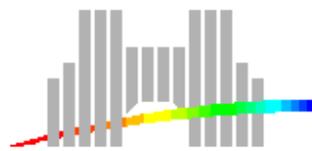


EWAN

Émulateur de nuage réseau de grille



Sous la tutelle de
Pascale Vicat-Blanc Primet et Olivier Glück (LIP)
et Alexandre Saidi (ECL)

Avril-Septembre 2004

Résumé

Ce rapport décrit mon travail de fin d'études effectué au Laboratoire d'Informatique et Parallélisme de l'École Nationale Supérieure de Lyon dans l'équipe RESO, consistant en la réalisation d'EWAN, un logiciel d'émulation de réseaux haut débit, longue distance.

Il se place dans le cadre de la recherche sur les grilles de calcul, agrégations de ressources informatiques interconnectés par un réseau longue distance.

Il offre des possibilités d'émulation nécessaires aux expériences dans ce domaine, en assurant une grande flexibilité, un haut niveau de performances, et un environnement contrôlé avec précision. Il se décompose principalement en deux parties : une interface de création de topologies simples, et un moteur de déploiement de topologies quelconques.

Ce document présente le contexte du travail, et les outils qui seront utilisés. Puis le fonctionnement des différentes parties du logiciel seront décrites et expliquées. Les résultats de tests seront reportés, confirmant la réussite des objectifs fixés, et des perspectives évoqueront les améliorations à apporter, et les travaux futurs.

Abstract

This report describes my end of studies training in the Laboratoire d'Informatique et Parallélisme of École Nationale Supérieure de Lyon in the RESO team, which consisted in developing a software for emulation of high bandwidth , long distance networks .

It takes place in the research effort on computer grids, aggregations of computer resources inter-connected by a wide area network

It offers an emulation framework needed by experiments in this field, bringing a great flexibility, a high level of performances, and a precise control. It can be divided into two main parts : an interface for creation of simple topologies, and an engine deploying every sort of topologies.

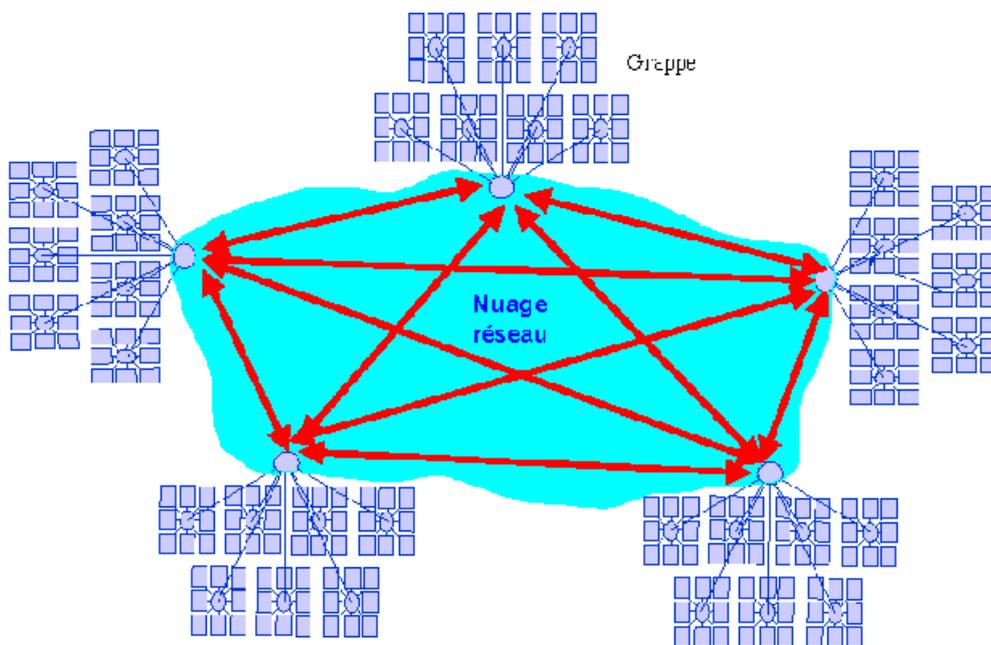
This document presents the context of this work, and the tools which will be used. Then the different parts of the software will be described and explained.

Experimental results will be given, proving the success of primary objectives, and possible improvements to this software and future works will be mentioned.

Introduction et contexte

Le logiciel EWAN réalisé pendant mon travail de fin d'études s'inscrit dans l'effort de recherche dans le domaine des grilles de calcul qui s'appuie sur plusieurs projets français ou européens de grilles expérimentales.

Une grille est un système informatique réparti à grande échelle, le plus souvent constitué d'un nuage réseau au cœur, et de grappes en périphérie.



EWAN a pour but de permettre à un utilisateur de tester son application, ou son protocole dans un environnement émulé et parfaitement maîtrisé, avant de le déployer sur des grilles de calcul réelles.

En effet, pour réaliser ses expériences, le chercheur a le choix entre la simulation (qui atteint vite ses limites, en terme de taille ou de durée), ou des tests grandeur nature. Ces derniers sont longs voire impossibles à mettre en place, et sujets aux aléas des expériences réelles, ce qui rend leur reproduction délicate : par exemple, les paramètres du réseau dont dépendent fortement les résultats des expériences ne sont pas contrôlables (taille des paquets, qualité de service...)

Il faut alors se tourner vers l'émulation, approche intermédiaire où une

partie des éléments sont réels (applications et nœuds d'extrémité), et une partie simulée (les liens longue distance), le tout dans un environnement totalement maîtrisé.

EWAN est donc un instrument orienté sur l'étude des protocoles et des logiciels haute performance pour la grille avec un très grand nombre de calculateurs interconnectés. Il doit offrir un haut niveau de performance, une fine maîtrise des paramètres de communication associé à une grande flexibilité d'utilisation.

A. Présentation du Laboratoire de Recherche

Mon travail de fin d'études s'est déroulé au LIP (Laboratoire de l'Informatique du Parallélisme), le laboratoire d'informatique de l'ENS Lyon également associé au CNRS, à l'INRIA et à l'Université Claude Bernard de Lyon.

Ce laboratoire est constitué de sept projets : Arénaire (Arithmétique des ordinateurs), Compsys (Compilation, systèmes enfouis et calcul intensif), MC2 (Modèles de calcul et de complexité), PLUME (Utilisation et amélioration de la déduction automatique), GRAAL (Algorithmique et ordonnancement pour plates-formes hétérogènes distribuées), TRIO (Temps Réel et InterOpérabilité) et RESO (Logiciels et protocoles optimisés pour réseaux haut-débit).

C'est dans cette dernière équipe, sous la tutelle de Pascale Vicat-Blanc Primet et Olivier Glück que j'ai effectué mes travaux.

Le projet RESO étudie le problème de l'utilisation efficace des réseaux très haut débit dans le contexte des grilles haute performance (qui seront définies ci dessous).

Ses travaux recouvrent deux axes :

- La conception d'architectures logicielles de communication optimisées pour les systèmes d'extrémité ou d'accès multiprocesseurs ou de type grappe.
- Les protocoles et traitements pour le transport performant de flux hétérogènes de grille.

Pour cela, l'équipe développe des outils notamment dans le domaine de l'émulation, comme EWAN.

B. Contexte

I. Grilles de calcul et topologies

Une grappe de machines (*cluster* en anglais) est un groupe d'ordinateurs mis en réseau local, afin de former un ensemble de forte capacité, très utilisé pour les calculs parallèles.

Une grille est un ensemble d'ordinateurs distribués géographiquement reliés par des réseaux longue distance (Internet, réseaux privés virtuels (VPN), ou réseaux privés réels), qui étend le principe du calcul parallèle des grappes.

Les grilles de calcul permettent une agrégation de ressources informatiques variées (comme la puissance de calcul ou de stockage de données).

Les premières grilles connues du grand public furent les logiciels de calcul distribué comme *SETI@Home* ^[1] puis *Genome@Home* ^[2] pour lequel les ordinateurs des particuliers travaillaient et envoyaient au serveur le résultat de leurs efforts.

Puis les grilles se sont affranchies du modèle *Client -Serveur* afin que chaque ordinateur puisse lancer des calculs ou utiliser la puissance des autres. Dans le but d'obtenir du calcul parallèle toujours plus puissant, des grilles regroupant plusieurs grappes furent créées.

Actuellement, la recherche dans ce domaine prometteur avance notamment grâce à la mise en place de plateformes expérimentales s'appuyant sur des réseaux haut débit (comme RENATER, réseau national regroupant des établissements scientifiques et universitaires, ou VTHD, premier réseau très haut débit français). Par exemple, Grid'5000 ^[3] est un projet de réalisation d'une grille expérimentale destinée à la communauté française des chercheurs en informatique, regroupant 8 sites en France dont un à Lyon impliquant le LIP. Les chercheurs seront invités à tester leurs expériences avec EWAN avant de les déployer sur Grid'5000.

Les travaux du Global Grid Forum ^[4] (organisme de standardisation dans le domaine des grilles de calcul) ont permis d'obtenir une caractérisation très complète des topologies réseau, servant à décrire la structure des grilles. ^[5]

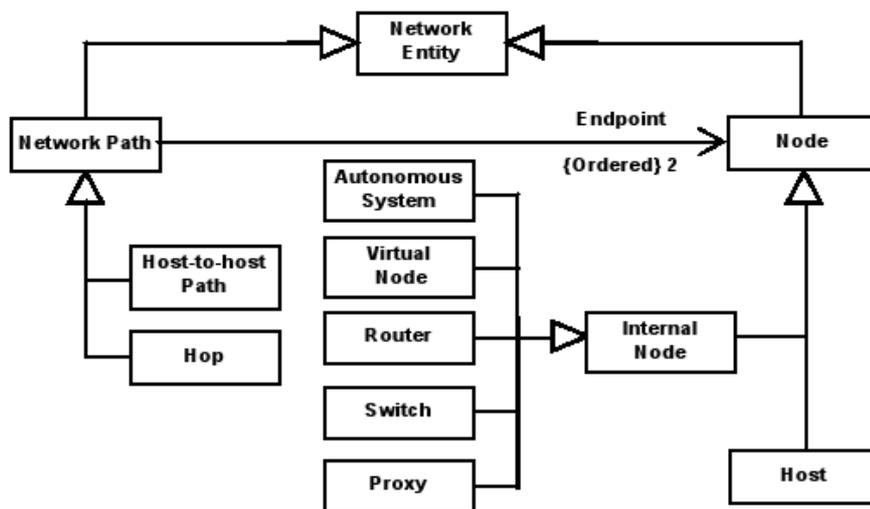
1. Entités

Un réseau est composé de nœuds et de liens (unidirectionnels ou

bidirectionnels).

Les nœuds peuvent être des clients (Host) ou des nœuds internes (routeurs, switches, proxy)

2. Propriétés



Les principales propriétés des liens (qui ont été étudiées pendant le stage) sont regroupées en 4 catégories :

La Bande passante, la Latence, les Pertes, et les Duplications.

a. Bande passante

La bande passante se mesure généralement en quantité de données par unité de temps.

Cependant, elle ne peut être représentée par une seule valeur : on introduit donc 4 caractéristiques :

La Capacité

C'est la quantité maximale de données pouvant être transmise, lorsqu'il n'y a pas d'autre trafic.

C'est une valeur théorique, et la seule que l'on connaît précisément.

L'Utilisation

C'est la quantité de données concrètement transmises à un instant donné.

La Bande Passante disponible

C'est la quantité maximale de données qu'un lien peut fournir lors d'une certaine utilisation.

La Bande Passante atteignable

C'est la quantité de données maximale pouvant être transmise selon une certaine utilisation, un certain protocole, et certains équipements.

C'est en fait la valeur la plus intéressante à étudier, car la seule réellement utile. Malheureusement, elle ne peut pas être formellement calculée, car dépendante de trop de paramètres.

On a :

Utilisation < Bande Passante atteignable < Bande Passante disponible < Capacité

Le but est de trouver les paramètres permettant d'approcher le plus possible la Bande Passante atteignable de la Bande Passante disponible.

b. Latence

Les caractéristiques de la Latence sont décrites en détail dans les RFC 2330, 2679 et 2681 ^[6].

Elle représente le temps entre le moment où le début d'un paquet part d'un émetteur et celui où la fin du paquet arrive au récepteur.

Cette définition entraîne des interrogations sur les problèmes de synchronisation, de fragmentation de paquets...

La Latence, (comme les autres grandeurs) peut être mesurée en aller simple ou en aller-retour. La mesure de l'aller simple est très importante, car de nos jours la plupart des liens sont asymétriques (ils peuvent par exemple avoir des qualités de service différentes selon le sens).

La Gigue

Classée dans la famille des latences, également appelée IPDV (IP Packet Delay Variation), elle mesure les variations de la Latence dans le temps.

La Gigue a des conséquences importantes pour les applications qui

nécessitent un flux régulier (comme le streaming audio ou vidéo).

c. Pertes

Les pertes représentent les paquets émis qui ne sont pas arrivés à destination.

Cette métrique est définie par la RFC 2680 ^[7].

Les pertes peuvent être caractérisées par un pourcentage de paquets perdus, ou par un profil de perte.

Dans ce dernier cas, seul un profil particulier (Loi de Poisson) a été formellement défini (dans la même RFC), mais l'Internet Society encourage des développeurs à utiliser également d'autres profils.

Le logiciel qui sera utilisé pour l'émulateur durant le stage (NIST Net) ne permet pas d'implémenter le profil de Poisson. En revanche, il propose deux autres profils : corrélation (c'est à dire une valeur moyenne associée à un coefficient de corrélation) et DRD (Derivative Random Drop) ^[8].

d. Duplications

Les duplications représentent les paquets qui arrivent plusieurs fois à destination.

On peut les mesurer en pourcentage, ou suivant un profil. NIST Net permet d'utiliser un profil de corrélation.

Un diagramme représentant la totalité des propriétés définies est donné en Annexe [1].

II. Outils

La première idée envisagée peut être d'utiliser un simulateur, comme NS ou OpenNet. Mais de part leur nature, ces programmes ne permettent pas des mesures expérimentales réelles.

Plusieurs outils complets d'émulations existent déjà (comme Emulab ^[9], X-bone ^[10], ou ModelNet ^[11]), mais ils n'étaient pas adaptés aux objectifs d'EWAN, notamment en ce qui concerne le haut débit.

D'autres outils se concentrent sur certains aspects précis de l'émulation (notamment pour les liens). EWAN n'a pas la prétention de récrire ces briques fondamentales : elles lui serviront de bases et son rôle sera de les paramétrer

correctement.

1. NIST Net

NIST Net ^[12] est un logiciel (sous licence libre basée sur la GPL) développé par le NIST (National Institute of Standards and Technology). Il se présente sous la forme d'un module du noyau linux.

Son but est d'émuler un lien. On souhaite recevoir des paquets par une interface, et les transmettre en sortie sur une autre, selon certaines conditions correspondant aux caractéristiques du lien.

NIST Net se place juste avant les fonction IP du noyau. Il va récupérer les paquets reçus, les stocker dans des files d'attente, et les libérer selon les paramètres précisés lors de sa configuration.

NIST Net peut ainsi émuler des latences (y compris des gigue), des pertes (par simple pourcentage ou selon deux profils : corrélation ou DRD), des duplications (par pourcentage ou selon le profil corrélation), et le débit.

Le profil DRD (Derivative Random Drop) correspond à une probabilité de pertes qui croît linéairement avec la taille de la queue, après un éventuel seuil.

Il existe un autre profil couramment utilisé, RED (Random Early Detection), mais les concepteurs de NIST Net ont estimé que dans le cadre de l'émulation de liens, DRD était amplement suffisant. En effet, RED (défini dans la RFC 2309) est un algorithme utilisé dans la gestion des queues des routeurs (pour déterminer la façon de jeter les paquets lorsque la queue est pleine).

L'émulation du débit par NIST Net ne sera pas utilisée, car elle n'est pas très précise, et car nous disposons d'un autre outil très simple mieux adapté : TC.

NIST Net dispose d'une interface graphique et d'une autre en ligne de commande. Vu que nous prévoyons de déployer des configurations sur plusieurs machines d'un *cluster*, seule cette dernière sera utilisée.

Packet source and destination addresses
(default matches all otherwise unmatched)
Either names or IP addresses may be used.

Mean and standard deviation of
delay times in milliseconds

Maximum allowed bandwidth
in bytes/second

Percentage of packets
dropped and duplicated

Source	Dest	Delay (ms)	Dev sigma (ms)	Bandwidth	Drop %	Dup %	DRI
default	default	0.000	0.000	0	0.0000	0.0000	
japin.antd.nist.gov	default	0.000	0.000	0	0.0000	0.0000	
naga.antd.nist.gov	japin.antd.nist.gov	0.000	0.000	0	0.0000	0.0000	
raisinet.cs.umd.edu	default	20.000	1.974	0	0.0000	0.0000	
naga.antd.nist.gov	raisinet.cs.umd.edu	0.000	0.000	30000	0.0000	0.0000	
itg.antd.nist.gov	snad.ncsl.nist.gov	0.000	0.000	0	4.9968	0.0000	
japin.antd.nist.gov	naga.antd.nist.gov	0.000	5.000	0	0.0000	0.0000	
		0.000	0.000	0	0.0000	0.0000	
		0.000	0.000	0	0.0000	0.0000	

Turn kernel emulator on and off

Read current kernel
emulator settings

Load changed settings
into kernel emulator

Add another row to
the user interface

Quit the user interface
(kernel emulator is not affected)

NIST Net stocke les paquets en attente dans des tables, dont les tailles sont donc conditionnées par la mémoire de la machine utilisée. Mais la facteur vraiment limitant selon diverses études de performances est le CPU. En effet, NIST Net doit régulièrement parcourir ces tables afin de stocker les paquets arrivant. Ainsi, plus la latence que l'on souhaite émuler est élevée, plus les tables seront longues, et plus le CPU devra être puissant.

2. Dummynet

Dummynet ^[13] est un outil sous licence BSD pour les plateformes FreeBSD.

Il permet également d'émuler des latences, des effets de queues, des limitations de débits et des réordonnements de paquets, en interceptant le trafic dans la pile protocolaire de la machine cliente.

Les machines manipulées durant le stage tourneront sous GNU/Linux, aussi c'est NIST Net qui sera utilisé au lieu de Dummynet.

Un port de ce programme pour linux avait débuté dans une équipe de recherche parisienne, mais suite à l'engagement des travaux du LIP sur l'émulation avec NIST Net, ce développement fut arrêté.

3. TC

TC (Trafic Control) est une commande standard de GNU/Linux, permettant notamment de limiter le débit en sortie d'une interface.

C'est l'outil qui sera utilisé pour émuler le débit d'un lien.

Mais TC peut également servir à définir des classes de service, qui pourront être utiles dans l'émulation de Qualité de Service.

Réalisation

A. Spécifications

EWAN signifie simplement Émulateur de WAN (Wide Area Network : Réseaux informatiques de grande taille) et a comme objectif d'être un émulateur de topologies haut débit, longue distance.

C'est à dire qu'EWAN doit configurer des machines liées entre elles par du haut débit (1Gb/s), et émulant des liens longue distance (donc en utilisant l'émulateur de lien NIST-Net).

EWAN devait se décomposer en briques :

- Une interface web générée en PHP qui permettrait de définir intuitivement des topologies simples.

- Un algorithme d'affectation des types de fonctionnalités aux machines de la grappe.

- Un algorithme de détermination des IP et sous-réseaux.

- Un algorithme de calcul de routage et de configuration des tables de routage.

- Une méthode de stockage des informations dans une base MySQL

- Et un procédé de déploiement sur la grappe.

Nous avons ensuite réalisé que PHP pouvait convenir pour toutes les briques, aussi ce langage fut retenu pour EWAN.

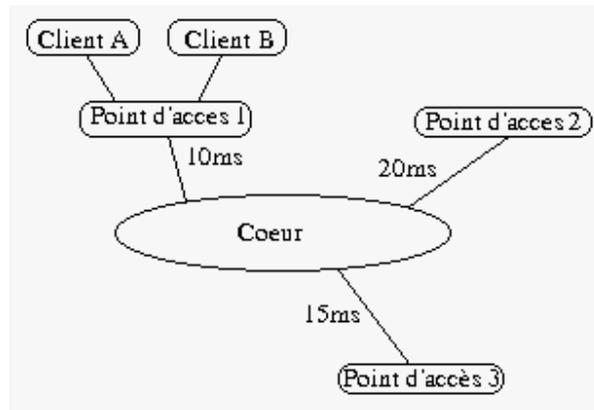
L'interface est prévue pour des expérimentateurs qui ne s'intéressent pas au cœur du réseau supposé surdimensionné, et qui veulent réaliser simplement une topologie classique telle qu'ils en rencontrent dans la réalité sans avoir à se préoccuper de sa structure interne (comme le routage).

B. Architecture logique

EWAN doit donc permettre de simuler des topologies au niveau 3 de la couche OSI, c'est à dire des nœuds reliés par des liens avec des métriques à définir, dans des sous réseaux IP.

La cas de figure le plus couramment étudié est celui de clients derrière des

points d'accès reliés par des liens (aux latences non négligeables) à un cœur de réseau surdimensionné.



On peut alors mettre en évidence certaines fonctionnalités qui devront être émulées :

- émulation de lien (latence, pertes et duplications)
- émulation de capacité (limitation de débit: token bucket)
- routage virtuel
- génération de trafic concurrent
- capture de trafic concurrent

Ces fonctionnalités seront réparties selon 5 types émulés par les nœuds :

- Client
- Point d'accès
- Routeur de cœur
- Émulateur de lien
- Générateur de trafic

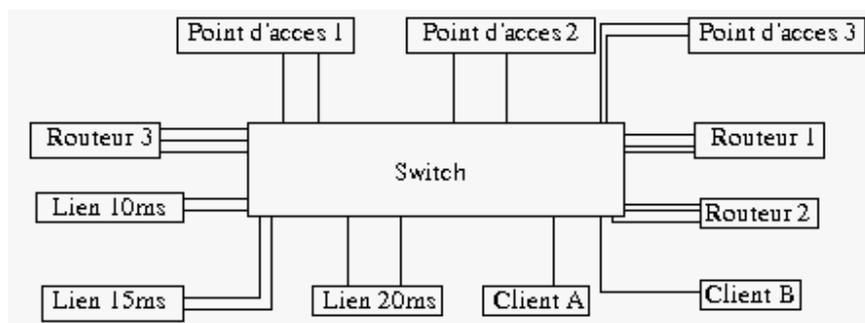
C. Architecture physique

EWAN est constitué d'une grappe de nœuds non nécessairement matériellement identiques (avec par exemple un nombre d'interfaces, une capacité mémoire, ou CPU différents), reliés par un ou plusieurs commutateurs (*switchs*) et d'un serveur qui s'occupera de configurer ces nœuds et sur lequel EWAN s'exécute.

Ces machines appartiennent donc initialement au même sous-réseau et sont accessibles entre-elles au niveau 2 de la couche OSI. (Ethernet)

Chaque nœud utilisé pour la topologie souhaitée sera de l'un des types prédéfinis et émulera une fonctionnalité :

- La latence, les pertes et les duplications sont assurées par les émulateurs de liens (utilisant NIST Net).
- Les limitations de débits sont mises en œuvre par les nœuds aux extrémités du lien à limiter.
- Le routage virtuel est réalisé par les routeurs de cœur et les points d'accès.
- La génération de trafic est la fonctionnalité spécifique des générateurs.
- La capture du trafic concurrent est à la charge des routeurs (de cœur ou d'accès)



Pour cela, tous les nœuds de la grappe doivent posséder les mêmes logiciels suivants :

- Debian GNU/Linux
- iproute
- au moins un générateur de trafic (Iperf ou Mgen)

- un serveur SSH
- un kernel compilé avec les options de routage avancées (IP: advanced router et IP: policy routing) et CPU (/dev/cpu/*/cpuid - CPU information support).

Par ailleurs, toute la configuration de la grappe et son déploiement sont gérés par un serveur web, qui doit disposer de :

- Apache
- PHP 4 ou PHP 5 avec les bibliothèques :
 - MySQL
 - XML et DOM
 - GD (avec PNG et FreeType)
- MySQL
- SSH

Le logiciel EWAN, réalisé en PHP, comporte deux principales parties : une interface de création de topologies classiques, et un moteur de déploiement (qui accepte n'importe quelle topologie)

D. Interface de création de topologie

En se connectant sur la page d'accueil d'EWAN, 4 choix sont possibles : charger une topologie, définir une nouvelle topologie, gérer les configurations matérielles et vérifier la configuration matérielle actuelle.

1. Charger une topologie.

Ce champ permet de charger une topologie définie préalablement, à partir d'un fichier RDF. Ces fichiers sont créés soit à partir de l'interface, soit à la main en respectant leur syntaxe. Celle-ci est détaillée dans la partie Déploiement des topologies.

Émulation de réseaux haut débits longue distance

Charger une topologie

Browse .

Valider

Définir une nouvelle topologie

Gestion des configurations matérielles

Vérification de la configuration matérielle

2. Définir une nouvelle topologie.

Ce lien conduit à l'interface qui va guider pas à pas la création d'une topologie classique, c'est à dire constituée de points d'accès reliés à un cœur surdimensionné.

À chaque étape, les choix offerts à l'utilisateur sont calculés à partir des informations qu'il a déjà saisies et des contraintes matérielles.

a. Points d'accès et Clients

Cette page détermine le nombre de points d'accès dans la topologie, ainsi que le nombre de clients pour chacun d'eux. Les limites sont déduites de la configuration matérielle utilisée (voir 3. Gestion des configurations matérielles).

Interface de configuration

Points d'accès et Clients

Nombre de points d'accès : (au plus 5)

Nombre de machines clients pour le point d'accès 0 :

Nombre de machines clients pour le point d'accès 1 :

Nombre de machines clients pour le point d'accès 2 :

Nombre de machines clients pour le point d'accès 3 :

(5 clients sur 7 possibles)

b. Définition des liens

Ici sont calculées les caractéristiques des liens.

Interface de configuration

Définition des liens

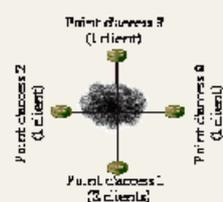
Choisissez le point d'accès le plus éloigné :

Lien C → 1 :
Latence :
Pertes :
Duplications :

Lien C → 2 :
Latence :
Pertes :
Duplications :

Lien C → 3 :
Latence :
Pertes :
Duplications :

Choisissez le lien le plus contraignant parmi ceux qui relient : de : à
Lien 1 → 2 :
Latence : (entre 40 et 60);
Pertes :
Duplications :



- Nombre de points d'accès : 4
- Nombre total de clients : 5

© ENS Lyon

Les liens qui nous intéressent ici, sont ceux liant les points d'accès aux routeurs de cœur.

En effet, les liens entre un point d'accès et ses clients correspondent à un réseau local, donc sont négligeables, et les liens à l'intérieur du cœur sont supposés surdimensionnés.

Les utilisateurs auxquels l'outil est destiné ne connaîtront habituellement pas les caractéristiques des liens Points d'accès - Routeur de cœur. Au lieu de ça, ils disposent d'informations mesurées de bout en bout, c'est à dire d'un point d'accès à un autre. Ils peuvent posséder alors les caractéristiques d'un graphe complet, soit $n(n-1)/2$ valeurs pour n Points d'accès.

On se retrouve avec un système de n inconnues à $n(n-1)/2$ équations.

Si l'utilisateur devait rentrer toutes ces valeurs, on aurait un risque d'incohérence.

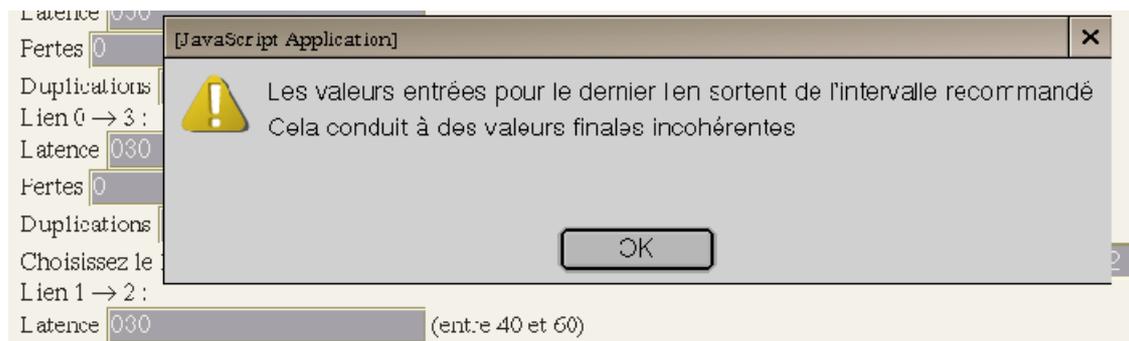
C'est pourquoi, l'utilisateur doit tout d'abord choisir le point d'accès le plus éloigné du cœur (selon la métrique la plus intéressante). Puis il renseigne la latence, les pertes et les duplications entre ce point d'accès et chacun des autres. (Soit $n-1$ valeurs).

Puis, il faut encore choisir un couple de point d'accès (en excluant le premier) et donner les mêmes renseignements.

Les valeurs données pour les liens du premier point d'accès entraînent un intervalle de cohérence pour le dernier lien qui est rappelé en temps réel.

(En effet, si le lien de A vers B a une latence de 100ms, et celui de A vers C 20ms, le lien entre B et C doit forcément se trouver dans l'intervalle 80-120ms,

sous peine d'obtenir des latences négatives entre un point d'accès et le cœur)



c. Limitation des débits

Il est possible de limiter le débit au sein d'un réseau local d'un point

Interface de configuration

Voulez vous limiter les débits ?

Sur le lien relié au point d'accès 0 ? Débit : 1000 (Mb)
Sur son réseau local ? Débit : 000 (Mb)

Sur le lien relié au point d'accès 1 ? Débit : 500 (Mb)
Sur son réseau local ? Débit : 000 (Mb)

Sur le lien relié au point d'accès 2 ? Débit : 1000 (Mb)
Sur son réseau local ? Débit : 00 (Mb)

Sur le lien relié au point d'accès 3 ? Débit : 1000 (Mb)
Sur son réseau local ? Débit : 000 (Mb)

d'accès, ou sur le lien du point d'accès au cœur.

d. Choix de la topologie

Le cœur peut être constitué d'un simple routeur central (topologie en étoile) ou d'un ensemble de routeurs (topologie en anneau).

Ce sont les seules topologies qui permettent d'émuler un cœur surdimensionné sans être trop gourmand en routeurs.

Si un utilisateur souhaite utiliser une topologie différente, il faut qu'il la crée directement, sans passer par l'interface.

Interface de configuration

Choix de la topologie

Choisissez la topologie du cœur :

Anneau
Etoile

Valider

Revenir à l'étape précédente

- ◆ Nombre de points d'accès : 4
- ◆ Nombre total de clients : 5

Il se peut que l'une de ces deux topologies ne soit pas accessible, du aux contraintes de la configuration matérielle.

e. Cross Traffic

S'il reste des machines de libre, elles peuvent servir de générateur de cross traffic, qui va congestionner une interface en sortie. Sur la page apparaît une représentation de la topologie. L'utilisateur peut choisir, en cliquant dessus, les interfaces qu'il souhaite congestionner. Il peut alors indiquer le débit du générateur dans le champ correspondant qui apparaîtra.

Dans la création de ces topologies simplifiées, on ne définit pas de puits pour le cross traffic (Une machine qui reçoit ce trafic et le jette. Cette opération est très peu coûteuse).

Comme nous le verrons plus tard, le moteur d'EWAN est capable d'en affecter un par défaut.

Interface de configuration

```

graph TD
    A[Point d'accès 0] ---|eth0-eth1| B[Point d'accès 1]
    B ---|eth1-eth0| C[Point d'accès 2]
    C ---|eth0-eth1| D[Point d'accès 3]
    D ---|eth1-eth0| A
    
```

Cross Traffic

Il reste 5 machines à éventuellement affecter au cross traffic

Interface : Routeur 0 > Routeur 3 : débit : = Facteur de la interface

- Nombre de points d'accès : 4
- Nombre total de clients : 5
- Topologie : Anneau

© ENS Lyon

f. Paramètres avancés optionnels

Pour finir, il reste 3 paramètres à préciser : La présence de qualité de service, la version du protocole IP, et la taille des trames utilisées. Les valeurs par défaut de ces 3 critères sont les plus communes.

Interface de configuration

Paramètres avancés optionnels

Qualité de Service : Oui Non

Version du protocole IP : IPv4 IPv6

MTU : Frames (standard) Jumbo Frames

Valider

Revenir à l'étape précédente

g. Fin de la configuration.

Cette page indique que la configuration de la topologie est terminée. Il est possible de sauvegarder son fichier de définition pour une réutilisation ultérieure, ou pour le modifier ponctuellement.

Puis si l'utilisateur veut déployer immédiatement cette topologie, il peut débiter le processus : Voir Déploiement des topologies

Interface de configuration

Configuration terminée

Vous pouvez sauvegarder le fichier de configuration, puis, lancer la procédure.

Ou revenir à l'accueil.

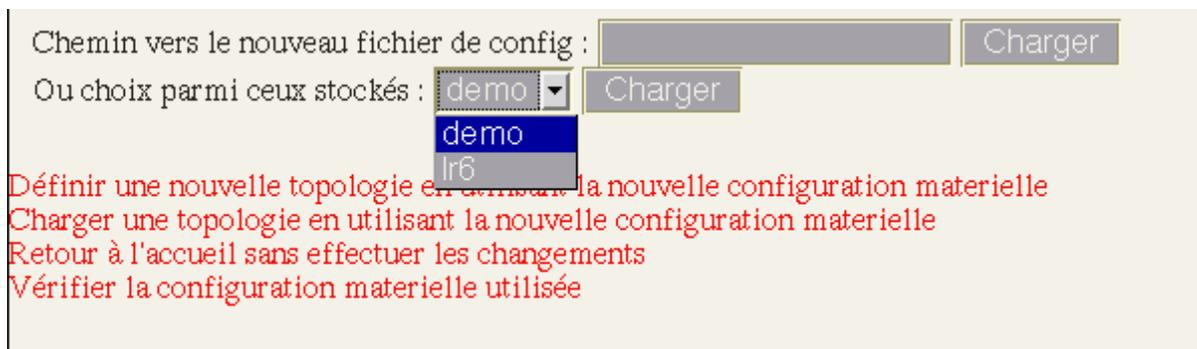
3. Gestion des configurations matérielles

La configuration matérielle est stockée dans le fichier `config.php`, dont le format est défini en Annexe [2].

Il est possible de créer d'autres fichiers de configuration matérielle, et de choisir d'utiliser ceux-ci plutôt que celui par défaut grâce à la page Gestion des configurations matérielles.

Il suffit d'indiquer le chemin du fichier choisi.

Si des fichiers de configurations sont déjà stockés dans le dossier `configmat`, ils seront proposés.



Chemin vers le nouveau fichier de config :

Ou choix parmi ceux stockés :

[Définir une nouvelle topologie en utilisant la nouvelle configuration matérielle](#)

[Charger une topologie en utilisant la nouvelle configuration matérielle](#)

[Retour à l'accueil sans effectuer les changements](#)

[Vérifier la configuration matérielle utilisée](#)

4. Vérification de la configuration matérielle

Cette page permet de vérifier la configuration matérielle utilisée (définie par le fichier `config.php` ou chargée dynamiquement par la fonction précédente). Si certaines machines ne sont pas accessibles par SSH, elles seront retirées de la configuration en mémoire, et ne seront donc pas utilisées par EWAN. Pour les autres, le script vérifie en plus certaines caractéristiques (CPU et nombre d'interfaces) et met à jour la configuration utilisée.

Vérification de la configuration matérielle

```
140.77.15.128
  Disponible
140.77.15.127
  Disponible
140.77.15.121
  Disponible
140.77.15.122
  Disponible
```

Définir une nouvelle topologie en n'utilisant que les machines disponibles
Retour à l'accueil sans effectuer les changements

E. Déploiement des topologies

La partie d'EWAN dédiée au déploiement des topologies utilise en entrée une définition de topologie, provenant soit directement de l'interface, soit d'un fichier de définition, puis va calculer les différents paramètres de configuration des nœuds de la grappe, et finalement va les déployer et mettre à jour ces machines.

1. *Format de définition des topologies*

Comme nous l'avons vu précédemment, une topologie est composée de nœuds et de liens, ayant chacun certaines propriétés.

Ce sont ces informations qu'il faut stocker dans le format de sauvegarde des topologies.

À des fins d'interopérabilités, nous avons retenu le langage RDF/XML ^[14] défini par le W3C.

En effet, et sans rentrer dans les détails qui sortent du cadre de ce mémoire, contrairement aux fichiers en simple XML où un parseur spécifique doit être utilisé pour "comprendre" leur contenu, n'importe quel parseur RDF suffit pour en retirer le sens. Par exemple, la portion de XML `<nœud><point-acces /></nœud>` indique-t-elle un nœud qui est un point d'accès, ou un nœud qui a un

point d'accès ? En RDF l'ambiguïté est levée (ici par exemple, si l'élément *nœud* correspond à une ressource *point-acces* est une propriété, ce qui se traduit par "le nœud a un point d'accès").

Un schéma RDF décrivant une nouvelle ontologie a alors été mis au point, pour préciser les classes, représentant principalement les entités de la topologie (par exemple nœud, Lien, Routeur, Client...), et les propriétés (débit, latence...)

Il est disponible à l'adresse <http://purl.org/net/topo/ns/> , qui est d'ailleurs le namespace de cette ontologie, et redonné en Annexe [3].

Ce format permet la description de n'importe quelle topologie de niveau 3, c'est à dire ensemble de nœuds (clients, routeurs ou points d'accès) reliés par des liens de façon quelconque.

Ainsi, EWAN n'est pas limité aux topologies simples produites par l'interface (comme c'est le cas pour d'autres produits), mais peut au contraire émuler toutes les topologies envisageables.

2. Exécution du déploiement

Le processus se décompose en deux phases : la génération des scripts pour chaque nœud réalisé en plusieurs étapes, et l'exécution de ces scripts sur les différents nœuds.

a. Affectation des machines

La première étape consiste à affecter à chaque fonctionnalité à émuler un nœud de la grappe.

Ces nœuds sont classés en deux catégories : Client et Réseau.

Les machines se trouvant dans la première ne peuvent émuler que des clients et des générateurs de trafic. Les autres n'ont pas ces restrictions.

(Dans le futur, il sera envisageable d'ajouter d'autres catégories si d'autres infrastructures (comme des nœuds aux applications très orientées ou des appareils spécifiques) sont incorporées à EWAN)

Par ailleurs, chaque fonctionnalité impose des contraintes physiques au nœud candidat :

Par exemple, un routeur de cœur doit avoir au moins 3 interfaces, un émulateur de lien doit posséder suffisamment de capacité CPU pour assurer la latence demandée (plus la latence est élevée, plus les files d'attente des paquets sont longues, et plus les opérations sur ces files prennent du temps).

L'algorithme du premier script cherche donc à assigner des nœuds aux fonctionnalités en essayant de trouver une combinaison satisfaisante.

b. Détermination des sous-réseaux et IP

Tous les nœuds sont reliés entre eux par un ou plusieurs commutateurs, formant ainsi un sous-réseau leur permettant de communiquer de l'un vers l'autre directement.

Or pour notre simulation, nous devons éviter cela : Si un client s'adresse à un autre client, ses paquets devront normalement passer par un point d'accès, un émulateur de lien, plusieurs routeurs de cœur, un autre émulateur de lien et enfin un autre point d'accès.

Pour cela, nous regroupons ces nœuds en sous-réseau, comme cela est déjà le cas dans une topologie réelle, en créant un sous-réseau pour chaque paire de routeurs possédant un lien en commun. La seule différence avec la réalité vient du fait que les émulateurs de liens sont eux même techniquement des routeurs. Si un point d'accès et son routeur de cœur étaient dans le même sous-réseau, les paquets ne traverseraient jamais l'émulateur de lien, et ne seraient pas retardés.

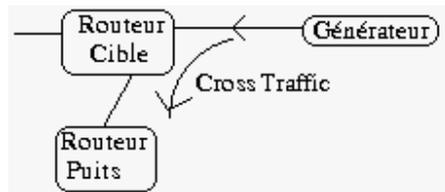
La détermination de ces sous-réseaux permet ensuite d'affecter des IP privées aux interfaces des nœuds.

Toutefois, une IP du sous-réseau initial est conservée, afin de former un réseau de contrôle, offrant ainsi un accès direct aux nœuds sans avoir à passer à travers les différents sous-réseaux. Cette sécurité permet d'intervenir sur des machines même si le routage ne fonctionne plus.

Afin de ne pas réduire le nombre d'interfaces disponibles par nœud (qui est un facteur très limitant), ces interfaces de contrôle sont en réalité des interfaces virtuelles : à une interface physique on associe deux interfaces virtuelles, aux IP distinctes qui se partageront le débit.

L'inconvénient des interfaces virtuelles est donc la baisse de performance qu'engendre ce partage. Mais comme l'interface de contrôle n'est utilisée qu'avant l'expérience, à sa fin, ou en cas d'erreur, elle ne perturbera pas l'autre interface durant les mesures.

C'est également dans cette phase que sont calculées les adresses nécessaires aux générateurs cross traffic : l'IP de l'interface à congestionner en sortie, l'IP de l'interface de ce même routeur, mais à utiliser en entrée, l'IP d'une interface du puits qui ne soit pas le même sous-réseau que le générateur ou l'interface d'entrée. Si aucun puits n'a été défini, l'algorithme en assigne un parmi les nœuds du sous-réseau de l'interface de sortie.



c. Calcul du routage.

Une fois ces sous-réseaux formés, nous pouvons calculer le routage qui sera réalisé par les routeurs de la topologie.

Celui-ci est statique, c'est à dire qu'il ne va pas varier durant l'expérience (par exemple aucun nœud ne va disparaître, aucun lien ne va se couper...). Ainsi, si un routage optimal est déterminé au début, il n'a aucune raison de changer.

C'est pourquoi le routage sera réalisé statiquement avant le lancement de l'expérience, routage dynamique étant ici inutile.

L'algorithme de calcul de routage retenu est celui d'OSPF ^[15], basé sur le calcul du plus court chemin (algorithme de Dijkstra).

EWAN intègre donc cet algorithme implémenté en PHP (et donné en Annexe [4]).

Préparation au Déploiement

Génération des tables MySQL : **Ok**

Affectation des machines : **Ok**

Affectation des IP et création des sous-réseaux : **Ok**

Calcul des tables de routage : **Ok**

Débuter le déploiement

d. Tables MySQL

Diverses informations aussi bien sur les nœuds réels que sur fonctionnalités émulées sont stockées dans une base MySQL à la fin du processus.

Cela permet à une personne ou une application extérieure d'avoir des renseignements sur la topologie émulée ou sur la grappe réelle (par exemple les tables de routage des nœuds).

Pour illustrer cette fonctionnalité, un petit script écrit à l'occasion d'une démonstration est fourni en Annexe [5]

e. Déploiement

Le déploiement se fait en 3 mouvements :

Pour chaque nœud à utiliser, un script de configuration est généré localement sur le serveur (et affiché sur la page web).

Chaque script est copié par SSH sur le nœud correspondant

Le script est exécuté (toujours par SSH) sur le nœud (Les erreurs et les messages d'alertes sont transmis à l'utilisateur).

Machine 12 IP : 140.77.255.13 émule : r3

```
ip address flush label eth+;
if [ \cat /proc/sys/net/ipv4/ip_forward\` -lt 1 ];
then echo Attention, activation de IP forward;
echo 1 > /proc/sys/net/ipv4/ip_forward;
fi;
if [ \cat /proc/sys/net/ipv4/conf/all/rp_filter\` -gt 0 ];
then echo Attention, desactivation de la protection anti-spoofing;
echo 0 > /proc/sys/net/ipv4/conf/all/rp_filter;
fi;
ifconfig eth0:0 mtu 9000 192.168.2.2;
ifconfig eth1 mtu 9000 192.168.6.2;
ifconfig eth2 mtu 9000 192.168.8.1;
ifconfig eth0:1 140.77.255.13 netmask 255.255.255.0;
route add -net 192.168.1.0 gw 192.168.2.1 netmask 255.255.255.0 dev eth0;
route add -net 192.168.3.0 gw 192.168.2.1 netmask 255.255.255.0 dev eth0;
route add -net 192.168.4.0 gw 192.168.2.1 netmask 255.255.255.0 dev eth0;
route add -net 192.168.5.0 gw 192.168.2.1 netmask 255.255.255.0 dev eth0;
route add -net 192.168.7.0 gw 192.168.2.1 netmask 255.255.255.0 dev eth0;
route add -net 192.168.9.0 gw 192.168.2.1 netmask 255.255.255.0 dev eth0;
route add -net 192.168.10.0 gw 192.168.2.1 netmask 255.255.255.0 dev eth0;
route add -net 192.168.11.0 gw 192.168.2.1 netmask 255.255.255.0 dev eth0;
route add -net 192.168.12.0 gw 192.168.2.1 netmask 255.255.255.0 dev eth0;
route add -net 192.168.13.0 gw 192.168.2.1 netmask 255.255.255.0 dev eth0;
route add -net 192.168.14.0 gw 192.168.8.2 netmask 255.255.255.0 dev eth2;
route add -net 192.168.15.0 gw 192.168.8.2 netmask 255.255.255.0 dev eth2;
route add default gw 140.77.255.1 netmask 0.0.0.0 dev eth0;
if [ "`cni2net -d 2>/dev/stdout| grep command`" ];
then echo NIST Net non dispo;
fi;
echo OK;
```

F. Résultats

Execution des scripts

p0 sur 140.77.15.128

NIST Net non dispo
OK

r0 sur 140.77.15.121

Attention, activation de IP forward
Attention, desactivation de la protection anti-spoofing
NIST Net non dispo
OK

c0 sur 140.77.15.122

Attention, activation de IP forward
Attention, desactivation de la protection anti-spoofing
NIST Net non dispo
OK

1. Validation

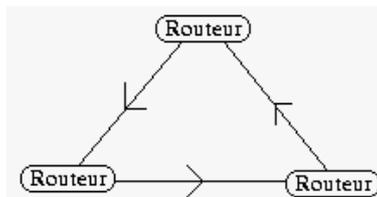
Les premiers tests réels ont été réalisés sur quatre machines mises à disposition pour EWAN, sur un réseau de 100Mb/s. Nous avons défini des tests unitaires que nous avons pu vérifier grâce aux outils : ping (pour vérifier l'accessibilité à un nœud et ses conditions, comme la latence...), traceroute (pour en plus connaître le chemin suivi pour atteindre un nœud), et iperf (pour notamment effectuer des mesures de débit et de pertes).

a. Émulation de lien

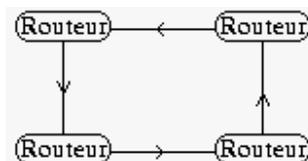


Testé avec ping et traceroute : **OK**

b. Routage

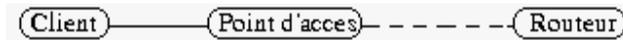


Puis



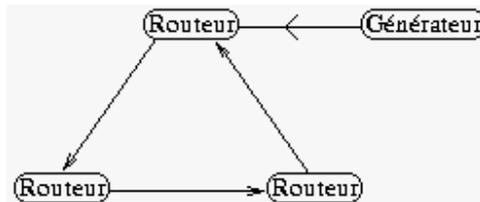
Testé avec ping et traceroute : **OK**

c. Limitation de débit



Testé avec ping, traceroute et iperf: **À calibrer** : Les résultats sont approximatifs pendant les premières secondes, mais cela peut se régler en choisissant de meilleurs paramètres pour tc. (Voir 2. Performances)

d. Génération de Cross Traffic



Testé avec ping et traceroute et iperf: **OK**

Il n'y a eu aucun obstacle majeur rencontré lors de ces tests, si ce n'est celui de la protection *rp_filter* :

Dans le cas des topologies à plus de quatre points d'accès, et avec un cœur de distance nulle, les routeurs de cœur sont organisés en anneau unidirectionnel, sans émulateur de liens entre eux. Cette configuration permet de réaliser un cœur surdimensionné par rapport au reste de la topologie. L'inconvénient de cette organisation est que l'on retrouve dans les tables ARP deux lignes pour une même adresse MAC : une associée à l'interface d'arrivée, et l'autre à celle de départ, différente de la première à cause de l'aspect unidirectionnel de l'anneau. Certains mécanismes de protection contre le *spoofing* (comme le *rp_filter* sous GNU/Linux) jettent alors les paquets reçus. C'est pourquoi EWAN les désactive (en le signalant) au moment de la configuration des nœuds.

Puis des tests sur une grappe de douze nœuds à 1Gb/s ont été effectués et passés avec succès :

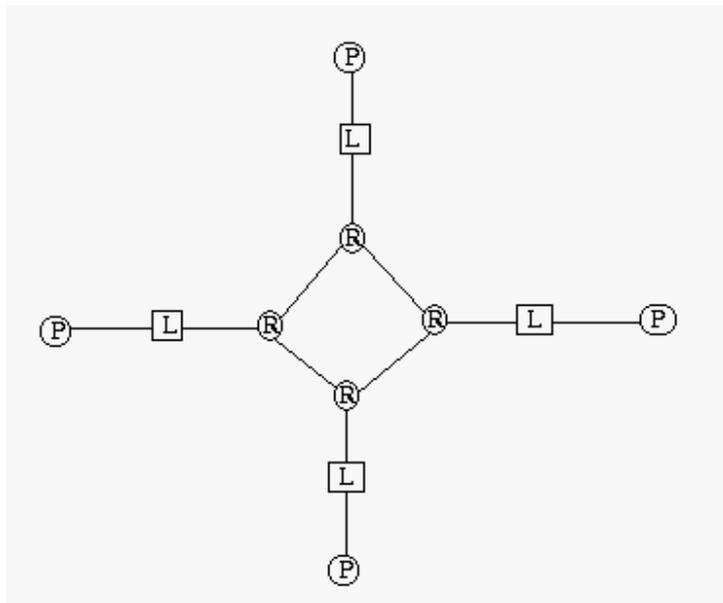
Les topologies essayées sont constituées d'un cœur en anneau et de quatre point d'accès, à distances égales dans le cas équilibré, ou avec l'un ou plusieurs

d'entre eux plus éloignés dans les cas déséquilibrés. Elles présentent également diverses limitations de débit.

L'architecture de la topologie déployée et le routage ont été vérifiés avec `ping` et `tracert`.

La latence émulée a été mesurée avec `ping`. Les valeurs, de 20ms à 200ms correspondaient bien aux prévisions.

Les débits, allant de 1Gb/s (donc sans limitation) à 1Mb/s ont été mesurés avec `iperf`.



2. Performances

L'étude des performances des topologies déployées par EWAN est l'une des tâches d'un ingénieur du laboratoire pour les semaines à venir.

Néanmoins, les premiers résultats sont encourageants :

Par exemple la limitation de débit effectuée par t_c (comme lors du troisième test unitaire) n'est pas effective instantanément, mais après un temps proportionnel au débit voulu (le temps que les tampons de t_c se remplissent).

La précision avec laquelle NIST Net émule la latence est au moins supérieure à celle de l'outil de mesure (`ping`).

Perspectives

Cette première version d'EWAN remplit les objectifs qui lui étaient fixés :

Il dispose d'une interface simple facilitant la définition de topologies élémentaires, puis d'un moteur permettant leur déploiement et celui de topologies quelconques.

Une publication à son sujet est en cours de rédaction dans l'équipe RESO.

Cependant, cet outil pourrait être amélioré en lui rajoutant des fonctions de visualisation, permettant par exemple d'observer en temps réel le trafic, les congestions...

MapCenter ^[16] est un outil implémentant ces fonctionnalités. Son ajout dans EWAN est étudié par des membres du laboratoire.

Durant ces cinq mois d'étude, d'autres solutions d'émulation de latence sont apparues. Des comparatifs entre eux et NIST-Net (utilisé par EWAN) sont en cours :

`sch_netem` est un module du noyau Linux qui reprend les mêmes fonctionnalités que NIST-Net.

Des cartes réseaux contenant des microprocesseurs programmables (*Network Processor*) commencent à sortir. Elles permettent de réaliser l'émulation au niveau de la carte, sans faire intervenir le processeur principal du nœud.

La dernière solution étudiée est GNET-1 ^[17], un appareil spécifiquement dédié à l'émulation de latence. Présenté sous la forme d'une boîte sur laquelle il suffit de brancher des câbles en entrée et en sortie, il émule un lien entre ces points. Pour l'instant le prototype atteint le débit de 1Gb/s, et il est prévu d'arriver à 10Gb/s.

Ces cinq mois de stage passés à l'ENS ont été très intéressants et enrichissants. J'ai énormément apprécié le travail d'équipe. Au vu des résultats positifs du projet, j'ai fortement envie de continuer à contribuer à la suite des travaux. L'année prochaine, ayant été accepté en deuxième année de Master à l'ENS, il me sera donc possible de participer à la suite des recherches dans le laboratoire.

Bibliographie

- [1] SETI@Home : <http://setiathome.ssl.berkeley.edu/>
- [2] Genome@Home : <http://www.stanford.edu/group/pandegroup/genome/>
- [3] Grid'5000 : <http://www.grid5000.org/>
- [4] GGF : Global Grid Forum <http://ggf.org/>
- [5] *A Hierarchy of Network Performance Characteristics for Grid Applications and Services*, Mémo du GGF (<http://www.gridforum.org>), disponible par exemple sur <http://www-didc.lbl.gov/NMWG/docs/measurements.pdf>
- [6] RFC 2330 - *Framework for IP Performance Metrics*
RFC 2679 - *A One-way Delay Metric for IPPM*
RFC 2681 - *A Round-trip Delay Metric for IPPM*
- [7] RCF 2680 - *A One-way Packet Loss Metric for IPPM*
- [8] DRD : *Proactive Packet Dropping Methods for IP Gateways* , Mark **Gaynor** <http://www.eecs.harvard.edu/~gaynor/final.ps>
- [9] Emulab : <http://www.emulab.net>
- [10] X-Bone : <http://www.isi.edu/xbone/>
- [11] ModelNet : <http://issg.cs.duke.edu/modelnet.html>
- [12] NIST Net : <http://snad.ncsl.nist.gov/itg/nistnet/>
- [13] **Dummynet** : http://info.iet.unipi.it/~luigi/ip_dummynet/
- [14] RDF : <http://w3.org/RDF/>
- [15] OSPF : <http://www.ietf.org/html.charters/ospf-charter.html>
- [16] MapCenter : <http://mapcenter.in2p3.fr/>
- [17] GNET-1 : <http://www.gtrc.aist.go.jp/gnet/gnet1e.html>

Remerciements

À l'équipe RESO, notamment :

Pascale Vicat-Blanc Primet

Olivier Glück

Francois Echantillac

Laurent Lefèvre

CongDuc Pham

Marc Herbert

Eric Lemoine

Pablo Neira Ayuso

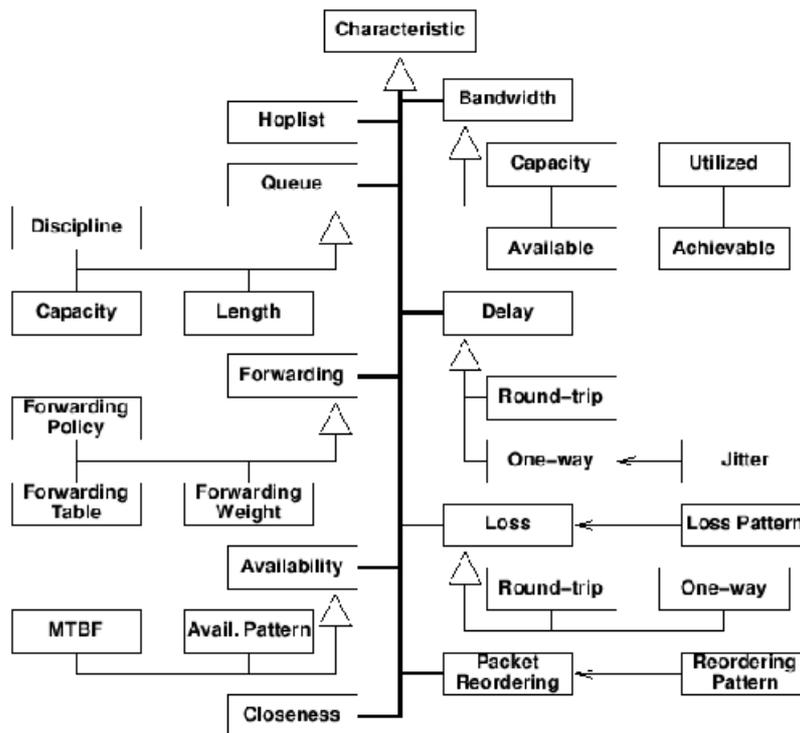
Dino Martin Lopez Pacheco

Au LIP et à son directeur Jean-Michel Muller

Et à Alexandre Saidi.

Annexes

Annexe 1 : Diagramme des propriétés des liens.



Annexe 2 : Exemple de fichier de configuration matérielle

```
config.php :  
<?php  
//Machines disponibles  
$machine1=array();  
$machine1['id']='140.77.15.128';
```

```

$machine1['cpu']=1000;
$machine1['interfaces']=2;
$machine1['Frames']=1500;
$machine1['JumboFrames']=9000;
$machine1['type']='reseau';

$machine2=array();
$machine2['id']='140.77.15.127';
$machine2['cpu']=1000;
$machine2['interfaces']=2;
$machine2['Frames']=1500;
$machine2['JumboFrames']=9000;
$machine2['type']='reseau';

$machine3=array();
$machine3['id']='140.77.15.121';
$machine3['cpu']=800;
$machine3['interfaces']=2;
$machine3['Frames']=1500;
$machine3['JumboFrames']=9000;
$machine3['type']='reseau';

$machine4=array();
$machine4['id']='140.77.15.122';
$machine4['cpu']=800;
$machine4['interfaces']=2;
$machine4['Frames']=1500;
$machine4['JumboFrames']=9000;
$machine4['type']='reseau';

$machine5=array();
$machine5['id']='140.77.15.126';
$machine5['cpu']=500;
$machine5['interfaces']=1;
$machine5['Frames']=1500;
$machine5['JumboFrames']=9000;
$machine5['type']='client';

$machines=array(
    $machine1,
    $machine2,
    $machine3,
    $machine4,
    //$machine5,
);

//Utilisation CPU :
$cpurout = 300;
$cpupt_acc = 200;
$cpucli = 100;
$stocken_buffer=15400000;

$switch_routeur=0;

//timeout SSH
$sshtimeout=3;

```

```

/* Passerelle par défaut pour chaque réseau de controle
potentiel */
$ip_par_defaut=array(
    '140.77.15.0' => '140.77.15.1'
);

$netmask_control='255.255.255.0';
$ip_privee='192.168.';

?>

```

Annexe 3 : Schéma RDFS

Ce fichier décrit le format utilisé pour sauvegarder et charger des topologies.

La dernière version se trouve sur purl.org/net/topo/ns/

```

<?xml version="1.0" encoding="iso-8859-15" ?>
<?xml-stylesheet type="text/css" href="style.css" ?>
<?xml-stylesheet type="text/css" href="fr.css" title="fr"?>
<?xml-stylesheet type="text/css" href="en.css" title="en" alternate="true"?>
<rdf:RDF xmlns:topo="http://purl.org/net/topo/ns/"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:owl="http://www.w3.org/2002/07/owl#"
xmlns:dc="http://purl.org/dc/elements/1.1/"
xmlns:foaf="http://xmlns.com/foaf/0.1/"
xmlns:doap="http://usefulinc.com/ns/doap#"
xml:base="http://purl.org/net/topo/ns" xml:lang="fr">
  <owl:Ontology rdf:about="http://purl.org/net/topo/ns/">
    <owl:versionInfo>0.1.3</owl:versionInfo>
    <dc:title xml:lang="fr">Ontologie de description des topologies de
réseaux informatiques</dc:title>
    <dc:title xml:lang="en">Ontology describing computers networks
topologies</dc:title>
    <foaf:maker>
      <foaf:Person rdf:about="http://perso.ens-
lyon.fr/cyril.otal/cotal.php#cotal">
        <foaf:name>Cyril Ota</foaf:name>
        <foaf:mbox rdf:resource="mailto:cyril.otal@ens-
lyon.fr" />
        <rdfs:seeAlso rdf:resource="http://perso.ens-
lyon.fr/cyril.otal/cotal.php#cotal" />
      </foaf:Person>
    </foaf:maker>
    <rdf:type rdf:resource="http://usefulinc.com/ns/doap#Project" />
    <doap:homepage rdf:resource="http://purl.org/net/topo/ns/" />
    <doap:created>2004-04-30</doap:created>
    <doap:description xml:lang="fr">
      Cette ontologie permet de decrire des topologies réseau. Elle
définit entre autres le format de fichier utilisé par le logiciel EWAN.
    </doap:description>
    <doap:description xml:lang="en">
      the goal of this Ontology is the descriptions of network
topologies. It is for exemple used by the software named EWAN.
    </doap:description>
    <doap:programming-language>RDF,RDFS,OWL</doap:programming-
language>
    <doap:licence
rdf:resource="http://usefulinc.com/doap/licences/GPL" />
    <doap:maintainer>

```

```

        <foaf:Person                                rdf:about="http://perso.ens-
lyon.fr/cyril.otal/cotal.php#cotal" />
        </doap:maintainer>
        <dc:rights> © 2004 ENS Lyon</dc:rights>

        <rdfs:comment></rdfs:comment>
</owl:Ontology>

<rdfs:Class rdf:about="http://purl.org/net/topo/ns/Topologie">
    <rdfs:isDefinedBy rdf:resource="http://purl.org/net/topo/ns/" />
    <rdfs:label>Topologie</rdfs:label>
    <rdfs:comment xml:lang="fr">Une topologie d'un
réseau</rdfs:comment>
    <rdfs:comment xml:lang="en">A network topology</rdfs:comment>
    <rdfs:subClassOf
rdf:resource="http://xmlns.com/wordnet/1.6/topology-4" />
</rdfs:Class>
<rdfs:Class rdf:about="http://purl.org/net/topo/ns/Machine">
    <rdfs:isDefinedBy rdf:resource="http://purl.org/net/topo/ns/" />
    <rdfs:label>Machine</rdfs:label>
    <rdfs:comment xml:lang="fr">Une machine du réseau</rdfs:comment>
    <rdfs:comment xml:lang="en">A computer of the
network</rdfs:comment>
    <rdfs:subClassOf
rdf:resource="http://xmlns.com/wordnet/1.6/computer" />
</rdfs:Class>
<rdfs:Class rdf:about="http://purl.org/net/topo/ns/Client">
    <rdfs:isDefinedBy rdf:resource="http://purl.org/net/topo/ns/" />
    <rdfs:label>Client</rdfs:label>
    <rdfs:comment xml:lang="fr">Une machine client</rdfs:comment>
    <rdfs:comment xml:lang="en">A computer acting as a
client</rdfs:comment>
    <rdfs:subClassOf
rdf:resource="http://purl.org/net/topo/ns/Machine" />
</rdfs:Class>
<rdfs:Class rdf:about="http://xmlns.com/wordnet/1.6/client-3" />
</rdfs:Class>
<rdfs:Class rdf:about="http://purl.org/net/topo/ns/Routeur">
    <rdfs:isDefinedBy rdf:resource="http://purl.org/net/topo/ns/" />
    <rdfs:label>Routeur</rdfs:label>
    <rdfs:comment xml:lang="fr">Un routeur de coeur</rdfs:comment>
    <rdfs:comment xml:lang="en">A core-router</rdfs:comment>
    <rdfs:subClassOf
rdf:resource="http://purl.org/net/topo/ns/Machine" />
</rdfs:Class>
<rdfs:Class rdf:about="http://xmlns.com/wordnet/1.6/router-2" />
</rdfs:Class>
<rdfs:Class rdf:about="http://purl.org/net/topo/ns/Pt-acces">
    <rdfs:isDefinedBy rdf:resource="http://purl.org/net/topo/ns/" />
    <rdfs:label>Pt-acces</rdfs:label>
    <rdfs:comment xml:lang="fr">Un routeur d'accès</rdfs:comment>
    <rdfs:comment xml:lang="en">An ingres-router</rdfs:comment>
    <rdfs:subClassOf
rdf:resource="http://purl.org/net/topo/ns/Machine" />
</rdfs:Class>
<rdfs:Class rdf:about="http://xmlns.com/wordnet/1.6/router-2" />
</rdfs:Class>
<rdfs:Class rdf:about="http://purl.org/net/topo/ns/Generateur">
    <rdfs:isDefinedBy rdf:resource="http://purl.org/net/topo/ns/" />
    <rdfs:label>Generateur</rdfs:label>
    <rdfs:comment xml:lang="fr">Un générateur de cross-
traffic</rdfs:comment>
    <rdfs:comment xml:lang="en">A cross-traffic
generator</rdfs:comment>
    <rdfs:subClassOf
rdf:resource="http://purl.org/net/topo/ns/Machine" />
</rdfs:Class>

```

```

    <rdfs:Class rdf:about="http://purl.org/net/topo/ns/Lien">
      <rdfs:isDefinedBy rdf:resource="http://purl.org/net/topo/ns/" />
      <rdfs:label>Lien</rdfs:label>
      <rdfs:comment xml:lang="fr">Un lien unidirectionnel entre deux
machines</rdfs:comment>
      <rdfs:comment xml:lang="en">An unidirectional link between two
computers</rdfs:comment>
      <rdfs:subClassOf rdf:resource="http://xmlns.com/wordnet/1.6/link-
9" />
    </rdfs:Class>
    <rdfs:Class rdf:about="http://purl.org/net/topo/ns/Def-pertes">
      <rdfs:isDefinedBy rdf:resource="http://purl.org/net/topo/ns/" />
      <rdfs:label>Def-pertes</rdfs:label>
      <rdfs:comment xml:lang="fr">La définition des pertes d'un
lien</rdfs:comment>
      <rdfs:comment xml:lang="en">Definition of the data loss on one
link</rdfs:comment>
      <rdfs:subClassOf rdf:resource="http://xmlns.com/wordnet/1.6/Loss-
2" />
    </rdfs:Class>
    <rdfs:Class rdf:about="http://purl.org/net/topo/ns/Moyenne-pertes">
      <rdfs:isDefinedBy rdf:resource="http://purl.org/net/topo/ns/" />
      <rdfs:label>Moyenne-pertes</rdfs:label>
      <rdfs:comment xml:lang="fr">Un taux moyen de pertes</rdfs:comment>
      <rdfs:comment xml:lang="en">The loss average</rdfs:comment>
      <rdfs:subClassOf rdf:resource="http://purl.org/net/topo/ns/Def-
pertes" />
    </rdfs:Class>
    <rdfs:Class rdf:about="http://purl.org/net/topo/ns/Profil-pertes">
      <rdfs:isDefinedBy rdf:resource="http://purl.org/net/topo/ns/" />
      <rdfs:label>Profil-pertes</rdfs:label>
      <rdfs:comment xml:lang="fr">Un profil de pertes. (Cette ressource
peut avoir une propriété rdfs:seeAlso pointant vers plus d'informations
caractérisant le profil)</rdfs:comment>
      <rdfs:comment xml:lang="en">A loss pattern. (This resource can
have a rdfs:seeAlso property linking to more information about this
pattern)</rdfs:comment>
      <rdfs:subClassOf rdf:resource="http://purl.org/net/topo/ns/Def-
pertes" />
    </rdfs:Class>
    <rdfs:Class rdf:ID="List-noeuds">
      <rdfs:isDefinedBy rdf:resource="http://purl.org/net/topo/ns/" />

      <rdfs:subClassOf rdf:resource="http://www.w3.org/1999/02/22-rdf-
syntax-ns#List" />

      <rdfs:subClassOf>
        <owl:Restriction>
          <owl:onProperty
rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#first" />
          <owl:allValuesFrom
rdf:resource="http://purl.org/net/topo/ns/Machine" />
        </owl:Restriction>
      </rdfs:subClassOf>
      <rdfs:subClassOf>
        <owl:Restriction>
          <owl:onProperty
rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#rest" />
          <owl:allValuesFrom rdf:resource="#List-noeuds" />
        </owl:Restriction>
      </rdfs:subClassOf>

    </rdfs:Class>
    <rdfs:Class rdf:ID="List-liens">
      <rdfs:isDefinedBy rdf:resource="http://purl.org/net/topo/ns/" />

      <rdfs:subClassOf rdf:resource="http://www.w3.org/1999/02/22-rdf-

```

```

syntax-ns#List" />
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty
rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#first" />
            <owl:allValuesFrom
rdf:resource="http://purl.org/net/topo/ns/Lien" />
            </owl:Restriction>
        </rdfs:subClassOf>
    </rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty
rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#rest" />
            <owl:allValuesFrom rdf:resource="#List-liens" />
            </owl:Restriction>
        </rdfs:subClassOf>
    </rdfs:Class>
<rdfs:Class rdf:ID="List-params">
    <rdfs:isDefinedBy rdf:resource="http://purl.org/net/topo/ns/" />
    <rdfs:subClassOf rdf:resource="http://www.w3.org/1999/02/22-rdf-
syntax-ns#Seq" />

    <rdfs:ContainerMembershipProperty
rdf:resource="http://www.w3.org/2000/10/XMLSchema#float"/>
</rdfs:Class>

    <rdf:Property rdf:about="http://purl.org/net/topo/ns/IPv">
        <rdfs:isDefinedBy rdf:resource="http://purl.org/net/topo/ns/" />
        <rdfs:label>IPv</rdfs:label>
        <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Literal"/>
        <rdfs:domain rdf:resource="http://purl.org/net/topo/ns/Topologie"
/>
        <rdfs:comment xml:lang="fr">La version du protocole IP
utilisé</rdfs:comment>
        <rdfs:comment xml:lang="en">The IP version used</rdfs:comment>
    </rdf:Property>
    <rdf:Property rdf:about="http://purl.org/net/topo/ns/QoS">
        <rdfs:isDefinedBy rdf:resource="http://purl.org/net/topo/ns/" />
        <rdfs:label>QoS</rdfs:label>
        <rdfs:range
rdf:resource="http://www.w3.org/2000/10/XMLSchema#boolean"/>
        <rdfs:domain rdf:resource="http://purl.org/net/topo/ns/Topologie"
/>
        <rdfs:comment xml:lang="fr">Utilisation ou non de Qualité de
Service</rdfs:comment>
        <rdfs:comment xml:lang="en">If Quality of Service is
used</rdfs:comment>
    </rdf:Property>
    <rdf:Property rdf:about="http://purl.org/net/topo/ns/MTU">
        <rdfs:isDefinedBy rdf:resource="http://purl.org/net/topo/ns/" />
        <rdfs:label>MTU</rdfs:label>
        <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Literal"/>
        <rdfs:domain rdf:resource="http://purl.org/net/topo/ns/Topologie"
/>
        <rdfs:comment xml:lang="fr">Le type de MTU utilisé</rdfs:comment>
        <rdfs:comment xml:lang="en">The MTU used</rdfs:comment>
    </rdf:Property>
    <rdf:Property rdf:about="http://purl.org/net/topo/ns/ptaccés">
        <rdfs:isDefinedBy rdf:resource="http://purl.org/net/topo/ns/" />
        <rdfs:label>ptaccés</rdfs:label>
        <rdfs:range rdf:resource="http://purl.org/net/topo/ns/Pt-acces"/>

```

```

        <rdfs:domain rdf:resource="http://purl.org/net/topo/ns/Client" />
        <rdfs:comment xml:lang="fr">Le point d'accès auquel le client est
relié</rdfs:comment>
        <rdfs:comment xml:lang="en">The ingres router linked to this
client</rdfs:comment>
    </rdf:Property>
    <rdf:Property rdf:about="http://purl.org/net/topo/ns/from">
        <rdfs:isDefinedBy rdf:resource="http://purl.org/net/topo/ns/" />
        <rdfs:label>from</rdfs:label>
        <rdfs:range rdf:resource="http://purl.org/net/topo/ns/Machine"/>
        <rdfs:domain rdf:resource="http://purl.org/net/topo/ns/Lien" />
        <rdfs:comment xml:lang="fr">La machine destination du
lien</rdfs:comment>
        <rdfs:comment xml:lang="en">The computer which is the destination
of this link</rdfs:comment>
    </rdf:Property>
    <rdf:Property rdf:about="http://purl.org/net/topo/ns/to">
        <rdfs:isDefinedBy rdf:resource="http://purl.org/net/topo/ns/" />
        <rdfs:label>to</rdfs:label>
        <rdfs:range rdf:resource="http://purl.org/net/topo/ns/Machine"/>
        <rdfs:domain rdf:resource="http://purl.org/net/topo/ns/Lien" />
        <rdfs:comment xml:lang="fr">La machine origine du
lien</rdfs:comment>
        <rdfs:comment xml:lang="en">The computer which is at the beginning
of this link</rdfs:comment>
    </rdf:Property>
    <rdf:Property rdf:about="http://purl.org/net/topo/ns/latence">
        <rdfs:isDefinedBy rdf:resource="http://purl.org/net/topo/ns/" />
        <rdfs:label>latence</rdfs:label>
        <rdfs:range
rdf:resource="http://www.w3.org/2000/10/XMLSchema#float" />
        <rdfs:domain rdf:resource="http://purl.org/net/topo/ns/Lien" />
        <rdfs:comment xml:lang="fr">La latence du lien</rdfs:comment>
        <rdfs:comment xml:lang="en">The link delay</rdfs:comment>
    </rdf:Property>
    <rdf:Property rdf:about="http://purl.org/net/topo/ns/perte">
        <rdfs:isDefinedBy rdf:resource="http://purl.org/net/topo/ns/" />
        <rdfs:label>perte</rdfs:label>
        <rdfs:range
pertes"/>
        <rdfs:domain rdf:resource="http://purl.org/net/topo/ns/Lien" />
        <rdfs:comment xml:lang="fr">Les pertes du lien</rdfs:comment>
        <rdfs:comment xml:lang="en">The data loss on the
link</rdfs:comment>
    </rdf:Property>
    <rdf:Property rdf:about="http://purl.org/net/topo/ns/debit">
        <rdfs:isDefinedBy rdf:resource="http://purl.org/net/topo/ns/" />
        <rdfs:label>debit</rdfs:label>
        <rdfs:range
rdf:resource="http://www.w3.org/2000/10/XMLSchema#float" />
        <rdfs:domain rdf:resource="http://purl.org/net/topo/ns/Lien" />
        <rdfs:comment xml:lang="fr">Le débit du lien</rdfs:comment>
        <rdfs:comment xml:lang="en">The capacity of the
link</rdfs:comment>
    </rdf:Property>
    <rdf:Property rdf:about="http://purl.org/net/topo/ns/duplication">
        <rdfs:isDefinedBy rdf:resource="http://purl.org/net/topo/ns/" />
        <rdfs:label>duplication</rdfs:label>
        <rdfs:range
rdf:resource="http://www.w3.org/2000/10/XMLSchema#float" />
        <rdfs:domain rdf:resource="http://purl.org/net/topo/ns/Lien" />
        <rdfs:comment xml:lang="fr">La duplication sur ce
lien</rdfs:comment>
        <rdfs:comment xml:lang="en">The duplication on the
link</rdfs:comment>
    </rdf:Property>
    <rdf:Property rdf:about="http://purl.org/net/topo/ns/liens">
        <rdfs:isDefinedBy rdf:resource="http://purl.org/net/topo/ns/" />

```

```

        <rdfs:label>liens</rdfs:label>
        <rdfs:range rdf:resource="#List-liens" />
        <rdfs:domain rdf:resource="http://purl.org/net/topo/ns/Topologie"
/>
    <rdfs:comment xml:lang="fr">Les liens existants dans la
topologie</rdfs:comment>
    <rdfs:comment xml:lang="en">The links of the
topology</rdfs:comment>
    </rdf:Property>
    <rdf:Property rdf:about="http://purl.org/net/topo/ns/noeuds">
        <rdfs:isDefinedBy rdf:resource="http://purl.org/net/topo/ns/" />
        <rdfs:label>noeuds</rdfs:label>
        <rdfs:range rdf:resource="#List-noeuds" />
        <rdfs:domain rdf:resource="http://purl.org/net/topo/ns/Topologie"
/>
    <rdfs:comment xml:lang="fr">Les noeuds existants dans la
topologie</rdfs:comment>
    <rdfs:comment xml:lang="en">The nodes of the
topology</rdfs:comment>
    </rdf:Property>
    <rdf:Property rdf:about="http://purl.org/net/topo/ns/debit-gen">
        <rdfs:isDefinedBy rdf:resource="http://purl.org/net/topo/ns/" />
        <rdfs:label>debit-gen</rdfs:label>
        <rdfs:range
rdf:resource="http://www.w3.org/2000/10/XMLSchema#float"/>
        <rdfs:domain rdf:resource="http://purl.org/net/topo/ns/Generateur"
/>
    <rdfs:comment xml:lang="fr">Le débit généré pour le cross
traffic</rdfs:comment>
    <rdfs:comment xml:lang="en">The speed of the cross
traffic</rdfs:comment>
    </rdf:Property>
    <rdf:Property rdf:about="http://purl.org/net/topo/ns/lien-cible">
        <rdfs:isDefinedBy rdf:resource="http://purl.org/net/topo/ns/" />
        <rdfs:label>lien-cible</rdfs:label>
        <rdfs:range rdf:resource="http://purl.org/net/topo/ns/Lien"/>
        <rdfs:domain rdf:resource="http://purl.org/net/topo/ns/Generateur"
/>
    <rdfs:comment xml:lang="fr">Le lien correspondant à l'interface à
congestionner</rdfs:comment>
    <rdfs:comment xml:lang="en">The link of the target
interface</rdfs:comment>
    </rdf:Property>
    <rdf:Property rdf:about="http://purl.org/net/topo/ns/noeud-cible">
        <rdfs:isDefinedBy rdf:resource="http://purl.org/net/topo/ns/" />
        <rdfs:label>noeud-cible</rdfs:label>
        <rdfs:range rdf:resource="http://purl.org/net/topo/ns/Machine"/>
        <rdfs:domain rdf:resource="http://purl.org/net/topo/ns/Generateur"
/>
    <rdfs:comment xml:lang="fr">Le noeud auquel appartient l'interface
à congestionner</rdfs:comment>
    <rdfs:comment xml:lang="en">The node whose interface is
targeted</rdfs:comment>
    </rdf:Property>
    <rdf:Property rdf:about="http://purl.org/net/topo/ns/noeud-puit">
        <rdfs:isDefinedBy rdf:resource="http://purl.org/net/topo/ns/" />
        <rdfs:label>noeud-puit</rdfs:label>
        <rdfs:range rdf:resource="http://purl.org/net/topo/ns/Machine"/>
        <rdfs:domain rdf:resource="http://purl.org/net/topo/ns/Generateur"
/>
    <rdfs:comment xml:lang="fr">Le puit vers lequel est dirigé le
trafic généré</rdfs:comment>
    <rdfs:comment xml:lang="en">The node where the generated traffic
goes</rdfs:comment>
    </rdf:Property>
    <rdf:Property rdf:about="http://purl.org/net/topo/ns/param-profil">
        <rdfs:isDefinedBy rdf:resource="http://purl.org/net/topo/ns/" />
        <rdfs:label>param-profil</rdfs:label>

```

```

        <rdfs:range rdf:resource="#List-params"/>
        <rdfs:domain      rdf:resource="http://purl.org/net/topo/ns/Profil-
pertes" />
        <rdfs:comment    xml:lang="fr">Les paramètres d'un profil de
pertes</rdfs:comment>
        <rdfs:comment    xml:lang="en">Parameteres of a loss
pattern</rdfs:comment>
    </rdf:Property>
    <rdf:Property rdf:about="http://purl.org/net/topo/ns/type-profil">
        <rdfs:isDefinedBy rdf:resource="http://purl.org/net/topo/ns/" />
        <rdfs:label>type-profil</rdfs:label>
        <rdfs:range      rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Literal"/>
        <rdfs:domain      rdf:resource="http://purl.org/net/topo/ns/Profil-
pertes" />
        <rdfs:comment    xml:lang="fr">Le nom d'un profil de
pertes</rdfs:comment>
        <rdfs:comment    xml:lang="en">Name of a loss pattern</rdfs:comment>
    </rdf:Property>

    <!-- Compatibilité partielle OWL DL -->
    <owl:AnnotationProperty rdf:about="http://purl.org/dc/elements/1.1/title"
/>
    <owl:AnnotationProperty
rdf:about="http://purl.org/dc/elements/1.1/rights" />
    <owl:AnnotationProperty rdf:about="http://xmlns.com/foaf/0.1/maker" />
    <owl:AnnotationProperty rdf:about="http://usefulinc.com/ns/doap#homepage"
/>
    <owl:AnnotationProperty rdf:about="http://usefulinc.com/ns/doap#created"
/>
    <owl:AnnotationProperty
rdf:about="http://usefulinc.com/ns/doap#description" />
    <owl:AnnotationProperty
rdf:about="http://usefulinc.com/ns/doap#programming-language" />
    <owl:AnnotationProperty rdf:about="http://usefulinc.com/ns/doap#licence"
/>
    <owl:AnnotationProperty
rdf:about="http://usefulinc.com/ns/doap#maintainer" />
</rdf:RDF>

```

Annexe 4 : Implémentation de l'algorithme de Dijkstra en PHP

```

<?php
/*
$links : tableau des liens
$routeurs : tableaux des routeurs (coeur + pt d'accès + NIST-Net)
$metrique : metrique utilisée pour le routage
$tables : tables de routage
    $tables[$i] : table du routeur $i
    $tables[$i][$j] : routeur inscrit sur la ligne de la table pour le sous réseau $j
$reseaux : tableau des sous reseaux. (créé au moment de l'affectation des IPs)
    $reseaux[$i]['noeuds'] : liste des routeurs appartenant au sous reseau $i
$schemins : tableaux des chemins
    $schemins[$i][$j] : numero du routeur à qui le routeur $i doit envoyer les messages

```


?>

Annexe 5 : Script de génération de nom d'hôtes

Ce script se connecte à la base MySQL tenue à jour par EWAN, récupère les interfaces des routeurs, et affiche en sortie des lignes à rajouter dans le fichier système `/etc/hosts` afin que lors d'opérations sur les nœuds de la grappe, leur nom soit utilisé à la place de leur adresse (qui est normalement inconnue de l'utilisateur)

```
<?php
include('inc/var.php');
if($link=mysql_connect($serveur,$user,$mdp))
{
    if(mysql_select_db($db))
    {
        $r=mysql_query('SELECT ID,Interfaces FROM Routeurs');
        while($n=mysql_fetch_array($r))
        {
            $ip=unserialize($n['Interfaces']);
            for($i=0;$i<count($ip);$i++)
            {
                echo "\n".$ip[$i].' '.$n['ID'];
            }
        }
    }
else
    {
        echo 'Base injoignable';
    }
}
else
    {
        echo 'Connexion impossible';
    }
?>
```

Annexe 6 : Listing des fichiers d'EWAN

```
affect.php
charger.php
config.php
configmat/
css/
    site.css
deploy.php
dijkstra.php
equation.php
exemples/
img/
    nuage.php
    nuage.png
    routeur.png
    stone.ttf
    topo.php
    VeraIt.ttf
inc/
    footer.php
    head.php
    headers.php
    nav.php
    prologues.php
    var.php

index.php
interface.php
ip.php
lat2cpu.php
process.php
routage.php
topologie.php
```

Table des Matières

Résumé	3
Abstract.....	3
Introduction et contexte	4
A. Présentation du Laboratoire de Recherche	5
B. Contexte	6
I. Grilles de calcul et topologies.....	6
1. Entités.....	6
2. Propriétés.....	7
a. Bande passante.....	7
La Capacité	7
L'Utilisation	8
La Bande Passante disponible.....	8
La Bande Passante atteignable.....	8
b. Latence.....	8
La Gigue	9
c. Pertes.....	9
d. Duplications.....	9
II. Outils	9
1. NIST Net	10
2. Dummynet.....	11
3. TC.....	12
Réalisation	13
A. Spécifications	13
B. Architecture logique.....	13
C. Architecture physique	15
D. Interface de création de topologie.....	16
1. Charger une topologie.	16
2. Définir une nouvelle topologie.	17
a. Points d'accès et Clients.....	17
b. Définition des liens.....	18
c. Limitation des débits	19
d. Choix de la topologie	20
e. Cross Traffic	21
f. Paramètres avancés optionnels.....	21
g. Fin de la configuration.	22
3. Gestion des configurations matérielles	23
4. Vérification de la configuration matérielle	23
E. Déploiement des topologies	24
1. Format de définition des topologies.....	24
2. Exécution du déploiement.....	25
a. Affectation des machines.....	25
b. Détermination des sous-réseaux et IP	26

c. Calcul du routage.....	27
d. Tables MySQL	28
e. Déploiement.....	28
F. Résultats.....	30
1. Validation.....	30
a. Émulation de lien.....	30
b. Routage.....	30
c. Limitation de débit.....	31
d. Génération de Cross Traffic	31
2. Performances.....	32
Perspectives	33
Bibliographie	34
Remerciements	35
Annexes	36
Annexe 1 : Diagramme des propriétés des liens	36
Annexe 2 : Exemple de fichier de configuration matérielle	36
Annexe 3 : Schéma RDFS	38
Annexe 4 : Implémentation de l'algorithme de Dijkstra en PHP.....	46
Annexe 5 : Script de génération de nom d'hôtes.....	48
Annexe 6 : Listing des fichiers d'EWAN	50
Table des Matières.....	51

This document was created with Win2PDF available at <http://www.daneprairie.com>.
The unregistered version of Win2PDF is for evaluation or non-commercial use only.