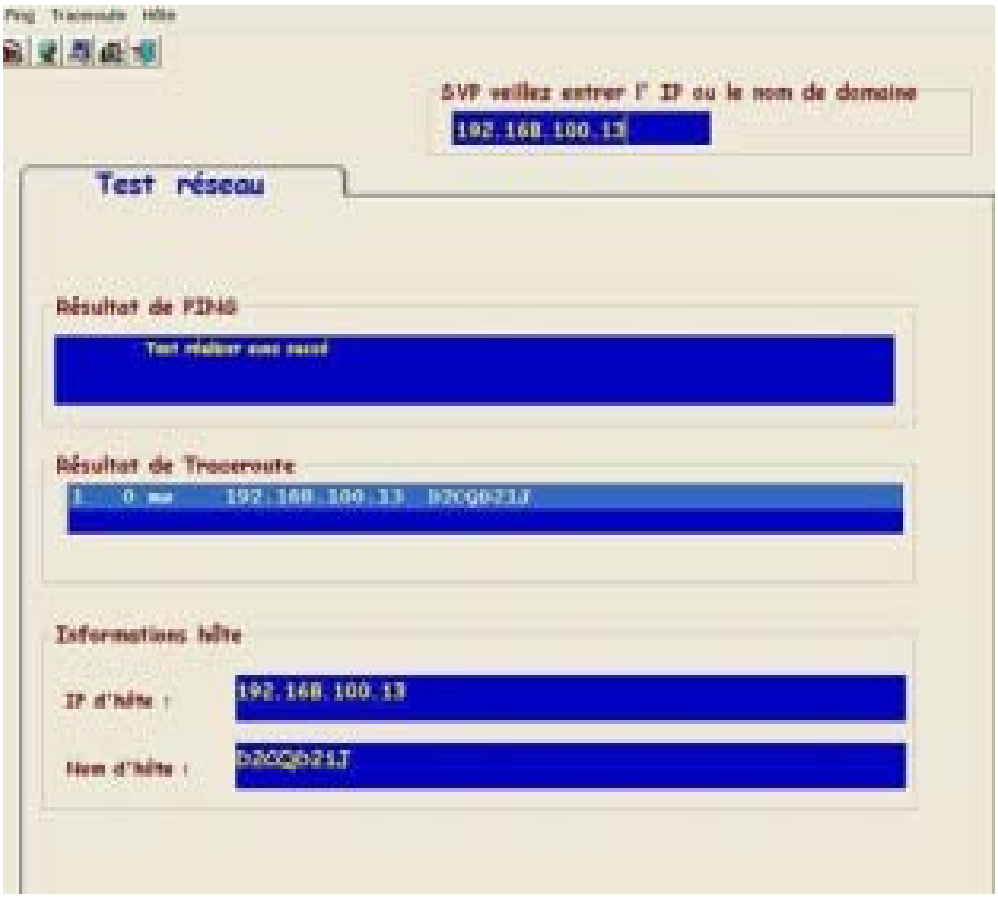


Figure 38 : Fenêtre de HC_ScanNetwork Test réseau

Code de ping:

```
Select Case Tp
      Case -2
         EasyPing = "Err"
      Case -1
         EasyPing = "    Test non valide"
      Case Else
         EasyPing = " Test réaliser avec succé   "
   End Select
```

4. **HC_ScanNetwork Génération de rapport :** Assure la génération de rapport, contenant les informations suivantes : adresses IP sources et destinations, numéros de ports sources et destinations, le protocole utilisé et le nombre de bit (voir figure 39).

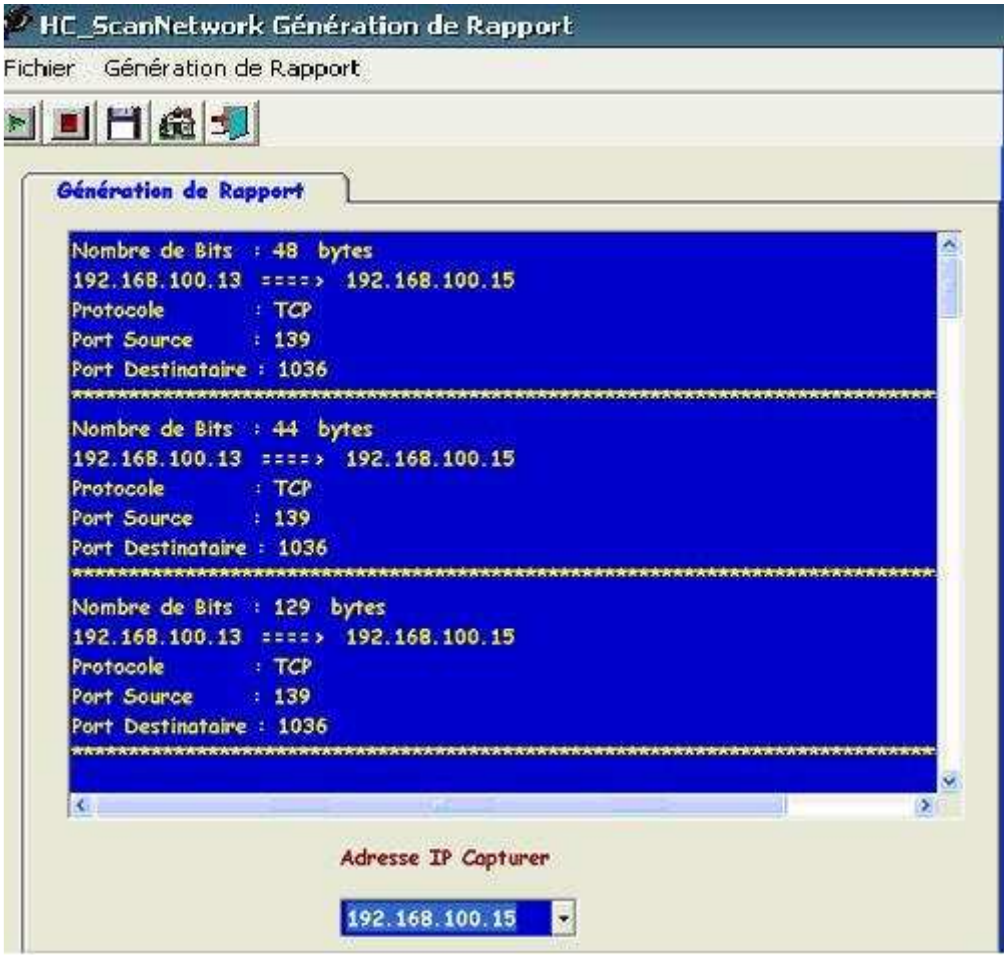


Figure 39 : Fenêtre de HC_ScanNetwork Génération de rapport

Code de capture pour le protocole TCP:

```
If ip_header.ip_protocol = 6 Then
Obj.Text = Obj.Text & "Protocole  : TCP" & vbCrLf
      CopyMemory_any tcp_header, ReadBuffer(0 + 20), Len(tcp_header)
Obj.Text = Obj.Text & "Port Source : " & ntohs(tcp_header.src_portno) & vbCrLf
Obj.Text = Obj.Text & "Port Destinataire : " & ntohs(tcp_header.dst_portno) & vbCrLf
End If
```

5. **HC_ScanNetwork Recherche hôte :** Assure la détection de toutes les hôtes présentent dans une plage d'adresse que l'on définie dés le début ainsi que les informations suivantes : adresse IP, état et le nom d'hôte (voir figure 40).

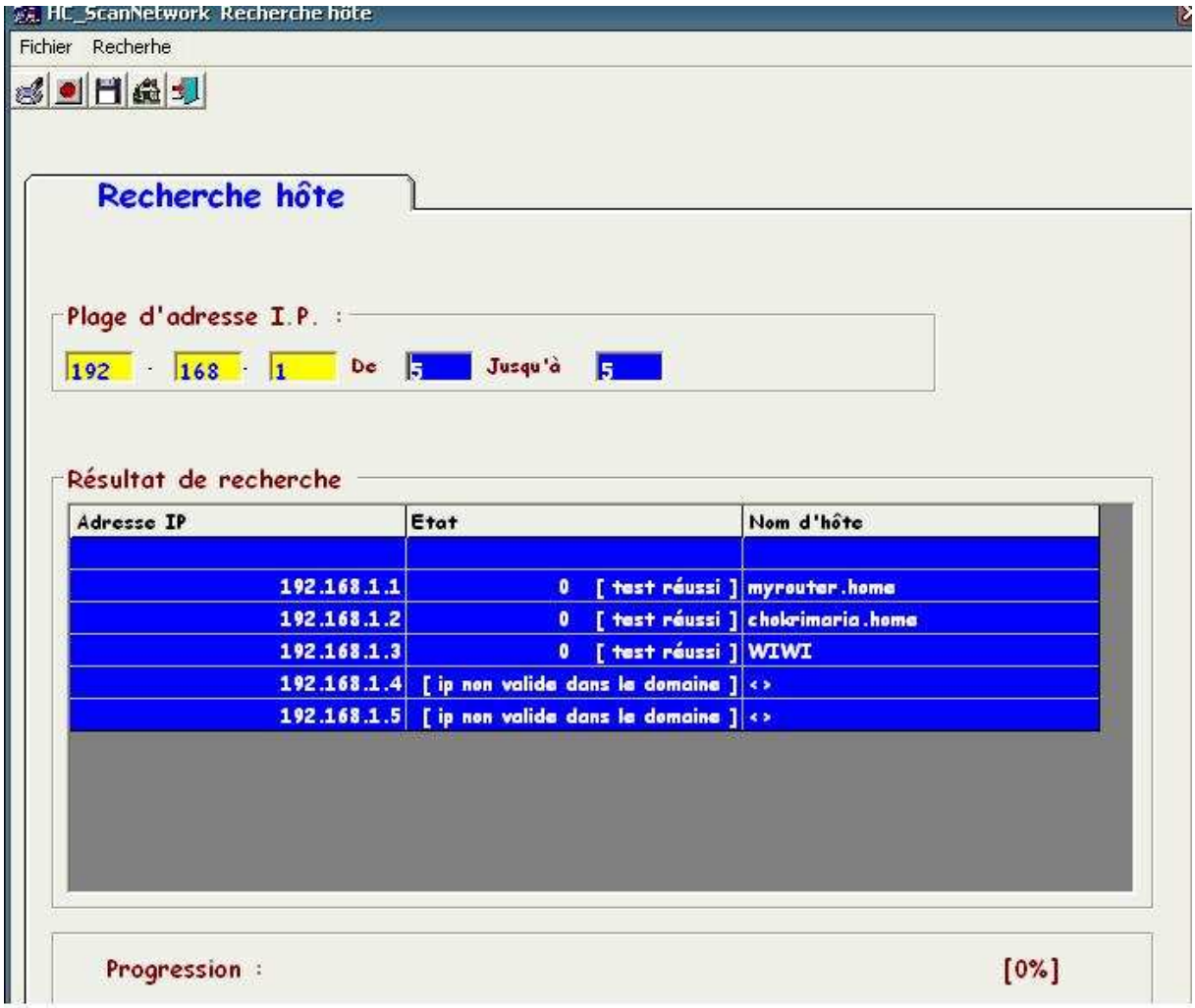


Figure 40 : Fenêtre de HC_ScanNetwork Recherche hôte

Code de la barre de progression :

```
For i = TxtIP(3).Text To TxtIP(4).Text
        TxtIP(3).Text = i
        tmpPos = (TxtIP(3).Text)
        tmpRes = tmpPos / Percent
         If tmpRes <= 1 Then
            Me.PrgPercent = (tmpRes) * 100
            Me.percentPos.Caption = "[" & Int(tmpRes * 100) & "%]"
            Me.percentPos.Refresh
        End If
```

IV. Test de fonctionnement :

Cette partie est consacré pour le test de mon projet, puisque la plateforme sur laquelle on travaille comporte que deux ordinateur, j'ai testé l'application sur un segment isolé du réseau d'où les résultats des figures ci-dessous.

1. **HC_ScanNetwork Capture :** La figure ci-dessous (voir figure 41) montre le fonctionnement de HC_ScanNetwork Capture, comme il est montré, il y a échange de données d'où la capture des adresses IP sources et destinations, les ports sources et destinations, le type de protocole des deux machines conservés ainsi que le nombre de bit transmis.

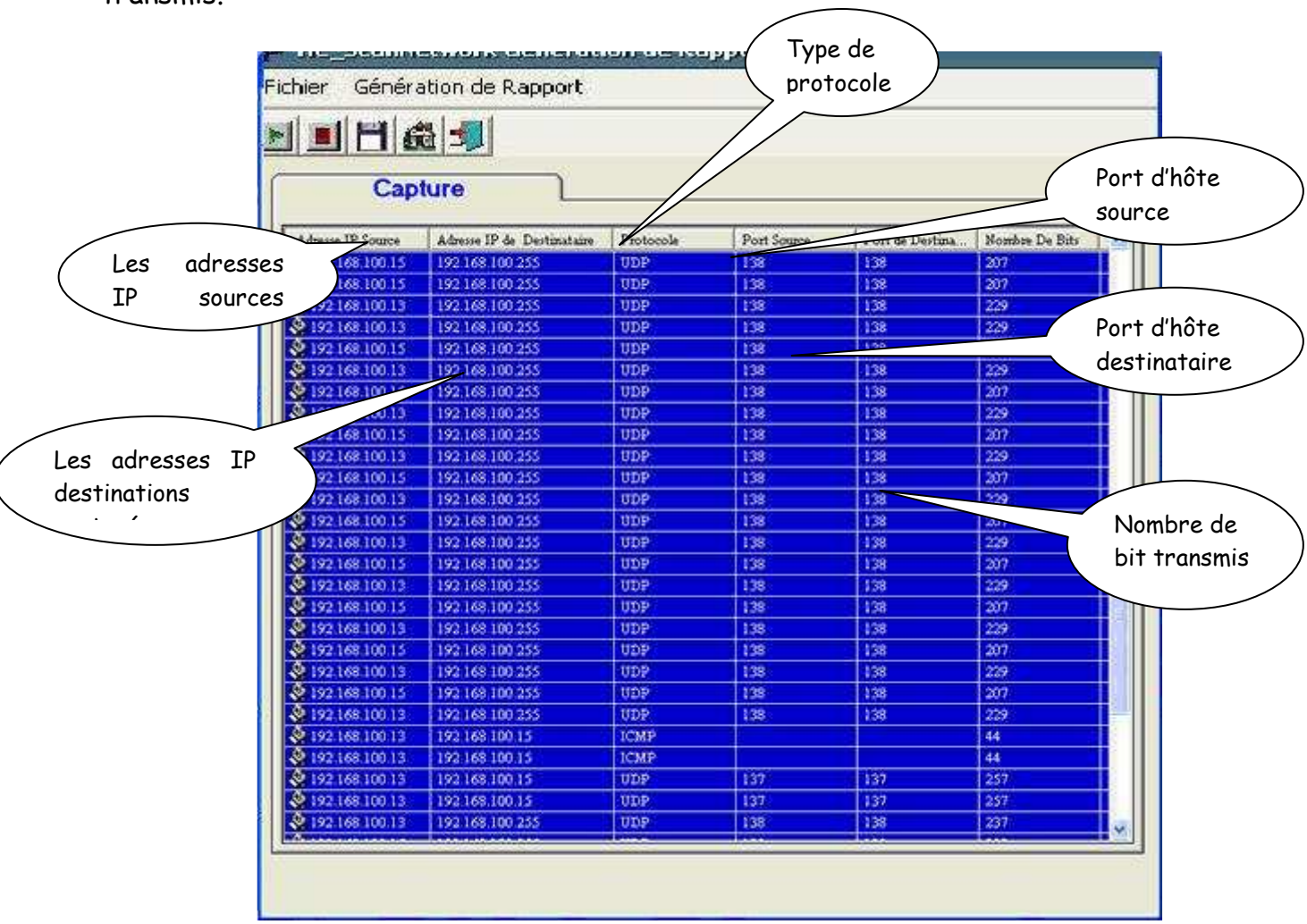


Figure 41 : Fenêtre de HC_ScanNetwork Capture

2. **HC_ScanNetwork Graphique :** Les figures ci-dessous (voir les figures 42, 43, 44) présente la destribution du trafic selon le type de protocole. HC_ScanNetwork Graphique me permet grâce à sa liste de représentation de choisir le type de graphe.

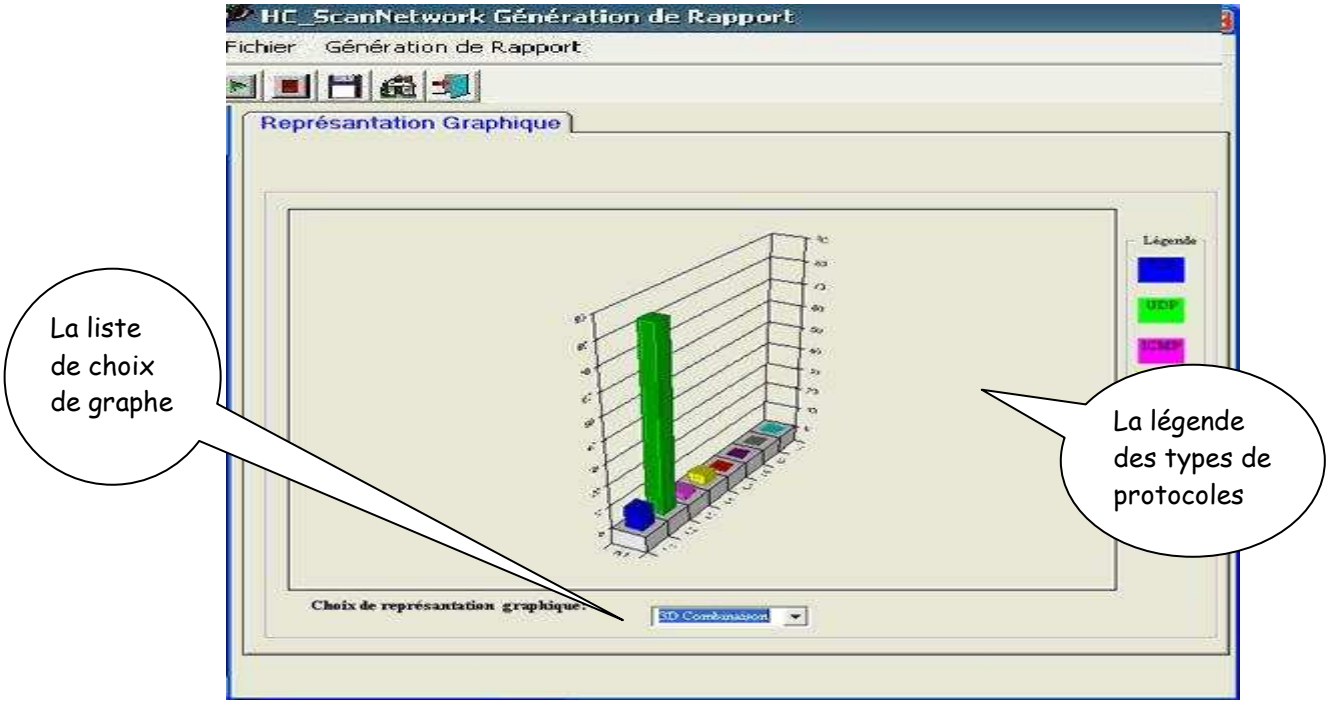


Figure 42 : Fenêtre de HC_ScanNetwork Graphique (type de graphe 3D Combinaison)

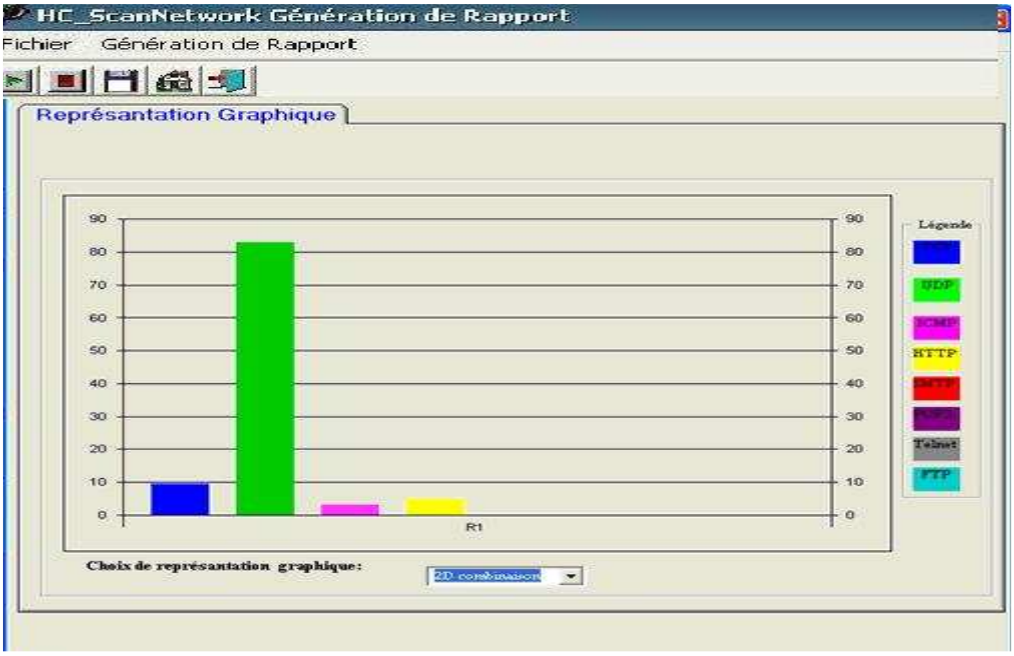


Figure 43 : Fenêtre de HC_ScanNetwork Graphique (type de graphe 2D Combinaison)

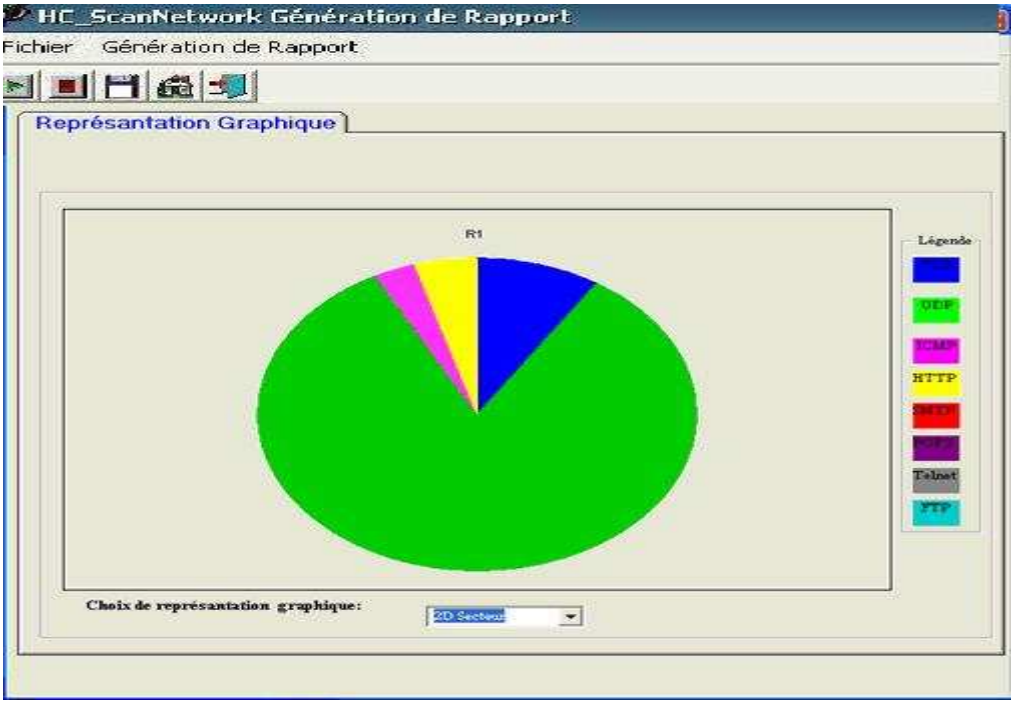


Figure 44 : Fenêtre de HC_ScanNetwork Graphique (type de graphe 2D Secteur)

3. **HC_ScanNetwork Test réseau :** Selon la figure 45, n'importe quelle machine appartenant au réseau local peut être identifié grâce au utilitaire de ping et trace route.

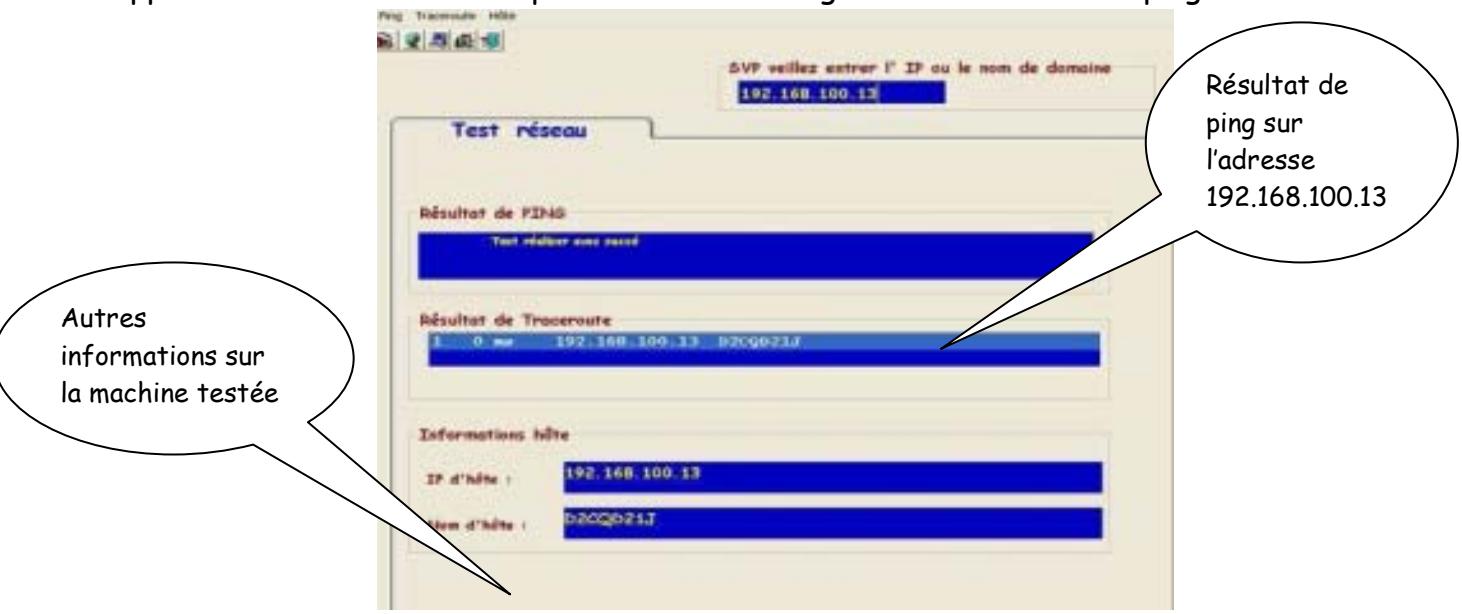


Figure 45 : Fenêtre de HC_ScanNetwork Test réseau

4. **HC_ScanNetwork Génération de rapport :** Tout paquet capturé sera mis en journal, un ensemble d'informations est affiché selon la figure 46 comme les adresses IP et les ports sources et destinations, etc..

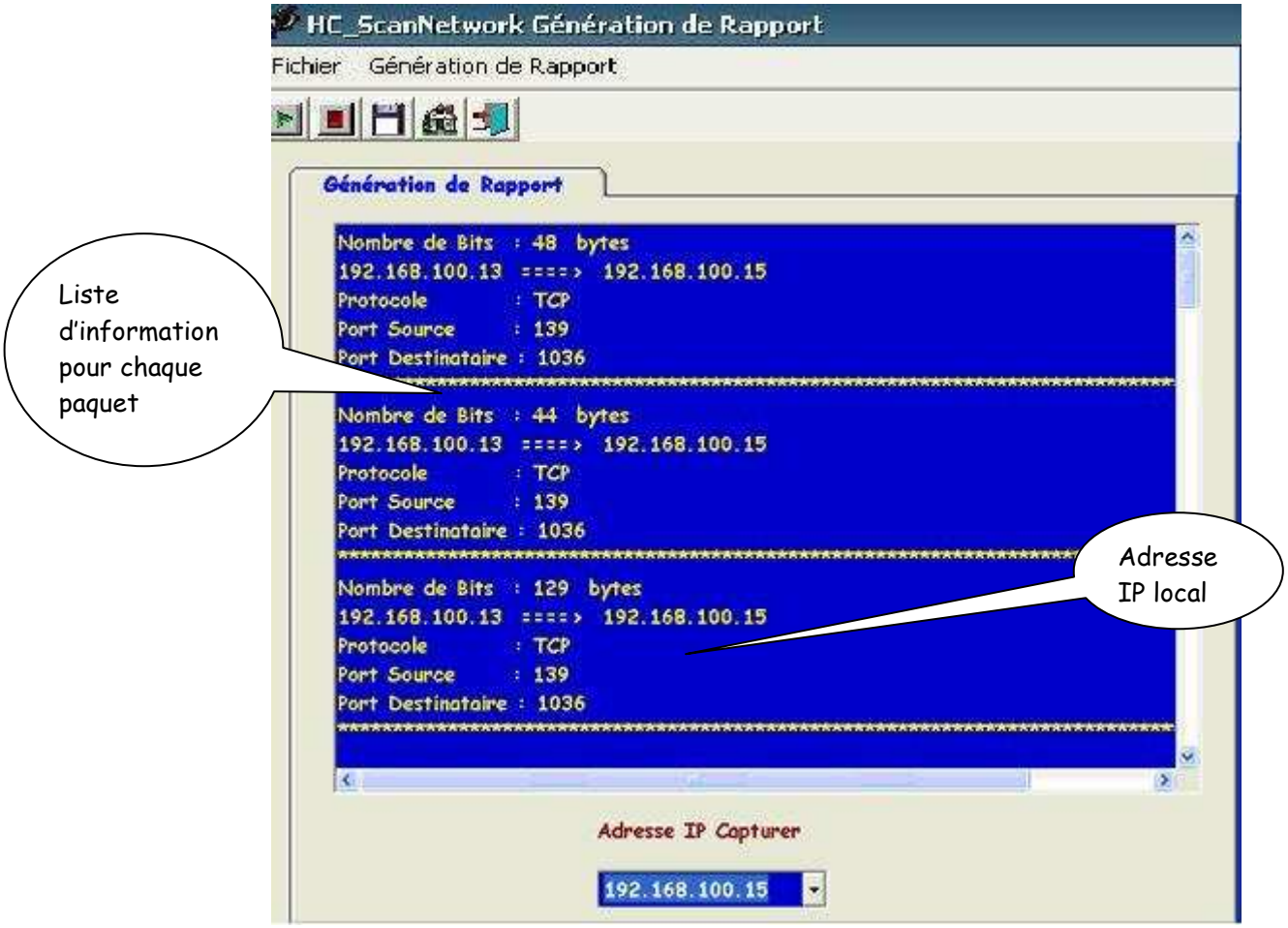


Figure 46 : Fenêtre de HC_ScanNetwork Génération de rapport

5. **HC_ScanNetwork Recherche hôte :** N'importe quelle machine appartenant à la plage d'adresse indiquée sera affichée ainsi que son adresse IP et son nom (voir figure 47).

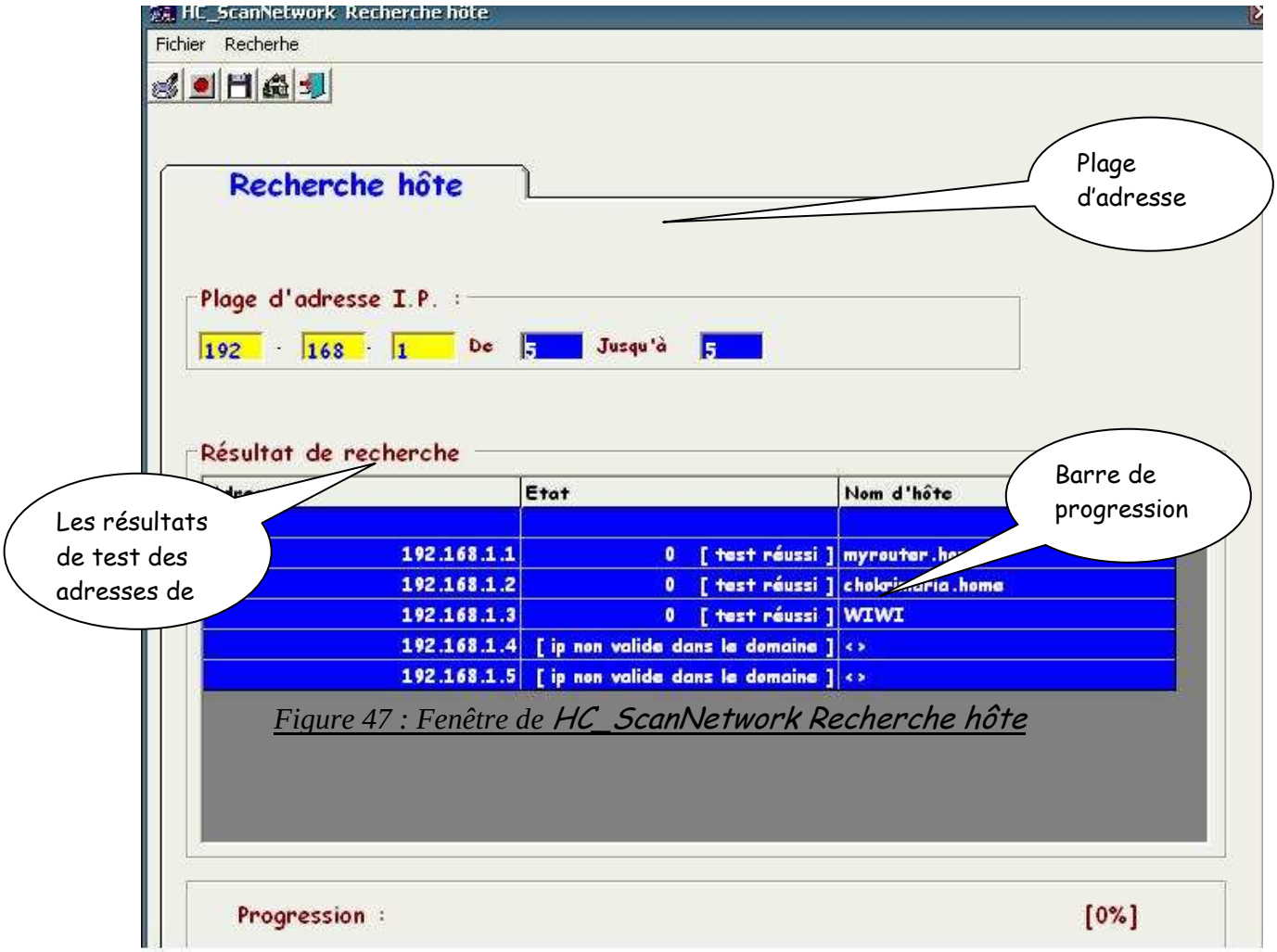


Figure 47 : Fenêtre de HC_ScanNetwork Recherche hôte

V. Dernier mot :

Lors de l'exécution du projet HC_ScanNetwork, une petite fenêtre va apparaître (voir figure 48) pour une durée de 5 secondes. En effet, cette fenêtre représente le logo de HC_ScanNetwork.

Figure 48 : Logo de HC_ScanNetwork

Vue critique

Tout au long de mon projet de fin d'étude, j'ai essayé de développer un outil d'administration (contrôleur de trafic) qui répond d'abord au contexte du cahier des charges puis aux besoins d'un administrateur et qui l'aide de surveiller le trafic réseau.

Malgré que j'aie pu atteindre les objectifs pour lesquelles j'ai réalisé ce projet, plusieurs inconvénients apparaissent. Ce projet ne permet pas :

- Le filtrage des paquets ni au niveau de protocole ou d'adresse.
- La représentation graphique selon les adresses (sources et destination) ou les ports (sources et destinations).
- L'affichage de la structure des paquets.

Ainsi l'inconvénient majeur de mon application est qu'il fonctionne comme un sniffer passif (Analyseur réseau) c'est-à-dire qu'il ne peut fonctionner qu'au niveau d'un réseau local non commuté. L'utilisation d'un switch (commutateur) rend l'application inefficace. D'autres moyens et techniques sont alors nécessaires pour contourner cette difficultée et permettre la capture des paquets circulant sur un réseau commuté.

Grâce à ses fonctions actuelles, HC_ScanNetwork représente comme même un noyau d'un contrôleur de trafic qu'il peut être amélioré et subir d'autres modifications pour le rendre plus performant.

Puisque le parfait n'existe jamais, je crois toujours à cette parole « Que ton intérêt porte sur l'action seulement, jamais sur les résultats »

Conclusion générale

Au cours de ce stage de projet de fin d'études j'ai pûs améliorer et acquérir des connaissances en terme de programmation réseau, chose importante dans le cadre de notre formation en informatique.

L'objectif escompté de ce projet de fin d'étude est de réaliser un outil d'administration (contrôleur de trafic) permettant à un administrateur la surveillance du trafic réseau pour un plateforme Windows.

Partant des fonctionnalités offertes par les API Windows, j'ai pu réaliser mon outil d'administration réseau qui offre à l'utilisateur (administrateur) une interface ergonomique et lui permet de bien contrôler les paquets circulants sur le réseau. Des fonctionnalités de capture, de statistique, test réseau, génération de rapport et recherche hôte sont offertes à travers des menus simples et concis.

Malgré quelques difficultés rencontrées au début, j'ai considéré que ce stage de projet de fin d'études a été très bénéfique et espérions que mon projet portera satisfaction aux utilisateurs.

ANNEXES

Annexe A

Comparaison de la pile de protocoles TCP/IP au modèle OSI :

La pile de protocoles TCP/IP correspond presque parfaitement au modèle de référence OSI pour les couches inférieures. Elle supporte tous les protocoles standard de couche physique et de liaison de données (voir la figure 16). L'information TCP/IP est transmise sous forme de séquences de datagrammes. Un message peut être transmis comme une série de datagrammes qui sont ensuite rassemblés à la destination pour reconstruire le message.

Le rôle de la pile de protocoles TCP/IP est de transférer de l'information d'un dispositif réseau à un autre. De ce fait, elle est étroitement liée au modèle de référence OSI, pour les couches inférieures, et supporte tous les protocoles physiques et liaison de données standard (voir la figure 1).

Les couches les plus étroitement liées aux protocoles TCP/IP sont les couches 7 (application), 4 (transport) et 3 (réseau). Ces couches comprennent d'autres types de protocole ayant des fonctions et des objectifs divers qui sont tous liés au transfert d'information. Certains de ces groupes de protocoles, ainsi que leur fonction, sont déjà listés.

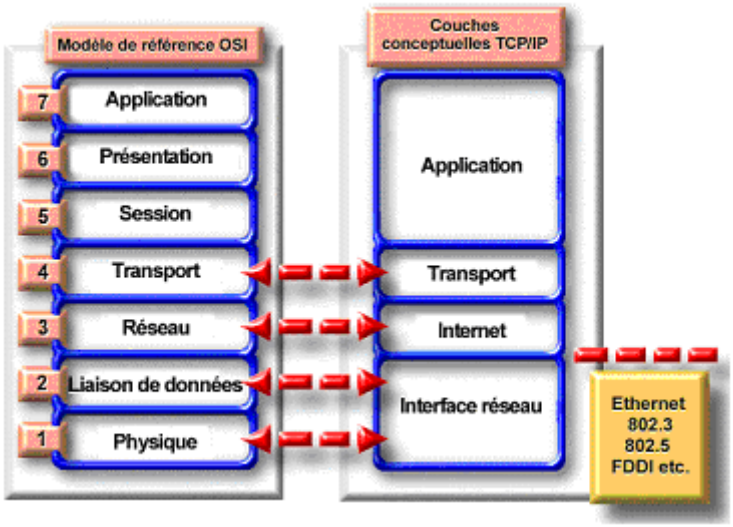


Figure 1 : Comparaison des modèles TCP/IP et OSI

En étudiant les deux modèles OSI et TCP/IP, on remarque d'une part quelques points de similitude entre eux :

- Tous deux comportent des couches.
- Tous deux comportent une couche application, bien que chacune fournisse des services très différents.
- Tous deux comportent des couches réseau et transport comparables.
- Tous deux supposent la technologie de commutation de paquets (et non de commutation de circuits).

D'autre part, c'est la différence qui sépare ou d'autre terme caractérise chaque modèle, on peut citer quelques points :

- TCP/IP intègre la couche de présentation et la couche session dans sa couche application.
- TCP/IP regroupe les couches physiques et liaison de données OSI en une seule couche.
- TCP/IP semble plus simple, car il comporte moins de couches.
- Les protocoles TCP/IP constituent la norme sur laquelle s'articule Internet, aussi le modèle.
- TCP/IP a-t-il acquis sa crédibilité en raison de ses protocoles. Par contraste, aucun réseau ne s'articule sur des protocoles particuliers au modèle OSI, même si tous se servent du modèle OSI comme guide.

Bien que les protocoles TCP/IP constituent les normes sur lesquelles repose Internet, dans le cadre du présent programme, le modèle OSI a été choisi pour les raisons suivantes :

- Il s'agit d'une norme universelle, générique et non tributaire des protocoles.
- Il comporte plus de détails, ce qui le rend plus utile pour l'enseignement et l'apprentissage.
- Il comporte plus de détails qui peuvent s'avérer fort précieux au moment du dépannage.

On doit familiariser avec deux modèles OSI et TCP/IP, le premier on l'utilise comme microscope pour l'analyse des réseaux et on utilise également le deuxième dans le cadre du programme d'études.

Annexe B

I. Le Visual Basic et les autres AGL (Atelier de Génie Logiciel) :

Dans le domaine des ateliers de génie logiciel (AGL), Visual Basic possède de sérieux concurrents sur le plan international, parmi lesquels Delphi de Inprise et PowerBuilder de PowerSoft. Bien que VB et Delphi soient de loin les plus complets, chaque outil possède des domaines d'excellence et des lacunes, VB n'échappant pas à cette règle.

Il arrive fréquemment qu'aucun outil ne corresponde exactement aux besoins de l'application. C'est pour cela que l'on trouve souvent des solutions hybrides qui mélangent plusieurs outils de développement à travers les différentes couches applicatives, à savoir données, traitement et présentation.

1. Environnement :

Le point fort de VB est sa facilitée d'utilisation due à son ergonomie qui utilise toutes les techniques possibles d'aide au développement (raccourci clavier, vérification syntaxique lors de la saisie du code, etc.).

Son langage de programmation (BASIC) est par ailleurs connu par un très grand nombre de développeurs du fait de sa simplicité, de son ancienneté et de sa présence dans la suite bureautique de Microsoft (Word, Excel et Access). Cette caractéristique distingue le visual basic d'autres outils qui, eux, nécessitent l'apprentissage d'un nouveau langage tel que le PowerScript (PowerBuilder), le NCL (Natstar) ou le W-Language (WinDev).

VB intègre parfaitement les dernières nouveautés Windows (notamment les technologies ActiveX) mais il supporte mal, voire pas du tout, les technologies non Microsoft bien souvent concurrentes. Ainsi, un développement CORBA n'est pas possible en VB ; Delphi est dans ce cas l'outil le mieux adapté.

La grande majorité des revendeurs de logiciels et de matériels offre des interfaces d'appel dédiées à VB. Cette caractéristique facilite grandement le développement et permet de créer des applications à forte valeur ajoutée. Ainsi, l'ajout de fonctionnalités de reconnaissance d'empreintes digitales ou de capture vidéo ne nécessite pas de programmation particulière.

Enfin, VB propose de nombreuses boîtes à outils qui facilitent grandement le développement, en particulier des applications de gestion. Ainsi, un paquetage de fonctions

financières permet d'effectuer des calculs d'amortissement ou de mensualités de crédit qui s'avèrent très utiles dans le développement d'applications bancaires.

2. Couche données :

VB s'interface avec pratiquement tous les types de bases de données, notamment à travers son support ODBC, tout en optimisant les accès à MS Access ou SQL Serveur. L'accès aux données est orienté « table » ou « requête » grâce aux composants RecordSet, ce qui oblige les développeurs VB à connaître parfaitement la structure des données.

3. Couche traitement :

VB possède de fortes limitations du fait qu'il n'offre pas la notion de pointeur où qu'il ne permet pas le développement multitâche, à l'opposé de son concurrent direct Delphi. Des études ont aussi montré que les traitements lourds en calcul ou les manipulations d'objets ont des performances bien que d'autres outils basés sur des langages tels que C++ (Visual C++) ou Turbo Pascal (Delphi). Pour combler ce manque, VB est souvent jumelé avec Visual C++.

De plus, la conception orientée objet est mal adaptée à VB qui préfère l'approche composant ou module, notamment à travers la technologie ActiveX. Ceci est en partie dû au fait que le BASIC n'est pas un véritable langage objet. En effet, sans l'héritage ou le polymorphisme, un modèle objet est très difficilement adaptable à VB, au contraire de Visual C++ ou Delphi qui permettent, au moyen d'outils tels que Rational Rose, de lier de manière automatique la conception objet à l'implantation.

Enfin, VB n'est pas un outil multi plate-forme, les exécutables générés ne fonctionnant que sur la plate-forme Windows ou le navigateur Internet Explorer (ActiveX). L'offre PowerBuilder ou Natstar est donc plus complète à ce niveau car elle permet de générer à partir du même code source un exécutable pouvant fonctionner sur un grand nombre de plates-formes, parmi lesquelles Unix.

4. Couche présentation :

La couche présentation reste la couche de prédilection de l'outil. Le développement des interfaces est orienté composants (ActiveX). Même si cette philosophie est présente dans la grande majorité des concurrents de VB, il n'en reste pas moins vrai que le nombre de composants ActiveX que l'on trouve sur le marché ou sur Internet est impressionnant et ne trouve pas d'égal.

II. Approche composants :

Le concept de contrôles ActiveX fait de VB un outil de premier choix pour toute application orientée composants. En effet, la facilité de mise en œuvre, d'empaquetage et de distribution des contrôles ActiveX permet une réutilisation optimale des fonctionnalités implantées par le composant ainsi qu'une commercialisation sûre et efficace de ce dernier.

En effet, les composants développés en VB protègent le distributeur de tout risque de piratage car aucun code source n'est livré, toute interaction avec le composant s'effectuant à travers ses paramètres et ses événements.

Avec la possibilité d'utiliser certains contrôles ActiveX avec le navigateur Internet Explorer, VB facilite la mise en œuvre de solutions intranets où l'on possède une parfaite maîtrise du navigateur installé sur chacune des machines clientes. En règle générale, VB s'intègre dans tout type de projet s'appuyant sur une architecture Web (intranet, Internet) mais l'intégration est d'autant plus facile que le projet repose sur des technologies purement Microsoft (serveur Internet Microsoft et navigateur Internet Explorer).

Si l'approche composante n'est pas un objectif en soi, VB n'en reste pas moins un bon outil d'implémentation d'interfaces graphiques qu'il faut adapter aux méthodes mises en place dans l'entreprise. Ainsi, sur une architecture 3-Tiers (séparation des couches données, traitement et présentation), VB se situe au niveau de la couche présentation et on utilise plutôt Visual C++ pour la couche traitement. Pour des développements objets sur VB, des règles d'implantation des concepts d'héritage et de polymorphisme doivent être définis. De plus, si le développement objets est jumelé à une base de donnée relationnelle, il est nécessaire de définir des règles de transposition entre objet et table de base de donnée.

Annexe C

Guide d'utilisation

Le HC_ScanNetwork vous offre quelques outils d'administration pour votre réseau.

Vous trouvez les outils les plus fréquents pour la surveillance d'un réseau local comme la capture des paquets, la représentation graphique selon les protocoles, identification des hôtes trouvant sur le réseau et la génération de rapport.

Ce guide vous aide à utiliser ce petit sniffer malgré sa simplicité.

I. Capture :

Pour lancer une capture, vous devez choisir dans la menu principale : **Contrôleur de trafic => Capture** ou en cliquant sur l'icône de la capture trouvant dans la barre d'outil.

Une fenêtre va apparaître dont son nom est HC_ScanNetwork Capture, pour lancer la capture soit vous cliquez sur l'icône de lancement trouvant à la barre d'outil ou suivre ce chemin : **Capture => Capturer**.

Pour arrêter la capture, soit vous cliquez sur l'icône d'arrêt ou suivre ce chemin : **Capture => Arrêter la capture**.

Vous pouvez aussi sauvegarder votre capture en cliquant sur l'icône d'enregistrement ou suivre ce chemin : **Fichier => Enregistrer**.

Vous pouvez retourner au programme principal par juste une petite clique sur l'icône (home) de la barre d'outils.

II. Statistique :

Vous devez choisir quel type de statistique vous souhaitez l'avoir : représentation graphique des paquets selon le protocole ou un rapport.

1. Représentation graphique :

Pour lancer une représentation graphique, vous devez choisir dans la menu principale : **Contrôleur de trafic => Statistique => Représentation graphique** ou en cliquant sur l'icône du graphique trouvant dans la barre d'outil.

Une fenêtre va apparaître dont son nom est HC_ScanNetwork Graphique, pour lancer la capture graphique soit vous cliquez sur l'icône de lancement trouvant à la barre d'outil ou suivre ce chemin : **Visualisation graphique => Commencer le suivi**.

Pour arrêter la capture graphique, soit vous cliquez sur l'icône d'arrêt ou suivre ce chemin : **Visualisation graphique => Arrêter le suivi**.

Vous pouvez aussi sauvegarder votre capture en cliquant sur l'icône d'enregistrement ou suivre ce chemin : **Fichier => Enregistrer**.

Vous pouvez retourner au programme principal par juste une petite clique sur l'icône (home) de la barre d'outil.

2. Génération de rapport :

Pour lancer le rapport des captures, vous devez choisir dans la menu principale : **Contrôleur de trafic => Statistique => Génération de rapport** ou en cliquant sur l'icône du rapport trouvant dans la barre d'outil.

Une fenêtre va apparaître dont son nom est HC_ScanNetwork Génération de rapport, pour lancer la capture soit vous cliquez sur l'icône de lancement trouvant à la barre d'outil ou suivre ce chemin : **Génération de rapport => Commencer**.

Pour arrêter la capture, soit vous cliquez sur l'icône d'arrêt ou suivre ce chemin : **Génération de rapport => Arrêter**.

Vous pouvez aussi sauvegarder votre capture en cliquant sur l'icône d'enregistrement ou suivre ce chemin : **Fichier => Enregistrer**.

Vous pouvez retourner au programme principal par juste une petite clique sur l'icône (home) de la barre d'outil.

III. Outils d'administration :

HC_ScanNetwork offre en plus de la capture des paquets quelques outils nécessaires pour l'administration d'un réseau comme le test réseau et recherche hôte.

1. Test réseau :

Pour tester s'il est en réseau ou non, vous devez lancer le test réseau, vous devez choisir dans la menu principale : **Auto découverte => Test réseau** ou en cliquant sur l'icône du test trouvant dans la barre d'outil.

Une fenêtre va apparaître dont son nom est HC_ScanNetwork Test réseau, vous devez entrer une adresse IP, ensuite, lancez le test soit vous cliquez sur l'icône de lancement trouvant à la barre d'outil ou suivre ce chemin : **Ping => Pinguer**.

Pour voir la trace route du réseau, vous devez lancer le test réseau, vous devez choisir dans la menu : **Trace route => Lancer** ou en cliquant sur l'icône du trace route trouvant dans la barre d'outil.

Pour avoir plus d'information sur la machine portant l'adresse IP déjà entré, vous devez choisir dans la menu : **Hôte => Lancer** ou en cliquant sur l'icône du trace route trouvant dans la barre d'outil.

Vous pouvez retourner au programme principal par juste une petite clique sur l'icône (home) de la barre d'outil.

2. Recherche hôte :

Pour chercher les hôtes connectés à votre réseau, vous devez choisir dans le menu principal : **Auto découverte => Recherche hôte** ou en cliquant sur l'icône du recherche trouvant dans la barre d'outil.

Une fenêtre va apparaître dont son nom est HC_ScanNetwork Recherche hôte, vous devez entrer une plage d'adresse IP, ensuite, lancez le recherche soit vous cliquez sur l'icône de lancement trouvant à la barre d'outil ou suivre ce chemin : **Recherche => Commencer**.

Vous pouvez aussi sauvegarder votre capture en cliquant sur l'icône d'enregistrement ou suivre ce chemin : **Fichier => Enregistrer**.

Vous pouvez retourner au programme principal par juste une petite clique sur l'icône (home) de la barre d'outils.

3. Information système :

Pour plus savoir sur votre système, HC_ScanNetwork vous offre la possibilité d'y découvrir quelques informations système, juste vous suivez le chemin suivant : **Auto découverte => Info système**.

Vous pouvez également me contacter sur mon émail :
Chokri : chocox2002@hotmail.com

Nétographies

http://www.multimania.com/

http://www.guill.net/

http://www.cisco.com/

http://www.commentcamarche.net

http://www.ethereal.com/

http://www.techniques-ingenieur.fr

http://www.vbapi.com/

http://www.eu.microsoft.com/France/vbasic/

Bibliographies

Livre : Visual Basic 6.0 de Michel Pelletier, collection : LE TOUT EN POCHE