

# NTOP

## Rapport de stage

---

### Thème :

L'étude et développement de l'outil de monitoring NTOP  
Network Top pour une extension RTP

Encadrement pédagogique : Eric SOUDAN-GRESSIER  
Encadrement technique : Jöel BERTHELIN  
Stagiaire : Samundeswary RAMACHANDRA  
Lieu : Laboratoire du CEDRIC  
CNAM PARIS

Septembre 2003

# Table des matières

---

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Quelques définitions . . . . .	3
<b>2</b>	<b>Les outils de métrologie</b>	<b>5</b>
2.1	Les applications utilisant SNMP . . . . .	5
2.1.1	SNMP . . . . .	5
2.1.2	MRTG . . . . .	6
2.1.3	RRDTool . . . . .	6
2.2	Les applications autonomes : sans le support SNMP . . . . .	6
2.2.1	Big Brother . . . . .	6
2.2.2	NetSaint . . . . .	7
2.2.3	Ethereal . . . . .	8
2.2.4	IpTraf . . . . .	8
2.2.5	NTOP . . . . .	9
<b>3</b>	<b>Les protocoles de routage</b>	<b>10</b>
3.1	DVMRP Distance Vector Multicast Routing Protocol . . . . .	10
3.2	MOSPF Multicast Open Shortest Path First . . . . .	11
3.3	PIM DM Protocol Independent Multicast Dense Mode . . . . .	12
3.4	PIM SM . . . . .	13
<b>4</b>	<b>NTOP Distribution.</b>	<b>15</b>
4.1	NTOP . . . . .	15
4.2	Architecture de NTOP . . . . .	17
<b>5</b>	<b>Développement</b>	<b>18</b>
5.1	RTP RFC 1889 . . . . .	18
5.2	RTCP . . . . .	19
5.3	Programmation . . . . .	20

5.3.1	Installation . . . . .	20
5.3.2	Développement . . . . .	20
<b>6</b>	<b>Apport personnel</b>	<b>22</b>
6.1	Apport technique . . . . .	22
6.1.1	Le système d'exploitation Linux . . . . .	22
6.1.2	C/C++ . . . . .	22
6.1.3	Réseaux . . . . .	23
6.2	Approfondissement des outils de monitoring . . . . .	23
6.3	Apport professionnel . . . . .	23
6.4	Apport sociaux . . . . .	23
6.5	Une déception cependant . . . . .	24
<b>7</b>	<b>Conclusion</b>	<b>25</b>
<b>8</b>	<b>Annexe</b>	<b>26</b>

# Introduction

---

Depuis quelques années les réseaux connaissent une forte croissance, ils deviennent de plus en plus vastes, utilisent de nouveaux médias de transmission, de nouveaux outils de communication. Face à cette croissance apparaît un besoin d'outils gestion, de contrôle et de supervision des réseaux, ces outils sont de plus en plus pointus et spécialisés. Ntop est à l'origine d'un besoin de sondage de réseau, de surveillance. La première partie de ce stage a été consacrée à la découverte et compréhension d'un outil de surveillance NTOP ( Network Top) et des protocoles de routage et la deuxième partie fut le développement d'un plugin qui permettra la visualisation du trafic multicast puis du trafic RTP et RTCP.

## 1.1 Quelques définitions

**Supervision :** Action de Superviser, c'est vérifier régulièrement et systématiquement le bon fonctionnement d'un service, d'un équipement et, en cas de dysfonctionnement, tenter une action correctrice ou alerter les personnes compétentes capables de rétablir la situation.

**Monitoring (audit)** Par l'intermédiaire de graphes, c'est la surveillance en temps réel de l'évolution de l'utilisation de l'infrastructure technique (bande passante consommée, espace disque consommé, charge CPU'). Les données collectées pour le monitoring peuvent être utilisées pour réaliser le reporting.

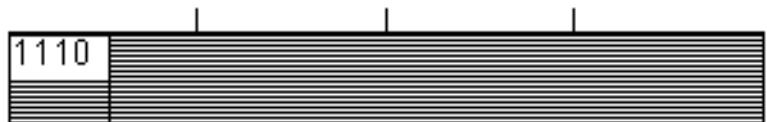
**Reporting** Repose sur l'archivage et le traitement des données collectées pour le monitoring. Le reporting permet donc de visualiser à moyen et long terme via des graphes l'évolution de la consommation des composants du système d'information. Le reporting est utilisé pour réaliser le capacity planning

**Capacity planning** Action qui consiste à prévoir, à l'avance, en fonction du reporting, l'évolution des composants du système d'information afin de prévenir toute pénurie de ceux ci.

**Multicast** Le principe du Multicast est la transmission d'un à plusieurs. Une seule source peut émettre des données vers plusieurs destinataires sans être obligée de dupliquer les datagrammes. Les avantages du multicast ne sont pas seulement logiques : pour en tirer partie, les médias de transmission utilisés (comme Ethernet) doivent aussi fournir un support pour le Multicast. Si le multicast est supporté dans le réseau, les performances s'améliorent énormément du fait que c'est un seul paquet qui est émis à N récepteurs, et transmis en une seule fois. Le Multicast IP est largement utilisé en vidéo, en conférence audio, etc. Le principe du routage Multicast est de transmettre les données pour un triplet (source, groupe) via un arbre du routage multicast de la source vers les destinations. A priori, quand une source émet, elle ne sait pas quels destinataires veulent recevoir ses paquets. Les membres dans chaque groupe lui sont inconnus. C'est le protocole de gestion de groupes IGMP (Internet Groupe Management Protocole) qui gère l'abonnement ou le désabonnement des membres aux groupes. Au niveau de l'adressage, les adresses multicast appartiennent à ceux de la classe D :

1. Adresses commençant par les 4 bits 1110
2. Plage d'adresses de 224.0.0.0 à 239.255.255
3. 28 bits d'adresses utiles soit 250 millions d'adresses environ

**Classe D  
multicast**  
224 - 239.255.255.255



# Les outils de métrologie

---

Il existe actuellement un grand nombre d'outils qui permettent de superviser l'ensemble des éléments du réseau tel que Angel Network Monitor, Autostatus, Big Brother, Netup, Node Watch.... Après un bref aperçu des applications les plus utilisées, l'accent sera porté sur celui qui servira de support au projet qui m'est affecté. Tout d'abord il existe deux catégories d'outils de monitoring : ceux qui s'appuient sur le protocole SNMP et ceux qui sont autonomes.

## 2.1 Les applications utilisant SNMP

### 2.1.1 SNMP

Il s'agit d'un protocole fonctionnant au niveau sept du modèle OSI qui permet à l'administrateur réseau d'interroger les éléments du réseau. Le principe est le suivant : sur chaque élément de l'infrastructure est installé un programme qui est un agent SNMP et ce dernier enregistre en permanence les informations relatives à l'élément. Par ailleurs, il stocke les informations ainsi recueillies au sein d'une base de données : la MIB ( Management Information Base). Le protocole s'opère sur le modèle d'un client-serveur de la manière suivante : la station d'administration, qui implante le client, ( NMS Network Management Station) exécute un programme de gestion SNMP. Son but principal est de contrôler les stations du réseau et de les interroger sur différentes informations, les agents SNMP implantent les serveurs. Le fonctionnement des agents : chaque agent est placé sur un nœud du réseau qui est dit administrable (MN : Managed Node) . Ces nœuds sont des hôtes qui peuvent être des stations de travail, serveurs, des éléments d'interconnexions. (ex : logiciels, stations de travail, routeurs, concentrateurs, ponts, etc.). Dans le cadre du projet, l'étude a été effectuée sur les serveurs, l'en-

semble des éléments d'interconnexion du réseau et les supports physiques. Simple Network Management Protocol est donc un protocole de gestion permettant de surveiller, de contrôler, de collecter des statistiques concernant les éléments du réseau.

### **2.1.2 MRTG**

MRTG est un outil graphique très avancé écrit par Tobias Oetiker et Dave Rand pour représenter graphiquement les données que les gestionnaires SNMP lisent sur les agents SNMP. Il produit une page HTML avec des images GIF sur le trafic entrant et sortant des interfaces du réseau, en temps réel. Les informations sont collectées par des requêtes envoyées par le serveur sur lequel il est installé, puis ces données sont inscrites dans le fichier de configuration de MRTG.

### **2.1.3 RRDTool**

RRD est un système pour stocker et montrer des données évoluant dans le temps (c.-à-d. largeur de bande de réseau, moyenne de charge de serveur). Il stocke les données d'une manière très compacte et il présente les graphiques utiles en traitant les données. Il peut être employé par l'intermédiaire de scripts simples. Les scripts nécessaires pour une telle structure :

- rrd-create.sh : ce script permet de créer/initialiser une base de données rrd
- rrd-update.sh : ce script permet d'ajouter les dernières valeurs mesurées à la base rrd. Il s'agit d'un script qui sera lancé régulièrement afin de réécrire les données périodiquement.
- graph.sh : ce script met à jour les images (graphs) dans le répertoire du serveur web. Ces graphiques feront l'objet d'un affichage dans le site. On peut donc ainsi visualiser des graphes en temps réel et en historique ( un jour, une semaine, un mois ou une année).

Remarque : RRDtool est une extension des fonctionnalités et capacités de Mrtg

## **2.2 Les applications autonomes : sans le support SNMP**

### **2.2.1 Big Brother**

Ce logiciel de supervision fonctionne sur le modèle Client-Serveur, il utilise des scripts afin de connaître en temps réel l'état des différents équipements d'une machine et du réseau (connexion, charge cpu, taux remplissage des disques, mémoire, trafic). Un client Big Brother est installé sur chacun des composants dont on souhaite connaître l'état et un serveur Big brother

est installé sur une machine qui se charge de collecter les informations des autres clients. Les informations ainsi recueillies sont définies par trois états

- état normal
- état «warning»
- état «panic»

Le seuil de ces différents états étant défini lors de l'installation et de la configuration du client et serveur de Big Brother.

Le serveur se charge donc de récupérer les différents états des composants et leur associe un code couleur(vert, orange et rouge). Ces données peuvent être visualisées grâce à la création d'une page HTML qui met en forme les données : en colonne les données sur lesquelles on effectue la supervision, et en ligne les différentes machines dont sont issues ces données. Chaque case du tableau contient une icône symbolisant l'état du matériel, avec un lien vers une page contenant plus d'éléments descriptifs de l'équipement monitoré. L'état du réseau est mis en évidence par la couleur du fond de la page HTML, l'administrateur peut- être également prévenu par mail lorsque l'état atteint un seuil alarmant.

### **2.2.2 NetSaint**

NetSaint est un logiciel de supervision des réseaux : supervision des services réseau, c'est aussi un logiciel de supervision des systèmes. Les fonctionnalités de NetSaint :

1. Supervision de services réseau ( SMTP, POP3, http, NNTP, PING.)
2. Supervision des ressources des hôtes (charge du processeur, utilisation du disque'.)
3. Contrôle parallélisé des services
4. Possibilité de définir une hiérarchie dans les hôtes grâce aux hôtes parents, permettant la détection et la distinction entre les hôtes en panne et ceux qui ne sont plus accessibles
5. Notification de l'apparition ou de la disparition de problèmes sur les hôtes ou les services ( mail..)
6. Possibilité de définir des gestionnaires d'événements qui sont lancés automatiquement lors de l'apparition d'événements concernant les hôtes ou les services, pour une résolution préventive des incidents
7. Rotation automatique des fichiers journaux ( qui servent à stocker l'état courant de tous les services et les hôtes supervisés). Ce fichier est supprimé à l'arrêt de NetSaint et recréé au démarrage
8. Support de la supervision redondante
9. Interface web optionnelle pour visualiser l'état du réseau, les notifications et l'historique des problèmes, les fichiers journaux'

### 2.2.3 Ethereal

Ethereal est un outil de surveillance réseau, qui permet de faire des captures de trames. Cet analyseur de trames utilise une interface graphique et il est basé sur la bibliothèque libpcap, qui fournit des outils pour capturer les trames réseau. Libpcap est une bibliothèque d'outils permettant de faire la capture des trames qui circulent sur le réseau, on peut ainsi faire des statistiques, de la surveillance de réseau, du débogage et bien d'autres choses.

Cet outil permet de visualiser les trames via une interface graphique dont l'affichage des résultats se décompose en trois parties : ( voir Schéma )

1. La liste des trames capturées disponibles en dessous de la barre de menu avec un affichage synthétique du contenu de chaque trame.
2. La décomposition exacte de la trame actuellement sélectionnée dans la liste. Cette décomposition permet de visualiser les champs des entêtes des protocoles ainsi que l'imbrication des différentes couches de protocoles connus.
3. La troisième zone contient la trame (le début s'il est trop gros) affichée en hexadécimal et en ASCII.

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

### 2.2.4 IpTraf

IPTraf est un outil d'observation réseau pour les réseaux IP. Il intercepte les paquets sur le réseau et donne plusieurs éléments d'information sur le trafic IP courant Les informations retournées par IPTraf incluent :

1. Le total IP, TCP, UDP, ICMP, et non-IP des octets comptés
2. Les adresses, les ports des sources et destinations des segments
3. Les segments et bits TCP comptés
4. L'état des drapeaux TCP
5. Les informations sur source et destination UDP
6. Les informations de type ICMP
7. Les informations source et destination OSPF
8. Les statistiques des services TCP et UDP
9. Les statistiques des stations LAN
10. IPTraf peut être utilisé pour observer la charge sur un réseau IP.

```

IPTraff
TCP Connections (Source Host:Port) ----- Packets ----- Bytes Flags Iface
ads1-65-71-187-105.ds1.okcynk.sw:3589 > 844 1263048 --A- eth0
ppp08-bacd.mozcom.com:1214 > 571 26374 --A- eth0
68.10.252.64.snet.net:3676 > 657 962504 --A- eth0
ppp08-bacd.mozcom.com:1214 > 465 21816 --A- eth0
pcp01238708pcs.frsrc101.ai.comca:1214 > 575 860632 --A- eth0
ppp08-bacd.mozcom.com:1176 > 390 18066 --A- eth0
ool-18ba0fc2.dyn.optonline.net:1063 > 316 472852 --A- eth0
208.160.255.153:2019 > 225 10350 --A- eth0
216.49.88.100:www = 4 1246 -FA- eth0
61.9.18.19:1298 = 7 878 -FA- eth0
pc03-bacd.mozcom.com:3329 = 199 11885 --A- eth0
server13.iicinternet.com:www = 198 292671 --A- eth0
TCP: 1809 entries ----- Active

Non-IP (0x4) (162 bytes) from 00d0bacceb44 to 0180c2000000 on eth0
ARP request for 207.0.115.44 (107 bytes) from 0030f212f000 to ffffffff d
ICMP echo req (84 bytes) from riker.mozcom.com to w1.scd.yahoo.com (src HMa
ICMP echo rply (84 bytes) from w1.scd.yahoo.com to riker.mozcom.com (src HMa
Non-IP (0x4) (130 bytes) from 00d0bacceb43 to 0180c2000000 on eth0
Non-IP (0x4) (46 bytes) from 00d0bacceb44 to 0180c2000000 on eth0
Bottom ----- Elapsed time: 0:01
Packets captured (all interfaces): 73890 | Flow rate: 119.60 kbits/s
Up/Dn/PgUp/PgDn-scroll M-more TCP info W-chg actv win S-sort TCP X-exit

```

Une capture de IpTraff

## 2.2.5 NTOP

Ntop est un outil de surveillance réseau qui permet de visualiser les communications, l'utilisation des protocoles et des services Internet, l'utilisation de la bande passante, la surveillance des paquets circulants sur le réseau. Ntop montre donc le trafic réseau. Pour la visualisation il crée une liste d'hôtes qui utilisent le réseau et indique les informations concernant le trafic IP généré par chaque hôte. De plus le trafic est ordonné suivant les hôtes et les protocoles. Ntop existe également sous mode Web, c'est-à-dire plusieurs utilisateurs peuvent accéder aux informations concernant le trafic grâce à un navigateur Web. Les données sont présentées selon différentes sections : tri du trafic par données envoyées, par données reçues, répartition du trafic, statistiques du trafic, trafic IP, trafic par protocole IP.

Cette étude nous montre l'importance du nombre d'outils existant actuellement pour la surveillance du réseau. Cependant chaque outil a sa spécificité, ce qui permet une certaine complémentarité entre ces différents outils.

# Les protocoles de routage

---

L'étude de cette partie est primordiale dans la compréhension du projet dans son ensemble. Il existe deux grands types de protocoles de routage : Les protocoles en mode dense

- DVMRP
- MOSPF
- PIM-DM

Le fonctionnement de ce mode est le suivant : Flood + Prune, (ie inondation + élagage). Il s'agit d'inonder le réseau puis d'élaguer les branches qui ne sont pas immigrées par le trafic multicast. L'élaguage demande des inondations périodiques, ce qui n'est pas très adapté à de très grands réseaux. Ce mode est intéressant lorsque le nombre d'abonnés à un groupe est important et compact du point de vue de la topologie.

Les protocoles en mode épars : (clairsemé) sont plus adaptés à une population dispersée dans le réseau, qui permet donc un ciblage des abonnés par l'envoi des données vers les lieux où elles sont demandées, donc avec un abonnement. Ce mode permet d'avoir un arbre partagé par plusieurs groupes multicast alors qu'en mode dense, chaque groupe possède un arbre. Ce facteur factorise l'utilisation des ressources dans un grand réseau. D'où son efficacité.

- PIM-SM

## 3.1 DVMRP Distance Vector Multicast Routing Protocol

- Il a sa propre table de routage DVMRP
- Il construit un arbre de distribution séparé pour chaque source/groupe
- Il utilise l'algorithme de "reverse path forwarding" RPF pour propager et élaguer

- La propagation : il diffuse les paquets sur toutes les interfaces de sortie de l'arbre de distribution, en supposant au départ que chaque branche mène à des membres actifs du groupe.
- Elagage : Eliminer les branches de l'arbre sans membre du groupe multicast, coupant la transmission sur les LANs sans récepteur intéressé, élague aussi les chemins redondants non optimaux de chaque récepteur vers la source

**Les avantages et inconvénients : Plus efficace pour les distributions denses de récepteurs multicast, largement utilisé sur le MBONE (par le passé), inapproprié pour les grands réseaux avec peu de récepteurs intéressés dûs au mécanisme "propager et élaguer"**

### 3.2 MOSPF Multicast Open Shortest Path First

Extension multicast OSPF protocole de routage unicast OSPF : les routeurs utilise des annonces d'état de lien (LSA) pour connaître toutes les liaisons disponibles du réseau (route et chemin au moindre coût)

MOSPF : Inclut une information multicast dans l'annonce de l'état de lien OSPF pour construire les arbres de distribution multicast (chaque routeur maintient une image à jour de la topologie de tout le réseau)

- Les LSAs d'appartenance à un groupe sont propagés dans l'ensemble du domaine de routage OSPF, afin que les routeurs MOSPF puissent mettre à jour et calculer les listes des interfaces de sortie.
- Utilise l'algorithme Dijkstra pour calculer l'arbre du plus court chemin
- Un calcul séparé est requis pour chaque couple Source/Groupe
- Ne propage pas partout le trafic multicast pour créer les états, utilise les LSAs et la base de données d'états de liens.
- Dépendant d'un protocole : marche seulement sur les réseaux basés sur OSPF
- Problèmes significatifs de passage à l'échelle l'algorithme Dijkstra s'exécute pour tous les couples multicast (S,G)!
- Ne supporte pas les arbres partagés
- Inapproprié pour les grandes interconnexions de réseaux avec beaucoup de sources et de récepteurs.
- MOSPF nécessite de propager artificiellement l'arbre multicast entre domaine OSPF, ce qui engendre une complexité superflue.

### 3.3 PIM DM Protocol Independent Multicast Dense Mode

- Utilise la technique du RPF Reverse Path Forwarding : Si un datagramme est reçu sur l'interface utilisée pour envoyer un datagramme unicast vers la source de ce datagramme ( reverse), il sera propagé sur les autres interfaces du routeur, sinon il sera rejeté.
- Utilise la table de routage unicast pour transmettre les données multicast aux destinataires ( à cause du RPF )
- Utilise l'arbre de diffusion basé sur la source si l'état (S,G) existe
- Le mode de fonctionnement de PIM-DM

Trois mécanismes sont utilisés :  
PRUNE, GRAFT et LEAF

**PRUNE** Il s'agit de l'élagage, un routeur PIM qui n'a plus de récepteur local ni de routeur en aval envoie un message prune (PrunePacket) à ses routeurs voisins et à la source émettrice pour ce groupe multicast

**Graft** Il s'agit de la greffe : à l'apparition d'une nouvelle demande pour un groupe donné, l'information est remontée vers le routeur voisin jusqu'à la source (graftPacket). Au passage les routeurs vont créer des entrées dans leur table de routage en ajoutant l'interface par laquelle le message est arrivé.

**Leaf Network Detection** Afin qu'un routeur apprenne l'existence des membres à desservir, et pour savoir s'il fait partie des routeurs feuilles ou non, il utilise la technique du « leaf-network detection ». La technique « leaf-network detection » est basée sur la détection des routeurs feuilles, ceci est réalisé quand chaque routeur dans le domaine PIM envoie un message Hello aux routeurs voisins. Si le routeur en amont ne reçoit pas des messages hello d'un routeur en aval pendant 30 seconds, le routeur sait qu'il est le routeur feuille et il est le responsable de la transmission des messages IGMP aux membres connectés avec lui.

Les messages PIM

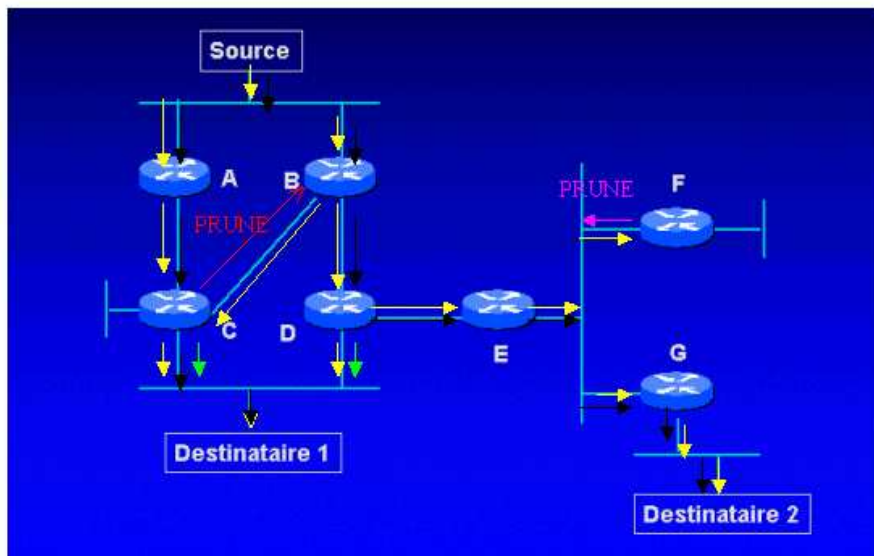
**PIM Hello** Tous les routeurs envoient un message Hello pour connaître quelles sont les interfaces utilisées pour chaque routeur et pour savoir aussi si le routeur est un routeur feuille ou non.

**PIM-Graft** PIM-DM l'utilise pour l'adhésion des membres à un groupe, si une entrée (negative cache entry) est construite, le routeur envoie un message Graft vers la source (rejoindre des branches à l'arbre multicast).

**PIM-Graft-Ack** chaque routeur reçoit le message Graft-Ack, il répond en envoyant un Graft-Ack Message.

**PIM- Assert** un routeur envoie le message Assert pour élire un transmetteur sur un LAN à plusieurs routeurs

**PIM-Join/Prune** un message Join/Prune envoyé pour joindre des membres à l'arbre multicast (ajouter des branches) ou pour quitter le groupe des membres de l'arbre multicast (couper des branches).

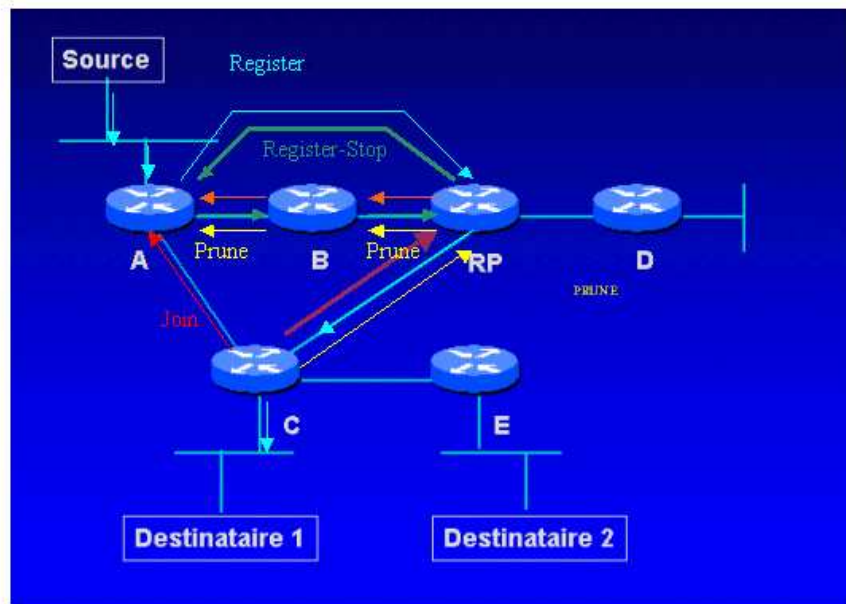


1. Inondation sur tout le réseau
2. Elagage de C vers un voisin non RPF
3. Assert sur le Lan de C et D : suppression des doublons
4. F n'ayant pas de membre, élagage
5. ARBRE MULTICAST FINAL

### 3.4 PIM SM

- Mode d'abonnement explicite (Join)
- La source s'enregistre auprès d'un point de rendez-vous (RP Rendez-Vous Point)
- Le RP est la racine de l'arbre de diffusion multicast partagé (RPT Rendez-Vous Point tree)
- Le RP est configuré statiquement ou dynamiquement par un «Auto-RP» ou «candidateRP »
- Pour s'abonner le destinataire s'enregistre, il envoie un Join vers le RP pour un groupe donné

- Le flux multicast parcourt l'arbre partagé (RPT) ou/puis l'arbre dont la source est l'origine SPT : Shortest Path Tree, il s'agit de l'arbre de chemin le plus court (Voir Annexe)



1. Destinataire 1 se joint au groupe, C envoie un Join au Rp et crée l'état (\*,G).
2. RP crée l'état(\*,G), met un lien vers C sur l'interface de sortie
3. Source envoie des données, qui sont encapsulées par A et il envoie un register vers RP
4. RP=> crée l'état (S,G),envoie les données sur l'arbre partagé, envoie un Join vers la source et au passage A et B créent l'état (S,G).
5. Quand les données arrivent normalement à RP, il envoie un Register-Stop
6. Destinataire 1 cherche le chemin le plus court, C envoie un Join vers la Source.
7. C reçoit (S,G), il envoie Prune vers la source sur l'arbre partagé, RP efface le lien vers C et transmet Prune vers Source

# NTOP Distribution.

---

## 4.1 NTOP

Ntop montre l'activité courante d'un réseau. Il affiche une liste des hôtes qui sont actifs dans le réseau et rapporte les informations concernant le trafic (quelqu'il soit IP ou non) généré par ces derniers. Ntop peut-être exécuté soit sur un terminal(mode interactif) soit sous forme de pages web. Le trafic est trié suivant les hôtes et protocoles. Dans les cas où Ntop serait exécuté en mode Web, plusieurs utilisateurs (ayant accès à ces données) peuvent consulter à distance ces informations. En mode web Ntop fait office de serveur http en écoutant sur le port spécifié. Ce mode propose un ensemble de données sous forme graphique concernant trafic et des statistiques. Les informations qui sont accessibles via ce web de consultation sont les suivantes :

- Adresse IP
- Date/Heure du dernier paquet transmis
- Adresse Mac
- Domaine
- Vendeur de la carte réseau
- OS utilisé
- Nom NetBios
- Localisation hôte (local/externe)
- Données envoyées (en paquets, en octets)
- Données reçues(en paquets, en octets)
- Routeur
- Répartition des protocoles IP
- Données envoyées et reçues par protocoles (UDP, ICMP, ARP, OSPF, IGMP'.)
- Les derniers protocoles, adresses, ports, sessions pour un hôte
- Répartition en pourcentage, nombre de données transmises par protocoles, par hôte, par domaine, par sous réseau, routeur, session, statis-

tiques unicast, multicast.

- Statistiques suivant la taille des paquets et débit horaire

Toutes ces informations sont représentées sous formes de camemberts, graphiques ou tableaux. ( Voir Annexes copies d'écrans ) La liste des protocoles et services filtrés par Ntop sont les suivants :

TCP/UDP/ICMP

(R)ARP

IPX

DLC,Decnet

AppleTalk

Netbios

IP

FTP

HTTP

DNS

Telnet

SMTP/POP/IMAP

SNMP

NFS

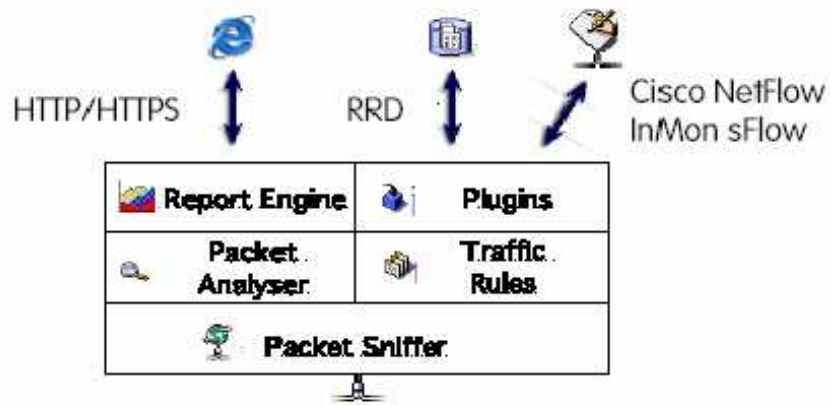
X11

Une fois que Ntop est démarré, les datagrammes capturés sont analysés et catalogués suivants leur hôte et protocole au fur et à mesure. En interne, Ntop garde une liste de tous les derniers hôtes actifs et connexion TCP établies. Pendant la capture des paquets, qui sont analysés, une connaissance du réseau est activée : sur la topologie du réseau, les relations entre les hôtes (Routeur, DNS).

Du point de vue technique :

Afin de minimiser la mémoire requise, lorsque le nombre d'hôtes ou de connexions dépasse un certain seuil (par défaut 2048), un processus de collecte est démarré. C'est une preuve du bon fonctionnement de Ntop malgré une croissance des données. Donc Ntop dénote une certaine capacité de passage à l'échelle. Ntop permet aux utilisateurs de spécifier un filtre qui restreint le type de trafic traité par Ntop. Si aucun filtre n'est spécifié, tous les paquets capturés seront traités. Dans le cas contraire seul les paquets correspondant au filtre précisé seront traités. Par exemple, la commande «ntop host panoramix.cnam.fr» entraînera comme conséquence un traitement des paquets qui concernent panoramix. Les filtres correspondent au standard BNF.

## 4.2 Architecture de NTOP



L'étude de l'architecture permet d'appréhender les améliorations qu'on peut apporter à l'outil déjà existant, au vue des attentes par rapport au projet Concert en réseau, Le besoin émanant de ce projet est le suivant : une visualisation des données RTP et RTCP. En fait il s'agit de pouvoir distinguer les données transportées par ces deux protocoles afin de juger l'état du réseau. Ceci conduit donc à une étude approfondie de ces deux protocoles.

# Développement

---

## 5.1 RTP RFC 1889

RTP est un protocole de transport adapté à la diffusion de flux audio et vidéo en temps réel et de bout en bout. Conçu pour fonctionner sur IP, il doit assurer un temps de réponse approprié pour des flux circulant sur des chemins non pré-établis (IP fonctionnant en mode datagramme). RTP se situe, dans l'architecture OSI, au dessus des protocoles de transports standards que sont UDP et TCP, mais est préférentiellement utilisé sur UDP (toutes les fonctionnalités de TCP ne sont pas nécessaires pour le streaming, en particulier le mode connecté). RTP est couramment utilisé avec RTCP pour assurer un contrôle des flux.

RTP propose les fonctionnalités suivantes :

- Support des flux en temps réel avec identification de leur contenu (Payload type)

La spécification précise du contenu des messages RTP permet au récepteur de traiter correctement le flux, qui peut être de type PCM, MPEG1/MPEG2 (Audio et vidéo), JPEG vidéo, ou flux vidéo H.261. Si un flux d'un type non supporté doit être utilisé, il est possible de spécifier un nouveau type et format de payload. Même si, à un moment donné, un seul type de payload RTP peut être envoyé, il est possible de le changer durant la transmission pour l'adapter à une éventuelle congestion du réseau ou à de nouvelles caractéristiques des liens traversés ( mixeurs/transmetteurs ).

- Numérotation des paquets pour détecter les pertes et réordonner ceux - ci dans l'ordre d'émission.
- Horodatage des paquets pour synchroniser l'émetteur sur le récepteur. Grâce à un système d'horloge nommé timestamping, émetteur et récepteur restent synchronisés. Cette horloge est envoyée avec le paquet RTP puis incrémentée régulièrement. L'horloge est utilisée par le ré-

cepteur pour reconstruire le flux selon le timing déterminé à l'émission. Elle sert également lors de la synchronisation des flux audios et vidéos.

- Support de l'unicast et du multicast.  
Originellement prévu pour de la diffusion en multicast, RTP fonctionne parfaitement au dessus d'un mode unicast.
- Identification de source (chaque récepteur peut connaître le récepteur qui est en train de lui «parler»)

L'ensemble de ces fonctionnalités sont implémentées dans un entête de 12 octets précédant chaque paquet. Cet entête est décrit par le tableau suivant :

0				15				16				32			
V	P	X	CC	M	PT			Numéro de Séquence							
Horodatage (Timestamp)												Identificateur de la Source de Synchronisation (SSRC)			
Identificateur(s) de la(les) Source(s) Contributrice(s) (CSRC)												Données			

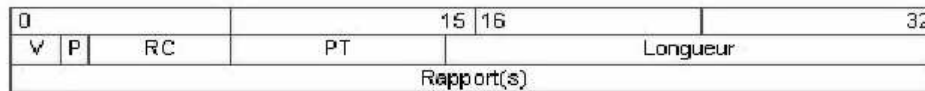
Champ	Nbr. bits	Fonction
V : Version	2	Définit le numéro de version de RTP : actuellement 2
P : Padding	1	Indice permettant de spécifier que les octets de données ont une partie de bourrage
X : Extension	1	Spécifie (si à 1) qu'un en-tête supplémentaire suit le paquet
CC : Nombre de CSRC	4	Contient le nombre d'identificateurs de sources contributrices contenues dans la liste CSRC
M : Marker	1	Indique la présence de descriptifs contenant la trace d'événements particuliers
PT : Payload Type	7	Donne le type de contenu audio et/ou vidéo transporté par le paquet ainsi que son format de codage (PCM, G721, JPEG).
Numéro de Séquence	16	Numéro d'ordre d'émission des paquets, il est incrémenté d'une unité à chaque paquet envoyé. Il permet ainsi au destinataire de détecter une perte de paquet et de réorganiser des paquets qui, du au réseau, seraient arrivés dans un ordre différent de celui d'émission.
Horodatage	32	Horloge système ou horloge d'échantillonnage de l'émetteur. Elle doit être monotone et linéaire pour assurer la synchronisation des flux.
SSRC	32	Identifie la source de synchronisation, c'est-à-dire l'émetteur sur lequel il faut caler la base de temps commune à tous les participants.
CSRC	32	Liste des participants ayant apportés leur contribution (audio, vidéo) aux données du paquet. Peut être nul.

Entête RTP

## 5.2 RTCP

Il est utilisé conjointement à RTP, RTCP ajoute essentiellement des fonctionnalités de contrôle en offrant la possibilité au récepteur d'informer l'émetteur sur l'état de la transmission. RTCP permet à l'émetteur comme au récepteur d'échanger des messages de contrôle (l'émission de ces paquets est toutefois limitée à 5% du volume de la session par RTP, pour éviter toute congestion).

Il existe cinq types de paquets RTCP. Chacun d'eux possède son propre en-tête, en plus de l'en-tête commun RTCP décrit par le schéma suivant :



Champ	Nbr. bits	Fonction
V : Version	2	Définit le numéro de version de RTP : actuellement 2.
P : Padding	1	Indice permettant de spécifier que les octets de données ont une partie de bourrage.
RC : Report Counter	5	Contient le nombre de rapports contenus dans le paquet (un rapport pour chaque source).
PT : Packet Type	8	Donne le type de rapport du paquet (SR, RR, SDES ou BYE).
Longueur	16	Longueur du paquet.

Entête RTCP

## 5.3 Programmation

### 5.3.1 Installation

La toute première étape du développement a été le téléchargement et l'installation de la dernière version stable de NTOP (version 2.2), afin que les modifications que j'apporterai à l'outil puissent être aisément intégrées.

Il s'agit d'une phase qui nécessite la compilation d'autres packages et bibliothèques. Donc dans un premier temps, j'ai effectué la compilation des programmes et bibliothèques de dépendances telles que gdchart, zib, lib png. Ce sont des bibliothèques qui sont dans Ntop mais dont il faut faire la configuration, l'installation et la compilation avant de faire celui de Ntop. Une fois cette étape effectuée, il a fallu modifier les chemins d'accès aux bibliothèques dans le fichier makefile. C'est une fois ces différentes étapes effectuées que la configuration, l'installation et la compilation ( ./configure, make, makeinstall ) de Ntop peuvent se faire correctement.

### 5.3.2 Développement

Toute l'étude que j'ai fait au préalable sur les protocoles était destinée à un but très précis, dans un premier temps afin d'avoir une connaissance globale du projet qui m'était destiné, puis dans un deuxième temps afin de m'aider dans le développement que j'avais à entreprendre.

Le développement s'est déroulé en plusieurs étapes :

- Le développement d'un programme à part qui capture les trames multicast dans un premier temps. Cela a été réalisé par la mise en place d'un plugin : multicast (voir annexe). Pour cela, j'ai tout d'abord fait l'étude de la gestion des trames multicast dans l'application. C'est-à-dire comment est gérée la structure qui contient les données des paquets multicast, comment et à quel niveau se fait l'affichage, la capture

des trames ( aussi bien multicast qu'unicast),l'étude des plugins, dans quels fichiers sont faits les déclarations des variables, le contenu de des différents fichiers. Cette première étude a entraîné une véritable connaissance de l'application NTOP, à savoir sa stucturation. Au-fur-et-mesure de l'avancement, un découpage très net de l'application s'est distingué. Je vais ci-dessous décrire brièvement les principaux fichiers :

**Globals-structures.c et Ntop-win32.c** : contiennent l'ensemble des déclarations des types et structures utilisés par NTOP, les déclarations des types d'entêtes des protocoles

**ReportUtils.c et report.c** : contiennent les fonctions qui servent à l'affichage des différents graphiques et des tableaux.

**Initialize.c** : contient les fonctions qui permettent d'initialiser les variables globales

**Pbuf.c** : il s'agit du cœur de Ntop, il contient les fonctions qui permettent la récupération des paquets, leur mise à jour ( des données affectant chaque hôte), leur classement

**Traffic.c** : contient les fonctions de mise-à-jour de la réception des trames (l'incréméntation, reset)

- C'est après cette étude très ciblée que j'ai effectué la mise en place du plugin. Pour cela je me suis basée sur les différents plugins existants : tels qu'NFS, IGMP. Le codage du plugin s'est déroulé de la manière suivante : en me basant sur les autres plugin, je récupère les hôtes qui ont envoyés ou reçus des données multicast, puis je procède à leur affichage.
- Après ce premier codage, il a fallu intégrer ce nouveau plugin dans les fichiers de configuration où sont faites les déclarations de différents plugin. J'ai pu faire aisément cette étape grâce à une simple recherche de déclaration des autres plugins. Une fois cette étape effectuée, j'ai pu faire l'intégration de ce nouveau plugin au niveau de l'interface web. [voir annexe.]

Une fois cette étape franchie, j'ai pu désormais m'intéresser uniquement à la capture des paquets RTP, plus précisément. Pour la gestion des paquets RTP, j'ai programmé un module où se fait la déclaration des entêtes rtp : rtp.h. Il s'agissait pour moi de pouvoir créer des modules que je pourrai aisément modifier et retrouver. Puis je me suis intéressée à la capture effective des trames RTP, ce qui m'a amenée à l'intégration d'une partie importante de code dans le module pbuf.c

# Apport personnel

---

## 6.1 Apport technique

Au cours de ce stage j'ai eu l'occasion d'approfondir mes connaissances informatiques dans divers domaines. Il est évident que tout au long de ce stage j'ai eu l'occasion de mettre en pratique les connaissances théoriques que j'ai acquies durant mon parcours au CNAM. Ce passage à une mise en pratique est très enrichissante pour qui a très peu de pratique hors du CNAM.

### 6.1.1 Le système d'exploitation Linux

Ayant jusqu'à présent travaillé uniquement sur Windows et Unix, j'ai eu l'occasion de connaître Linux. Du fait qu'il s'agisse d'un environnement proche d'Unix, j'ai eu des facilités pour m'adapter à ce nouvel environnement.

### 6.1.2 C/C++

Au cours de mon parcours scolaire j'ai pu apprendre deux langages de programmation : Ada et Java. Le manque de temps et d'une réelle motivation est à l'origine du fait que je ne me sois jamais lancée dans l'apprentissage d'un nouveau langage. Ce stage m'a été donc un excellent tremplin pour une introduction au langage C. Par ailleurs, ce travail m'a permis de voir la difficulté d'un projet lorsqu'il est effectué par plusieurs personnes puisqu'il s'agissait de comprendre le code existant pour pouvoir intégrer ma partie. Je pense que c'est une expérience enrichissante dans la mesure où un projet est le fruit d'une collaboration. Donc par conséquent, je serais toujours amenée à travailler sur le code de divers personnes. C'est donc un travail d'adaptation dont la gestion n'est pas toujours aisée.

### **6.1.3 Réseaux**

Ce stage m'a permis de mettre en pratique les cours de réseaux auxquels j'ai assisté. De manière plus précise j'ai eu l'occasion d'examiner les trames du réseau à travers `ethereal`, de voir le découpage en clair à travers les différents entêtes, le corps des trames. Cette étude réelle du réseau m'a permis de voir les réseaux sous un aspect plus vivant. J'ai pu étudier en détail les protocoles les plus utilisés : IP, UDP, TCP, à travers les RFC dont je n'avais pas eu une réelle connaissance. Et ce pour comprendre comment étaient sectionnées les structures de l'outil `NTOP`.

## **6.2 Approfondissement des outils de monitoring**

Lors de mon dernier stage j'ai pu découvrir et travailler sur un outil de monitoring : `RRDTool`. Mettant en jeu essentiellement le protocole `SNMP`, il s'agit d'un outil de supervision de l'infrastructure informatique, tandis que l'approche réseau de ce second stage est dirigée vers les flux, le trafic réseau. Donc il s'agit d'une approche différente. Ces deux approches m'ont permis de voir l'importance du réseau, surtout la puissance des outils existant ( aussi bien de supervision que de sécurité).

## **6.3 Apport professionnel**

Ces différents apports techniques constituent un réel avantage au niveau expérience, au niveau pratique dans le cadre professionnel. Ce sont des atouts que je pourrai mettre en avant lors d'un entretien professionnel notamment.

D'autre part ce projet s'inscrit dans un projet plus vaste qui est le Concert Virtuel Réparti, la thèse de Nicolas Bouillot. La partie du projet que j'ai développé sera le support de supervision utilisé pour le concert qui sera présenté à IRCAM. Présentation durant laquelle je serais amenée à déployer le plugin que j'ai développé.

## **6.4 Apport sociaux**

C'est un projet durant lequel j'ai eu l'occasion de me rendre compte de l'importance de la communication. Etant donné que mon projet est rattaché au projet de Rémy Bonafous et de la thèse de Nicolas, j'ai beaucoup travaillé en leur collaboration. Cette collaboration m'a été très utile et enrichissante dans la mesure où je pouvais m'adresser à eux lorsque j'avais une question. Cela m'a permis d'avancer plus rapidement dans le projet.

## 6.5 Une déception cependant

Dans l'ensemble je ne peux nier le fait que ce stage fut une abondante source d'enrichissements, néanmoins du fait que je travaillais sur un outil en opensource, je souhaitais que mon travail (s'il arrivait à terme) soit intégré dans cet outil. Malheureusement, cette opportunité ne s'est pas présentée, car les personnes l'ayant développé n'ont pas montré d'intérêt particulier face à ce développement.

# Conclusion

---

Tout d'abord, je tiens à remercier M. Gressier pour m'avoir permis de faire ce stage et encouragée au départ, lorsque j'étais remplie de doutes. Puis je souhaite remercier M. Berthelin qui a assuré l'encadrement technique et m'a fait part de ses conseils, et enfin les autres étudiants du laboratoire du Cédric, à savoir Pierre Agret, Rémy Bonnafous et Nicolas Bouillot qui ont toujours pris le temps de m'aider lorsque j'en faisais la demande.

Ce stage m'a beaucoup apporté au niveau de l'expérience surtout dans le domaine des réseaux, un domaine qui connaît une réelle demande et surtout avec un futur enrichissant à prévoir avec l'arrivée Ipv6. Par ailleurs, ce stage a aussi confirmé mon engouement pour le domaine de la programmation. C'est d'ailleurs ces deux constatations qui m'ont aidée dans le choix de mes valeurs C. C'est donc un stage qui a permis d'appuyer mes connaissances théoriques ( sur les protocoles notamment) en passant par cette étape de pratique. Et enfin au delà de l'application je me suis rendue compte de l'importance et la puissance des outils de supervision dans le cadre de l'administration réseau.

# Annexe

---

# Bibliographie

---

- [1] Steve Oualline : Pratical C Programming *O'Reilly Associations, Inc*
- [2] P. Harbison, Guy L.Steele JR, Tartan Laboratoires : A reference manual
- [3] <http://xavier.dusart.free.fr/netsaint/index.html> *Juillet 2003*
- [4] <http://www.cis.ohio-state.edu/cgi-bin/rfc/rfc1889.html> *Juillet 2003*
- [5] <http://www.faqs.org/rfcs/rfc1889.html> *Juillet 2003*
- [6] <http://lea-linux.org/> *Juillet 2003*
- [7] <http://www.commentcamarche.net/> *Juillet 2003*
- [8] <http://www.networksorcery.com/enp/protocol/rtp.htm> *Juillet 2003*
- [9] <http://maxime.chambreuil.free.fr/prof/csi/report.pdf> *Juillet 2003*
- [10] <http://herveguillaume.free.fr/docs/Rapport.pdf> *Juillet 2003*
- [11] <http://people.ee.ethz.ch/~oetiker/webtools/mrtg/> *Juillet 2003*
- [12] <http://www.urec.cnrs.fr> *Juillet 2003*
- [13] [http://www.infres.enst.fr/~dax/polys/multicast\\_api/rtp.html](http://www.infres.enst.fr/~dax/polys/multicast_api/rtp.html) *Juillet 2003*

[14] <http://www.cultdeadsheep.org/old/Caracteristiques/caracteristiques.html>  
*Juillet 2003*

[15] <http://www.ntop.org/ntop.html> *Juillet 2003*