Réalisation systématique de traces des principaux protocoles réseau

Encadrant : O. Fourmaux

Etudiants : S. Benazzouz, F. Diagne, D. Oriol, F. Signorello

Table des matières

1 Cahier des charges

La complexité croissante des informations numériques échangées, l'émergence des nouvelles applications et des usages qui en découlent, ainsi que l'hétérogénéité des utilisateurs entraînent une méconnaissance importante du trafic Internet. Or pour satisfaire aux futurs besoins, l'évolution du réseau mondial est indissociable d'une parfaite connaissance et compréhension des caractéristiques du trafic.

C'est pourquoi le développement de techniques et d'outils de métrologie réseau - au sens propre « la science des mesures » - pour en capturer le trafic ainsi que des méthodologies pour en analyser ses caractéristiques est aujourd'hui un sujet de recherche de premier plan.

L'objectif de notre projet s'inscrit directement dans cette optique. Il consistera à produire de façon systématique des traces des principaux protocoles réseau afin de pouvoir les analyser avant que nous puissions en extraire une information simple et synthétique, permettant son utilisation à des fins pédagogiques.

La réalisation devra permettre de proposer une ou plusieurs traces des protocoles suivants :

- VLAN
- IPv4, IPv6, ICMP, DHCP, DNS
- Routage (RIP, OSPF, BGP)
- MPLS (RSVP-TE, LDP, multistack)
- Multicast (IGMP, PIM ...)
- QoS (RSVP, IntServ et DiffServ)
- Sécurité (IPSEC)
- Tunneling (L2TP)
- Management (SNMP)
- Applications usuelles (Telnet, FTP, TFTP, HTTP, SMTP, IMAP, POP, Webmail, ...)
- Applications temps réel (RTP, RTSP)
- Téléphonie (SIP)

Pour mener à bien ce projet, nous disposons d'une plate-forme simulant un environnement réel d'interconnexion de réseaux. Elle est constituée de quatre systèmes autonomes : deux entreprises connectées à travers deux ISP. De plus ces quatre AS sont composés de matériels hétérogènes sur lesquels seront implémentés les différents protocoles que nous allons étudier.

Chaque protocole imposant une stratégie d'analyse qui lui est propre, nous définirons systématiquement un scénario et une configuration d'écoute (points de capture) qui révéleront au mieux les caractéristiques du protocole étudié. Nous isolerons dans un deuxième temps les traces obtenues de toute information verbeuse et superflue -notamment les protocoles secondaires-, afin de leur rattacher une description claire et pédagogique.

2 Plan de developpement

Documentation Cette phase correspond à un temps de recherche d'informations :

> **Le matériel** que nous avons à notre disposition est très performant mais également très hétérogène, aussi la recherche de documentation sur la configuration, la maintenance et les possibilités offertes est vaste.
>
> **Les captures** de trafic exigent une recherche permettant de répondre aux questions : Quels logiciels de capture choisir ? Quelles options de filtrage offrent-ils ? Les fichiers de capture obtenus peuvent-ils être traités afin d'en extraire des informations ? Comment faire pour les extraire ?
>
> **Les protocoles et fonctionnalités** à mettre en place sur la plateforme demande une recherche importante. Celle-ci passe, entre autre, par la lecture des RFCs (Request For Comments) et tutoriaux disponibles.

Durée totale : 7 semaines

Réflexion La période de réflexion correspond aux choix à effectuer lors de l'implémentation. Il s'agit de définir précisément la topologie que nous souhaitons mettre en place afin que nous puissions par la suite déployer tous les protocoles, fonctionnalites et utilitaires nécessaires sur la plateforme.
Durée : 10-14 jours

Câblage et implémentation Cette étape correspond à la mise en pratique de tous les éléments dont nous avons convenus. Elle se réalise en plusieurs étapes :

> **Câblage** Nous avons commencé par interconnecter les appareils les uns avec les autres selon la topologie théorique que nous avons décidée de mettre en place.
>
> **Implémentation** Dans ce second temps, nous configurerons les matériels avec une configuration minimale afin de pouvoir faire communiquer toutes les machines de la plateforme : installation de routage statique ou dynamique et installation de systèmes d'exploitation sur les serveurs. Nous installerons quelques services, à savoir un serveur TFTP qui nous permettera de sauvegarder les congurations des matériels durant nos travaux et un service Telnet afin de pouvoir nous connecter depuis l'Internet sur la plateforme via une passerelle dédiée accessible par SSH.

Durée totale : 20-30 jours

Avec ces manipulations nous aurons une base solide pour implémenter chacun des protocoles et fonctionnalités dont nous souhaiterons réaliser la trace.

Rapport intermédiaire La rédaction du présent rapport nous a poussé à mettre en place des outils de travail collaboratif afin de répartir puis de synchroniser nos productions.
Durée : 10 jours

Implémentation des protocoles et réalisation de traces Nous allons implémenter en parallèle les protocoles et nous effectuerons les captures et le filtrage des traces. Cela nous imposera, notamment, l'installation puis la configuration des logiciels qui produiront le trafic.
Durée : 5 semaines

Extraction des informations Les données essentielles des traces réalisées après toutes les captures seront mises en évidence puis analysées. Ce travail sera fait avec un souci de clarté pédagogique et les traces seront complétées par des descriptifs.
Durée : 4 semaines

Préparations finales Phase ultime où nous préparerons à l'aide de réunion de groupes le rapport final et la soutenance.
Durée : 2 semaines

La page suivante présente le plan de développement sous la forme d'un diagramme de Gantt. Pour le compléter, un diagramme de charge est représenté en annexe.

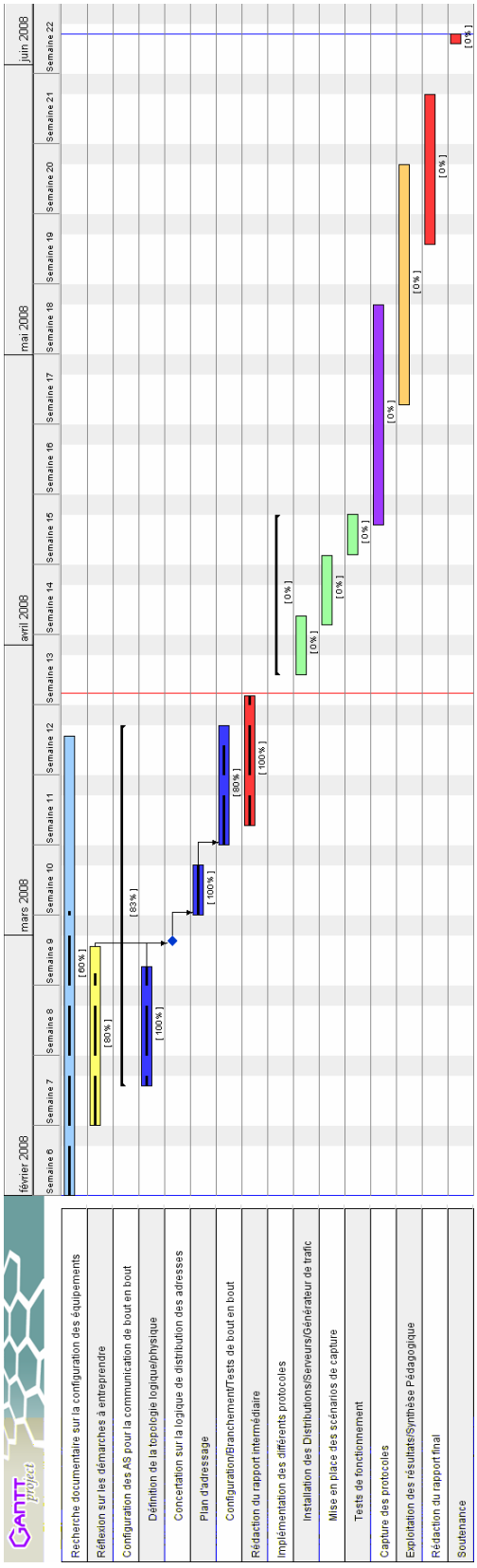


Diagramme de Gantt

3 Contexte technologique

En s'adaptant aux évolutions du monde réel, les protocoles réseau se sont multipliés et ont connu d'importantes évolutions. En effet, la forte augmentation du nombre d'utilisateurs au cours des dernières années a amené de nouveaux besoins. Aux usages historiques, tels que le courrier électronique et la navigation Web, se sont ajoutées diverses fonctionnalités qui ont conduit à la création de différents types de protocoles. Cette diversité a causé très rapidement un besoin de normalisation. L'un des plus grands organismes effectuant ce travail, l'IETF, a permis la rédaction de plusieurs RFCs dans le monde de l'Internet standardisant ainsi les implémentations pour une utilisation à grande échelle.

Suite à cette croissance, la quantité de trafic a évidemment été décuplée ; l'anticipation de la pénurie des adresses disponibles est devenue inévitable, et a directement conduit au développement de la version 6 d'IP. L'augmentation du nombre de flots a induit une nécessité de différenciation de ceux-ci en termes de contraintes, comme à cause de l'ajout de la téléphonie qui demande un débit constant et de fortes exigences de délai. Des mécanismes de qualité de service tentent d'apporter des solutions. L'intégration de services a également apporté des flots multimédias insérant une grande quantité de données dans les réseaux. Pour diminuer le nombre de paquets en circulation, le multicast à été développé. L'envoi de paquet d'un émetteur vers plusieurs récepteurs amoindrit la charge que doivent traiter les routeurs. Une utilité certaine de cette technique est, par exemple, la transmission de chaînes de télévision à un groupe d'abonnés ; elle pourrait de même résoudre les problèmes des sites web en vogue qui deviennent de sérieux goulots d'étranglement. De plus en plus d'utilisateurs se servant de réseaux informatiques pour des services variés, des données sensibles transitent par Internet, il faut les protéger en assurant le triptyque authentification, confidentialité et intégrité. La sécurité, inhérente à toutes communications numériques, a dû être pris en compte, que les communications se fassent de façon locale ou distante. Le découpage d'un réseau local en plusieurs réseaux virtuels est une première approche pour les échanges de type local. En assurant un nomadisme non contraignant, les procédés de tunnels sécurisés sont appréciés des entreprises dont les collaborateurs ne se trouvent pas forcément tous sur le même site physique. Le passage d'information de paiement sur des sites marchands ou le transfert de données personnelles sont garantis par des protocoles fiabilisant ces échanges. Cet aperçu de besoins, ayant chacun des exigences particulières, montrent pourquoi il y a eu prolifération des protocoles, qui sont apparus pour y répondre. Bien entendu, cette liste est loin d'être exhaustive.

Devant cette multiplication, il devient de plus en plus complexe d'analyser le trafic des différents protocoles transitant, pour la plupart d'entre eux, sur le réseau planétaire. L'évaluation d'un protocole, notamment l'étude de ses performances, peut s'effectuer tant au niveau expérimental qu'au niveau exploitation. L'expérimentation permet d'obtenir des résultats quantitatifs à partir d'un modèle théorique simulant, par le calcul, le fonctionnement d'un réseau réel. Le choix du modèle est primordial ; il doit décrire au plus près le fonctionnement des systèmes sur lesquels le protocole sera implémenté, tout en restant suffisamment simple pour être raisonnable en temps de calcul. Il est fortement probable que de nombreuses modifications soient apportées avant d'obtenir des résultats satisfaisants

répondant aux exigences définies pour ce protocole. L'exploitation, précédée ou non d'une étude expérimentale, permet de valider la marche d'un protocole dans des conditions d'utilisation concrètes et réelles, avec tout ce que cela peut contenir d'aléatoire. Toute la difficulté de cette étape réside dans la manière de capturer les paquets. Ils doivent être interceptés en des points précis de la topologie du réseau afin d'être pertinents pour montrer certaines caractéristiques des protocoles étudiés. C'est ce que nous allons mettre en oeuvre tout au long de notre projet.

4 Analyse

4.1 Techniques et méthodes de capture

4.1.1 Mesures actives

Le principe des mesures actives consiste à générer du trafic dans le réseau à étudier et à observer les effets des composants et protocoles – réseaux et transport – sur le trafic : taux de perte, délai, RTT, etc. Cette première approche possède l'avantage de prendre un positionnement orienté utilisateur. Les mesures actives restent le seul moyen pour un utilisateur de mesurer les paramètres du service dont il pourra bénéficier. Il conviendra alors de s'orienter vers ce type de mesure pour mettre en évidence les caractéristiques de la QoS. Les mesures actives simples restent tout de même monnaie courante dans l'Internet pour lequel de nombreux outils de test, validation et/ou mesure sont disponibles. Parmi eux, on peut citer les très célèbres ping et traceroute. Ping permet de vérifier qu'un chemin est valide entre deux stations et de mesurer certains paramètres comme le RTT ou le taux de perte. Traceroute permet de voir apparaître l'ensemble des routeurs traversés par les paquets émis jusqu'à leur destination et donne une indication sur les temps de passage en chacun de ces nœuds. On peut également citer MGEN qui présente la particularité d'émettre des paquets en multicast, c'est-à-dire à destination de plusieurs récepteurs à la fois, et permet donc, grâce à un mécanisme de duplication des paquets au dernier moment dans le réseau, de minimiser le trafic des paquets sondes.

Pour illustrer ce type de mesure, présentons brièvement le logiciel Iperf. Il s'agit d'un outil très utilisé pour tester les réseaux. Il peut créer des flux de données TCP et UDP et mesurer l'influence qu'ils produisent. Différents paramètres peuvent être définis à cet effet. Iperf est constitué d'une partie client et d'une partie serveur qui se disposent aux extrémités du réseau étudié ; il est ainsi possible d'effectuer des mesures entre les hôtes hébergeant ces deux parties.

L'un des inconvénients majeurs pour le réseau avec les mesures actives est la perturbation introduite par le trafic de mesure qui peut faire évoluer l'état du réseau et ainsi fausser la mesure. En effet, le résultat de la mesure donne une information sur l'état du réseau transportant à la fois les données normales des utilisateurs et de signalisation du plan de contrôle du réseau, mais également l'ensemble des paquets sondes. Or on souhaiterait avoir une information qui correspond au trafic normal uniquement, sans les paquets sondes lesquels

ont forcément un impact sur les performances du réseau. Il faut donc, soit être capable d'estimer l'impact des paquets sondes sur les performances du réseau, soit être sûr que ces paquets sondes auront un impact minimal (si possible quasi nul), on parle alors de trafic de mesure non intrusif. C'est cette dernière proposition, a priori plus simple, qui suscite le plus d'efforts de recherche. Ainsi, de nombreux travaux menés actuellement abordent ce problème en essayant de trouver les profils de trafic de mesure qui minimisent les effets du trafic supplémentaire sur l'état du réseau.

4.1.2 Mesures passives

Les projets de mesures passives sont apparus beaucoup plus récemment que les projets de mesures actives car ils nécessitent des systèmes de capture ou d'analyse du trafic en transit relativement avancés.

Le principe des mesures passives est de regarder le trafic et d'étudier ses propriétés en un ou plusieurs points du réseau. L'avantage des mesures passives est qu'elles ne sont absolument pas intrusives et ne changent rien à l'état du réseau. Elles permettent des analyses très avancées. En revanche, il est très difficile de déterminer le service qui pourra être offert à un client en fonction des informations obtenues en métrologie passive. Il vaut mieux pour cela utiliser des techniques actives. Parmi les outils de métrologie passive à notre disposition, il existe tout un ensemble d'outils logiciels, dont il serait illusoire de vouloir faire une liste exhaustive. Nous nous limiterons donc pour cette étude à la plus célèbre de ces familles d'outils logiciels est celle composée de TCPdump/TCPtrace, Wireshark que nous retiendrons pour ce projet, etc. qui sont tous des outils basés sur l'utilisation de la librairie LIBPCAP. Cette librairie permet d'aller lire sur une interface réseau les paquets qui transitent, d'en récupérer une copie, et de la stocker et/ou de l'analyser.

Anciennement appelé Ethereal, Wireshark est un analyseur de protocole réseau qui examine les données à partir d'un réseau en direct ou à partir de fichier disque. Il est possible de naviguer de façon interactive dans les données capturées, de visionner le résumé et l'information détaillée pour chaque ensemble. Souvent, on capture la totalité d'un flux, puis, par la suite, on souhaite se focaliser sur une sous-partie plus précise. Wireshark fourni des outils puissants de filtrage. Le filtre de capture doit être configuré avant de lancer la capture, ce qui n'est pas le cas pour les filtres d'affichage qui peuvent être modifiés à n'importe quel moment pendant la capture. Il est possible de filtrer des données en tenant compte de leur numéro de port, de leur adresse IP source ou destination, etc. Par exemple, le filtre "not ICMP" permet de ne pas filtrer les pacquets ICMP. Il existe aussi une méthode permettant de ne filtrer que les données venant d'un réseau bien spécifié. Bref, tout type de filtrage pouvant être effectué avec Wireshark, nous allons fortement nous en servir dans le cadre de ce projet.

La principale contrainte qui se pose pour l'installation de sonde de mesure consiste à ne pas perturber le fonctionnement des interconnexions, le réseau doit continuer de fonctionner. Le système de mesure ne devra donc pas provoquer de pannes ou d'erreurs de transmission et ne pas introduire de délais pour ne pas modifier le profil du trafic et les performances du réseau. Par ailleurs, le choix des sondes de mesure passive doit garantir la précision et la

validité des traces produites. Ainsi, il est essentiel de ne pas omettre de paquets transitant sur le réseau et d'avoir des informations précises sur le passage de ces paquets, notamment au niveau temporel. Le système devra donc être bien dimensionné et offrir une horloge précise. Enfin, il doit être possible de corréler des événements de plusieurs traces, par exemple de suivre un paquet en plusieurs points du réseau. Pour cela, il faut disposer d'une base temporelle commune à toutes les sondes.

4.2 Techniques et méthodes d'analyse

Une fois les données recueillies, il faut être en mesure de les exploiter afin de comprendre les fonctionnalités des protocoles étudiés. Nous pouvons alors procéder de deux manières. Soit nous définissons des filtres (avec Wireshark par exemple), auquel cas nous récupérons directement un sous ensemble des données qui nous intéresse. Soit, nous pouvons extraire un ensemble de données et appliquer un script qui en effectuera un tri selon certains critères.

4.2.1 Le filtrage

Deux types de filtrage peuvent être appliqués dans un analyseur de réseau, les filtres de captures et les filtres d'affichage.

Les filtres de capture permettent de sélectionner précisément les trames à capturer. Par conséquent, la capture est ciblée sur les informations voulues et la taille des données capturées est limitée, ceci afin d'empêcher la génération de fichiers journaux trop volumineux et de ne conserver que les informations pertinentes. Seules les trames pour lesquelles les filtres sont vrais seront conservées. Ils sont définis avant le démarrage de la capture. Ce genre de filtre prend forme avec une série d'expressions nommées primitives, jointes grâce à des opérateurs logiques. Le filtrage respecte la syntaxe :

Protocole + Direction + Hôte(s) + Valeur + Operations logique + Autre expression.

Le filtre d'affichage est quant à lui bien plus puissant (et complexe), il permet de rechercher exactement les données souhaitées dans des fichiers de capture. De même que dans les filtres de capture, seuls les paquets pour lesquels l'expression du filtre est vrai seront gardés. Les expressions sont basées sur les champs disponibles dans un paquet. On peut aussi utiliser les opérateurs de comparaison pour confronter les champs avec des valeurs. Les expressions ainsi fabriquées peuvent être combinées avec les opérateurs logiques.

4.2.2 Les scripts

Les traces obtenues représentent les données telles qu'elles ont été émise par une entité sur le réseau. Or, ces informations peuvent être difficiles à déchiffrer pour le débutant ou

rébarbative pour l'administrateur réseau qui a besoin d'une information succinte, concrète lui permettant de tirer rapidement des conclusions sur l'état du réseau qu'il a à sa charge. Dans ce but, il est possible d'écrire un script Shell, Python ou Perl afin de répondre à cette attente.

Une fois les communications capturées et stockées dans des fichiers de type pcap, un ou plusieurs traitements automatiques peuvent leur être appliqués. Ils auront pour tâche d'organiser, trier ou supprimer des données de ces fichiers afin de n'en garder que les informations désirées.

5 Conception

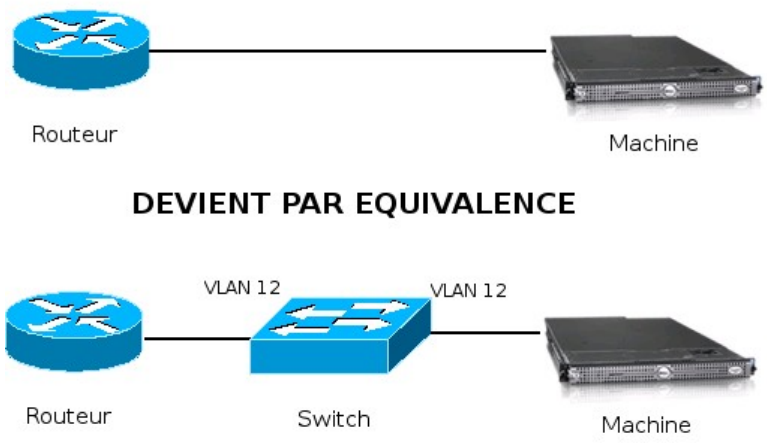
5.1 Architecture de la plate-forme

Théoriquement, capturer un paquet en écoutant l'activité d'un brin de notre réseau consisterait à intervenir physiquement sur ce câble afin d'en écouter le trafic sans pour autant perturber ou ralentir la liaison. Cette approche se révèle rapidement limitée en ce sens que nous ne pouvons "pincer" directement nos interconnexions.

Il s'agit par conséquent de se placer astucieusement entre deux entités communicantes grâce au commutateur en exploitant sa fonctionnalité de redirection de port que nous allons détailler dans les paragraphes qui suivront.

En prenant cette condition de départ, nous avons donc contraint -de manière transparente pour l'activité du brin- chaque cable à transiter par le switch en lui dédiant systématiquement un vlan respectif.

Ainsi, ces deux représentations physiques correspondent à la même topologie logique :



5.2 Le port mirroring

Contrairement aux hubs (concentrateurs), les commutateurs (switchs) prémunissent généralement de l'écoute dite "promiscuous" puisqu'ils redirigent intelligemment les flux d'information sans les dupliquer. Par conséquent, dans un environnement réseau commuté, un sniffer logiciel se limitera donc à capturer le trafic envoyé/reçu par la machine qui l'exploite, ainsi que les émissions de type broadcast et multicast.

Pourtant, la plupart des commutateurs récents supportent la fonctionnalité de redirection de port (port mirroring) -aussi dénommé SPAN chez Cisco- qui consiste pour le commutateur à dupliquer l'activité d'un port A sur un ou plusieurs autres ports, autorisant ainsi les interfaces qui y sont rattachées à capturer le trafic transitant en A sans en perturber son flux.

Il s'agira cependant de ne pas dupliquer plus d'un port lors de la configuration de cette opération afin de s'assurer qu'un flux de paquet n'est pas 'mirroré' plus d'une fois. Dénombrer en double certains paquets tandis que d'autres seraient comptabilisés une fois compliquerait inévitablement nos résultats, amplifiant par la même occasion inutilement le trafic Broadcast et Multicast par le nombre de ports.

Le matériel dont nous disposons nous permet de mettre en place la fonctionnalité du port mirroring sur chacun des commutateurs des AS.

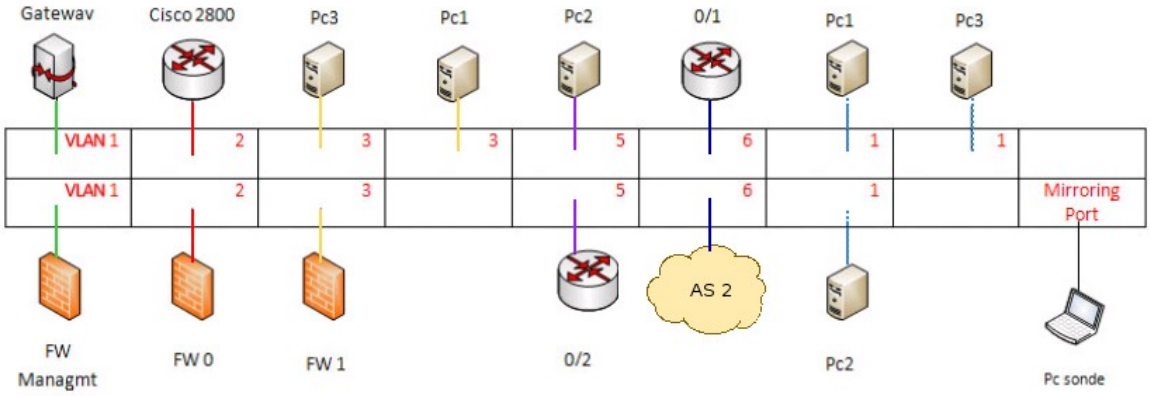
Ainsi pour les switch cisco: on peut copier des interfaces physiques (port) ou virtuelles (vlan).

Dans cet exemple, tout le trafic passant par le port 2 sera copié sur le port 1 :

```
c3560(config)# interface FastEthernet0/1
c3560(config-int)# port monitor FastEthernet0/2
```

Exemple pour le switch HP Procurve 2900 :

```
hp2900(config)# mirror-port ethernet 4
hp2900(config)# interface ethernet 1
hp2900(eth-1)# monitor
```



Topologie physique du switch de l'AS1

Une telle mise en oeuvre provoque l'apparition de nouveaux liens et complique ainsi les éventuelles modifications de topologies ou les manipulations de maintenance.

La phase de réflexion sur la topologie de départ nous impose donc une certaine prudence. Les choix du plan physique se révèlent cruciaux dès la mise en place de la communication de bout en bout de départ, l'architecture globale ne nous laissant que peu de souplesse.

La numérotation des brins avec les étiquettes à disposition réduiront malgré tout les contraintes en cas de force majeur.

5.3 Répartition des taches

Nous avons initialement décidé de nous scinder en deux binômes pour l'administration de la plateforme. Cette répartition perdurera probablement, car elle nous assure une intervention sur deux AS hétérogènes (Cisco/Juniper-Extreme pour le binôme1 et Cisco/Juniper-HP pour le binôme2).

A ce jour, la configuration et l'analyse des protocoles majeurs se répartit comme suit :

S. Benazzouz : Tunneling/Managment
F. Diagne : IPv6/Téléphonie
D. Oriol : QoS/IpSec
F. Signorello : MPLS/Multicast

5.4 Création des traces

5.4.1 Où capturer?

Dans la mesure où l'étude précise des scénarios de capture est prévue dans les semaines qui suivront la rédaction de ce rapport intermédiaire, nous nous contenterons d'évoquer ici une théorie d'approche que nous nous attacherons à mieux justifier dans le rapport final.

Envisageons l'écoute d'un trafic généré par la commande traceroute. Pour mener à bien cette opération, il s'agira d'effectuer cette capture à la source : la machine qui émet le premier paquet icmp (niveau 3). Il conviendra donc de se positionner avec une machine sonde sur son interface de sortie et de n'accepter que les paquets ICMP à l'aide de l'expression icmp.type==30 and ip.addr == (@ de la machine qui émet la requete).

Mais dans le cas d'une application client/serveur, vaut-il mieux sniffer du côté du client ou bien du serveur? A priori les résultats obtenus seront équivalents, mais il serait intéressant de comparer les résultats obtenus, la configuration du port mirroring étant rendue facile.

Il existe tout de même un cas particulier, celui où nous souhaiterions montrer la sécurité offerte par un protocole comme SSH. Il serait intéressant de sniffer dans le réseau pour simuler l'écoute d'un pirate et visualiser ce qu'il peut voir et s'il peut tenter l'attaque man in the middle.

Avec l'étude du multicast, il serait judicieux de coupler l'analyse de l'abonnement aux groupes avec celle d'IPv6, ce dernier gérant cette fonctionnalité nativement. L'observation et la comparaison avec des communications simplifiées unicast permettront de revenir sur les principes de duplication énoncés en cours. Nous capturerons du trafic auprès de l'émetteur pour constater les procédures d'abonnement, puis nous nous focaliserons sur les réseaux coeurs pour observer la génération de l'arbre de routage et les échanges du protocole PIM, pour finalement illustrer des questions soulevées en TD (Multicast ING).

L'étude de la qualité de service, quant à elle, prend son sens dans les variations de trafic, que nous pourrons simuler avec des générateurs de trafic tel qu'Iperf ou Mgen.

5.4.2 Nettoyage et analyse des résultats

Comme vu dans l'analyse, il est apparu nécessaire de filtrer les résultats obtenus lors de notre capture afin d'en retirer l'éventuel surplus d'information. L'écriture d'un script ou d'un programme exploitant les bibliotheques libpcap scrutera notre fichier afin d'y enlever toute redondance et surplus d'information.

Il nous est alors demandé, pour chaque trace, une analyse descriptive et explicative des resultats observés. Pour cette tâche, l'usage d'un script Perl tel que Chaos reader, générant un fichier HTML en sortie, est d'une aide précieuse pour extraire une information claire de la trace concernée. Il ne nous restera plus qu'à en expliquer le contexte par une approche aussi claire et succinte que possible, sans pour autant oublier la vocation pédagogique de cette étude.

6 Etat d'avancement

Dans un premier temps nous avons découvert les baies mises à notre disposition sur la plateforme. Tous les matériels étaient interconnectés, ce qui nous a permis d'observer afin de nous donner un exemple de ce qui pouvait être fait. Une réunion a d'ailleurs été organisée avec les anciens étudiants de l'année précédente afin de recueillir leurs remarques et leurs suggestions.

La documentation fut longue et continuera probablement tout au long du projet. Nous tenons pour essentielles références les importantes ressources mises à disposition par les constructeurs pour la gestion du matériel, et les moteurs recherches dédiés aux université (Sudoc, Google Scholar...) pour affiner nos approches théoriques.

Nous avons par la suite déconnecté puis réinitialisé chacun des appareils avec leur configuration d'usine afin de pouvoir y implémenter notre propre contexte d'interconnexion.

Une fois la topologie et le plan d'adressage, prévu pour être flexible, définis, nous avons établi sur les matériels une configuration minimale afin de simplifier la première communication de bout en bout : installation de routage statique, de règles simplifiées dans les pare-feu et choix des futurs systèmes d'exploitation sur les serveurs.

A ce jour, nous avons implémenté le protocole BGP entre les système autonomes, RIP ou OSPF dans chaque AS.

Pour terminer nous avons rendu nos matériels accessibles de l'extérieur en reliant la passerelle SSH administrée par le LIP6 aux VLAN d'administration de chacun des commutateurs.

Ces premières manipulations nous offrent une base solide pour faire évoluer progressivement la configuration des noeuds qui sera à même d'accueillir les protocoles plus complexes comme MPLS ou le Multicast.

L'ensemble de nos travaux sont tenus à jour sur le Trac mis en place par l'équipe de l'année préxédente [https://www-rp.lip6.fr/trac/pfres] et le Google Group plateforme6 [http://groups.google.fr/group/plateforme6] afin de synchroniser nos interventions.

Références

[JAC 89] JACOBSON, V. , LERES C., MAC CANE S., *libpcap*, software (latest release : version 0.4), 1989

[CUN 95] CUNHA C. R., BESTAVROS A., *Characteristics of WWW Client-based Traces*, rapport n° BU-CS-95-010, Computer Science Department, Boston University, 1995

[FEL 98] FELDMANN A., *Continuous online extraction of HTTP traces from packet traces*, Position paper for the W3C Web Characterization Group Workshop, 1998

[BAY 00] BAYNAT, B., *Théories des files d'attente, des chaînes de Markov aux réseaux à forme produit*, Hermes Science Publications, 2000

[LAR 02] LARRIEU N., *Métrologie des réseaux IP : développement de nouveaux outils pour caractériser, analyser et rejouer le trafic réseau*, rapport de diplôme ingénieur INSA, 2002

[STR 03] STRAUSS J., KATABI D., KAASHOEK F., *A Measurement Study of Available Bandwidth Estimation Tools*, ACM SIGCOMM Internet Measurement Workshop, 2003

[HU 03] HU N., STEENKISTE P., *Evaluation and Characterization of Available Bandwidth Probing Techniques*, IEEE Journal on Selected Areas in Communication, n°21, 2003

[LAB 05] LABIT Y., OWEZARSKI P., LARRIEU N., *Evaluation of active measurement tools for bandwidth estimation in real environment*, 3rd IEEE/IFIP Workshop on End-to-End Monitoring Techniques and Services, Nice (France), 2005

Annexe

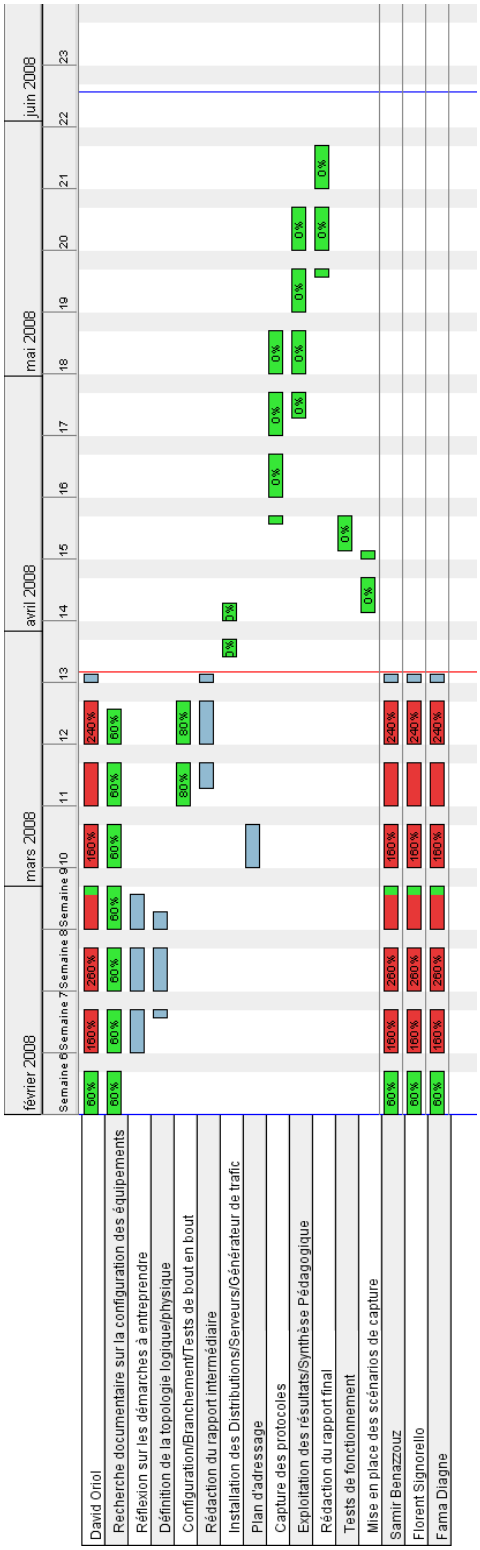


Diagramme de ressources