



# Visualisation dynamique du backbone IPv6 RENATER II

Julien Pieraut

► **To cite this version:**

Julien Pieraut. Visualisation dynamique du backbone IPv6 RENATER II. [Stage] A02-R-197 || pieraut02a, 2002, 22 p. inria-00099443

**HAL Id: inria-00099443**

**<https://hal.inria.fr/inria-00099443>**

Submitted on 26 Sep 2006

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Présentation de l'entreprise</b>	<b>3</b>
2.1	Présentation générale . . . . .	3
2.2	Personnel . . . . .	3
2.3	Thématiques de recherche . . . . .	3
<b>3</b>	<b>Contexte de travail</b>	<b>5</b>
3.1	Présentation de l'équipe de recherche . . . . .	5
<b>4</b>	<b>Présentation du problème à résoudre</b>	<b>6</b>
4.1	IPv6 et présentation du G6 . . . . .	6
4.1.1	IPv6 : objectifs et transition . . . . .	6
4.1.2	Une plate-forme de test : le 6bone . . . . .	7
4.1.3	Le pilote IPv6 de Renater et le G6 . . . . .	8
4.2	Les outils de supervision du Loria . . . . .	9
4.3	Problème à résoudre . . . . .	10
<b>5</b>	<b>Analyse du problème</b>	<b>12</b>
5.1	Objectifs . . . . .	12
5.2	Choix des outils de développement . . . . .	12
5.2.1	Environnement de développement . . . . .	12
5.2.2	Scripts CGI . . . . .	13
5.2.3	Perl . . . . .	15
5.2.4	Feuilles de style et Javascript . . . . .	15
5.3	Mise en œuvre . . . . .	15
5.3.1	Apprentissage de Perl . . . . .	15
5.3.2	Simplification de l'arborescence . . . . .	16
5.3.3	Mise aux normes XHTML 1.0 et CSS 2 . . . . .	16
5.3.4	Mise en place d'une charte graphique . . . . .	16
5.3.5	Mise en place de la carte dynamique . . . . .	16
5.3.6	Evaluation . . . . .	16
5.3.7	Améliorations . . . . .	16
<b>6</b>	<b>Résultats des réalisations</b>	<b>17</b>
6.1	Description . . . . .	17
6.2	Problèmes rencontrés et limitations . . . . .	19
<b>7</b>	<b>Conclusion</b>	<b>20</b>
7.1	Améliorations possibles . . . . .	20
7.2	Intérêt du stage . . . . .	20
	<b>Bibliographie</b>	<b>20</b>

# Chapitre 1

## Introduction

Le stage présenté dans ce mémoire a été effectué au Loria<sup>1</sup> du 11 mars au 29 juin 2002, au sein de l'équipe de recherche Resedas, qui s'occupe de développer "concepts et outils logiciels pour les télécommunications et les systèmes distribués".

Il s'est déroulé dans le cadre d'une licence professionnelle effectuée à l'IUT Charlemagne lors de l'année scolaire 2001-2002, option "concepteur-intégrateur de systèmes Internet/Intranet".

Après une présentation du laboratoire et de l'équipe de recherche, ce mémoire présentera rapidement le contexte de travail dans lequel ce stage fut effectué, puis fera un bilan détaillé du travail traité : après une présentation du problème à résoudre, son analyse et le résultat des réalisations permettront d'établir une conclusion sur l'intérêt du stage.

En annexes figurent les documents nécessaires à la bonne compréhension du mémoire.

---

<sup>1</sup>Laboratoire Lorrain de Recherche en Informatique et ses Applications

# Chapitre 2

## Présentation de l'entreprise

### 2.1 Présentation générale

Le Loria est une unité mixte de recherche commune au CNRS<sup>1</sup>, à l'Inria<sup>2</sup>, à l'INPL<sup>3</sup>, et aux universités Henri Poincaré, Nancy 1 et Nancy 2.

Cette unité, dont la création a été officialisée en 1997, succède au Crin<sup>4</sup> et aux équipes communes entre celui-ci et l'Unité de Recherche Inria de Lorraine.

### 2.2 Personnel

Depuis le 1er janvier 2001, c'est Hélène Kirchner qui dirige le laboratoire. Actuellement, plus de 300 personnes y travaillent. Ce personnel est réparti en 21 équipes de recherche et 7 services d'aide à la recherche. Chaque équipe rassemble des chercheurs, des doctorants et des assistants techniques ou administratifs, pour la réalisation d'un projet de recherche.

### 2.3 Thématiques de recherche

Dans le secteur des sciences et technologies de l'information et de la communication, le Loria possède des compétences dans des secteurs en pleine évolution. Les activités de ces équipes sont centrées autour de cinq thématiques principales :

- **Calculs, réseaux et graphismes à hautes performances** : regroupant des chimistes, des mathématiciens, des physiciens et des informaticiens, ce thème fixe deux objectifs principaux : la maîtrise des calculs à distance sur des réseaux hétérogènes de machines hétérogènes (*meta-computing*), et la visualisation et la simulation dans des environnements immersifs et interactifs en trois dimensions.
- **Télé-opérations et assistant intelligents** : l'objectif de ce thème est de concevoir et maîtriser les assistants intelligents et les services actifs à distance, omni-présents dans la vie courante (télé-surveillance et soins à domicile, assistants intelligents pour les handicapés, apprentissage des langues, télé-opérations pour le commerce électronique...)

---

<sup>1</sup>Centre National de la Recherche Scientifique

<sup>2</sup>Institut National de Recherche en Informatique et en Automatique

<sup>3</sup>Institut National Polytechnique de Lorraine

<sup>4</sup>Centre de Recherche en Informatique de Nancy

- **Ingénierie des langues, du document et de l'information scientifique et technique :** comment coder et normaliser les contenus d'Internet et du Web, au niveau mondial ou dans les entreprises, et les rendre accessibles pour les exploiter de façon répartie via les réseaux ? Ce thème a pour objectif de proposer des outils performants pour le codage et la gestion informatique de cette information, pour la structuration des connaissances, et pour l'accès à travers un dialogue entre l'utilisateur et le système.
- **Qualité et sûreté des logiciels et systèmes informatiques :** les travaux dans ce domaine visent à apporter une assistance à la normalisation de la production de logiciels et à l'établissement de procédures de certification associées, afin de produire des logiciels sûrs en concevant et validant des composants logiciels de qualité.
- **Bioinformatique et applications à la génomique :** deux directions de travail sont actuellement proposées sur ce thème ; d'une part, l'utilisation et la confrontation des méthodes combinatoires et connexionnistes pour répondre à des questions de prédiction de structure et de fonction des macromolécules biologiques, et d'autre part l'application des méthodes informatiques en recherche d'information et en extraction de connaissances, aux problèmes de visualisation, de traitement et d'interprétation des données sur les génomes.

# Chapitre 3

## Contexte de travail

### 3.1 Présentation de l'équipe de recherche

Le projet Resedas, commun à l'Inria et au Loria, a pour cadre général *"la conception d'outils logiciels expérimentaux pour faciliter le développement, le déploiement et l'exploitation de services, protocoles et applications distribuées sur des réseaux de télécommunications et des réseaux locaux"*.

Dans ce cadre, le groupe développe des activités sur les trois thèmes suivants :

- **Gestion des réseaux et des services** (application au Réseau de Gestion des Télécommunications).
- **Calcul de répartition sur des réseaux de stations hétérogènes.**
- **Protocole IP nouvelle génération (IPv6)** : expérimentation, validation de plateforme et conception de nouvelles classes d'applications.

C'est sur ce dernier thème que ce mémoire va s'attarder, en présentant tout d'abord IPv6, ses enjeux et sa plate-forme de test française, dont plusieurs outils de supervision sont développés au Loria au sein de Resedas.

Cette présentation nécessaire permettra un exposé précis du problème à résoudre.

# Chapitre 4

## Présentation du problème à résoudre

### 4.1 IPv6 et présentation du G6

#### 4.1.1 IPv6 : objectifs et transition

IPv6 est la nouvelle version d'IP<sup>1</sup>, qui devrait se substituer progressivement au protocole IPv4 actuellement en place. IPv6 a été principalement conçu afin d'étendre l'espace d'adressage actuel devenu trop petit. Les adresses IPv6 définies sur 128 bits multiplient de ce fait la taille d'Internet puisque l'espace d'adressage passe maintenant à  $2^{128}$  adresses utilisables.

Conséquence de cette extension de l'adressage, une nouvelle notation a été définie pour décrire les adresses IPv6 de 16 octets. Elle comprend 8 groupes de 4 chiffres hexadécimaux séparés avec le symbole deux-points. Par exemple :

```
8000:0000:0000:0000:0123:4567:89AB:CDEF
```

Puisque plusieurs adresses ont de nombreux zéros dans leur libellé, 3 optimisations ont été définies. Tout d'abord, les premiers zéro d'un groupe peuvent être omis, comme par exemple 0123 qui peut s'écrire 123. Ensuite, un ou plusieurs groupes de 16 zéros consécutifs peuvent être remplacés par un double deux-points. C'est ainsi que l'adresse ci-dessus devient :

```
8::123:4567:89AB:CDEF
```

Enfin, les adresses IPv4 peuvent être écrites en utilisant la représentation de l'adresse en notation décimale pointée précédée d'un double deux-points, comme par exemple :

```
::192.31.320.46
```

Les caractéristiques d'IPv6 prennent en compte :

- **Une configuration *Plug'n'Play***, grâce aux mécanismes d'auto-configuration des machines : l'adresse d'une machine se décomposant en une partie dépendante du réseau sur laquelle elle se trouve (préfixe) et d'une partie correspondant à l'adresse physique de sa carte réseau (son adresse ethernet par exemple), il devient alors très simple pour un équipement de fabriquer son adresse.

---

<sup>1</sup>Internet Protocol

- **Un routage efficace**, par une réduction de la taille des tables de routage et une simplification du protocole, pour permettre aux routeurs de router les datagrammes plus rapidement.
- **Des mécanismes standard de sécurité**, puisque l'authentification et la confidentialité constituent les fonctions de sécurité majeures d'IPv6.
- **La mobilité**, à savoir la possibilité de rester accessible et connecté tout en changeant de réseau dans Internet grâce à l'obtention d'une adresse universelle.
- **La transition IPv4 vers IPv6**, afin d'assurer la communication entre des réseaux IPv4 et des réseaux IPv6 et la communication entre deux réseaux IPv6 éloignés en passant par le monde IPv4.

Ce dernier point nécessitait logiquement la mise en place d'une plate-forme de test au niveau mondial, étant donné la longue période de cohabitation prévue entre les deux protocoles.

#### 4.1.2 Une plate-forme de test : le 6bone

Le 6bone est le nom qui a été donné à cette plate-forme de test mondiale. Le 6bone rassemble plus de 40 pays participants, et plus de 460 sites concernés. Démarré en 1996 entre les 3 premiers réseaux expérimentaux (Wide au Japon, NRL aux Etats-Unis et G6 en France), il restera en place jusqu'à un très large déploiement d'IPv6.

Le 6bone est en fait un réseau IPv6 expérimental construit au-dessus d'Internet IPv4 par le biais de *tunnels*. Ces tunnels permettent de relier entre eux des îlots IPv6 éparpillés sur la planète, tout en cohabitant avec IPv4.

En effet, comme le montre la figure 4.1, une machine A, avec les deux couches réseau IPv6 et IPv4, peut atteindre une machine B avec une adresse IPv6 en envoyant une trame IPv6. Si A et B sont sur le même réseau, aucun problème si ce n'est que le format des trames obéit au nouveau protocole. Si B n'est pas sur le même réseau que A, alors A envoie la trame IPv6 à une machine qui va remplir une fonction dite de tunnel. Cette dernière va fabriquer une trame IPv4 la contenant (mécanisme d'encapsulation) qui sera acheminée à travers Internet jusqu'au tunnel qui remplira la fonction symétrique pour la machine B (en déballant la trame IPv4 reçue pour en émettre la trame IPv6).

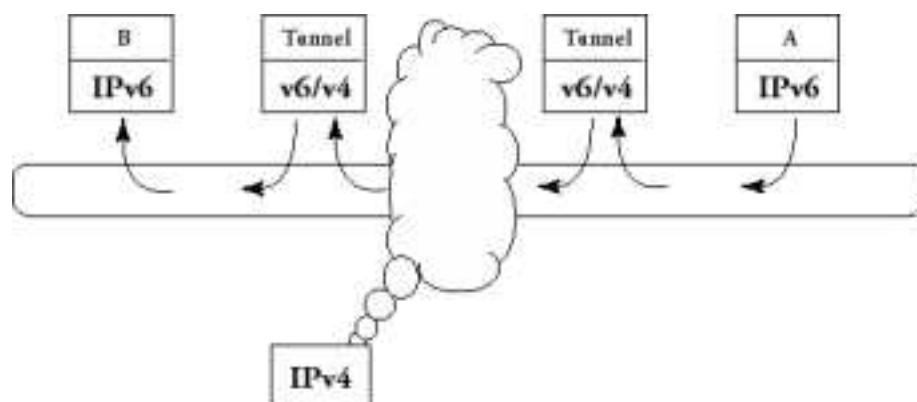


FIG. 4.1 – Tunnel IPv6



Cette technique assure un fonctionnement correct pendant une phase de transition qui peut durer un certain temps avant que toutes les couches réseau de toutes les machines ne soient au même niveau. Elle garantit également une grande simplicité d'implémentation.

### 4.1.3 Le pilote IPv6 de Renater et le G6

Le G6 est une association Loi de 1901 qui s'est fixé pour objectif de "favoriser le développement et le déploiement de la nouvelle version du protocole Internet, IPv6, successeur d'IPv4". Cette association, présidée par Alain Durand et Bernard Tuy, est née en 1995, officialisée en janvier 2001. Ses acteurs se sont donné comme charte :

- L'échange d'expériences avec IPv6.
- L'établissement de plates-formes d'expérimentations.
- La construction et l'administration d'une infrastructure IPv6 en France.
- La diffusion d'information sur IPv6.
- Les relations avec les différents groupes de travail (Ripe, IETF, IOL).
- La conduite d'expérimentations dans le cadre de projets nationaux ou internationaux.
- La mise en œuvre d'un réseau national IPv6 natif.

Le Pilote IPv6 de Renater est un service expérimental bâti sur l'infrastructure ATM de Renater-2. Il constitue un réseau privé virtuel de niveau 2. Ce réseau a succédé au G6-Bone dont il a acquis l'expérience.

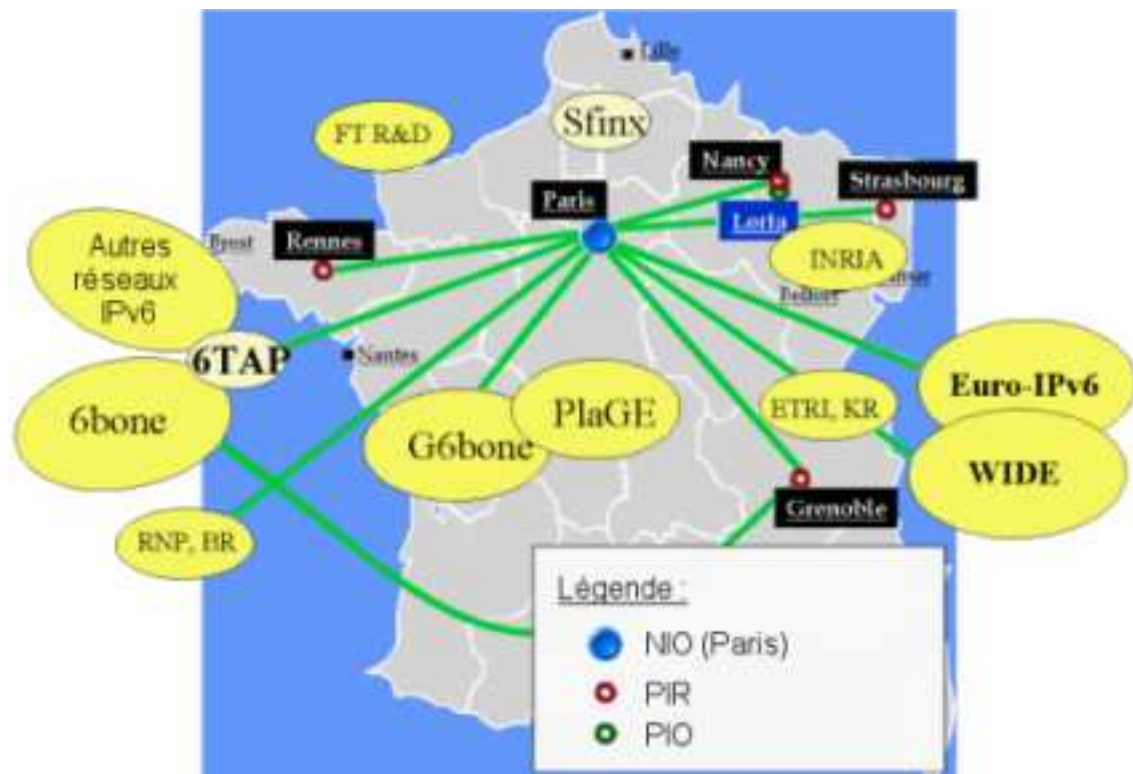


FIG. 4.2 – Pilote IPv6 de Renater

Comme l'indique la figure 4.2, le pilote IPv6 de Renater 2 rassemble plusieurs plate-formes de test, raccordées entre elles via des Points d'Interconnexion Organisationnels (PIO) et des Points

d'Interconnexion Régionaux (PIR). Ces points d'interconnexion (PIx) constituent le cœur du réseau. Ils permettent aux plates-formes de tests qui y sont connectées de pouvoir échanger du trafic IPv6 entre elles. Il en existe actuellement à Nancy, Rennes, Strasbourg, et Grenoble, tous reliés de manière centrale au Noeud d'Interconnexion d'Operateur (NIO) de Paris, qui permet au pilote IPv6 de Renater d'être lui-même relié au 6bone.

En pratique, les sites régionaux s'adressent en priorité au responsable du PIR géographiquement le plus proche pour être raccordés au réseau pilote. Les sites qui relèvent des PIO s'adressent en priorité au responsable du PIO dont ils relèvent.

## 4.2 Les outils de supervision du Loria

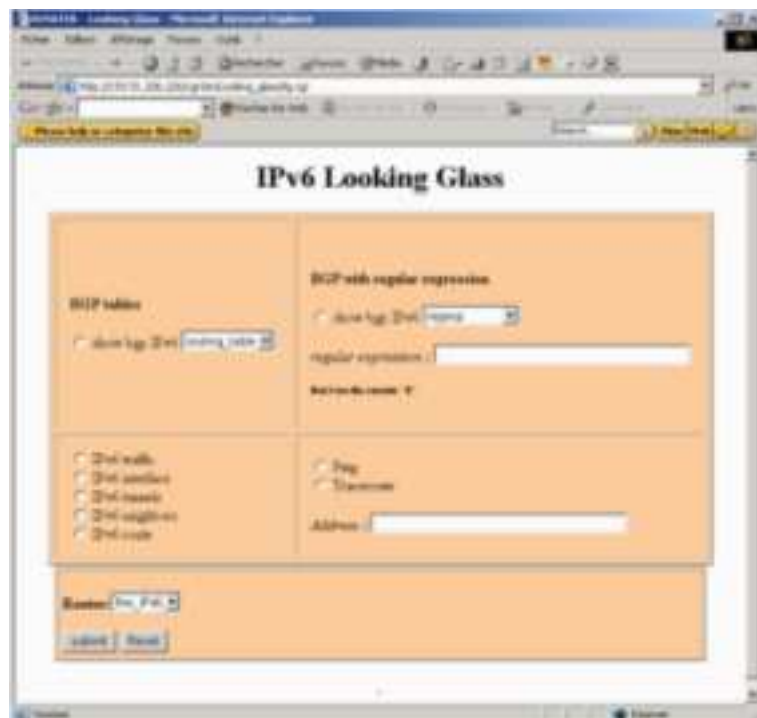


FIG. 4.3 – Interface du *Looking Glass de Paris*

Le réseau pilote IPv6 de Renater est principalement supervisé au moyen d'un outil écrit en *Perl*, le *Looking Glass*. Cet outil est utilisé par le biais d'une interface Web (figure 4.3), qui propose un ensemble de choix de requêtes sur le routeur désiré ; l'utilisateur peut ainsi obtenir, entre autres :

- L'ensemble des tables BGP<sup>2</sup>.
- La table de routage.
- Un résumé des caractéristiques du routeur.
- Les routeurs voisins (*neighbors*).
- Des statistiques sur le trafic IPv6, ICMP et UDP en réception, en envoi et en multicast.
- Des détails sur l'interface.
- Les tunnels établis.

<sup>2</sup>Border Gateway Protocol

Les résultats ainsi obtenus s'affichent de façon détaillée et brute, comme le montre la figure 4.4.

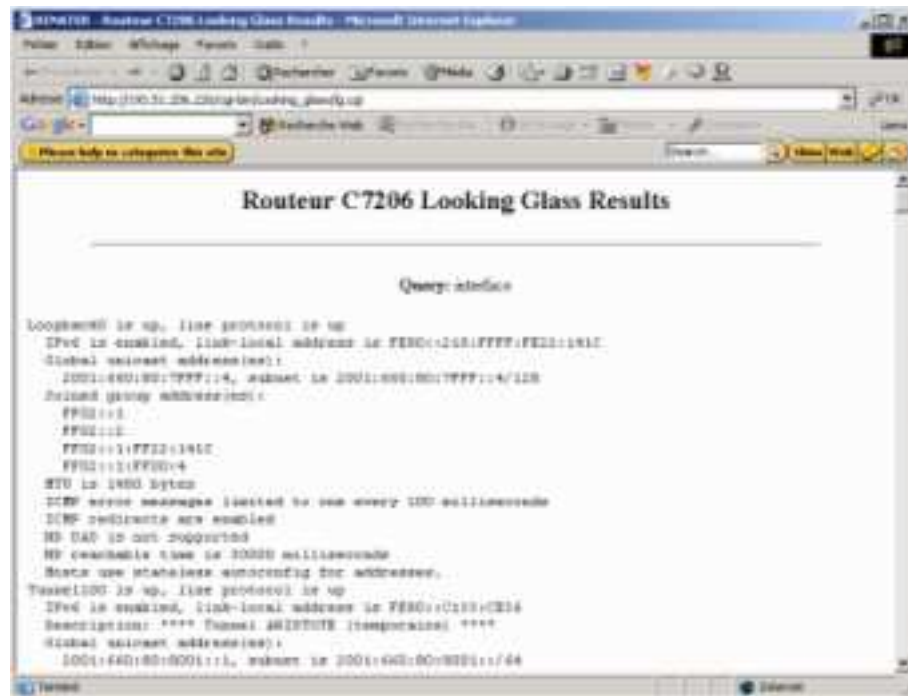


FIG. 4.4 – Résultat d’une requête

Au Loria, le site Web *w6.loria.fr* constitue l’un des sites de supervision du réseau pilote de Renater 2. Ce site, qui met à disposition les outils développés au sein du laboratoire et les différentes publications disponibles, possède également une partie présentant une carte de France des routeurs du backbone. Un lien vers chacun des routeurs permet d’interroger les différents *Looking Glass* installés pour la supervision.

### 4.3 Problème à résoudre

Le problème principal se posant en Mars 2001 est le caractère trop statique de ce site de supervision, et en particulier de cette fameuse carte de France des routeurs. En effet, même si l’utilisateur peut effectuer des requêtes sur chacun des *Looking Glass*, détecter un problème sur l’un des routeurs n’est pas immédiat et nécessite l’utilisation de chacun des *Looking Glass*, et la présentation des résultats est trop peu intuitive pour en détecter immédiatement la cause, voire même pour obtenir des informations rapidement.

Il est donc nécessaire de trouver une façon plus dynamique, rapide et intuitive de présenter certaines statistiques et résultats, tout en gardant la présentation très pratique sous forme de carte de France et les requêtes complètes sur les routeurs grâce aux *Looking Glass* installés.

D’autre part, le site actuel manque de clarté et la carte des routeurs n’est pas assez précise ni mise à jour en ce qui concerne les liens entre routeurs et leur dénombrement, comme le montre la figure 4.5.



FIG. 4.5 – Carte de France du backbone IPv6 au 10 Mars 2001

# Chapitre 5

## Analyse du problème

### 5.1 Objectifs

L'objectif principal est donc la mise en place d'un affichage dynamique de la carte des routeurs du backbone IPv6 constituant le pilote IPv6 de Renater 2, basé sur une carte de France présentant l'ensemble des routeurs et leurs caractéristiques, à savoir une visualisation dynamique et immédiate :

- Des liens entre routeurs voisins.
- Du status de chacun des routeurs.
- De statistiques quotidiennes sur le trafic IPv6 de chaque routeur.
- De diverses caractéristiques telles que les adresses liées au routeur.

En outre, un accès immédiat et rapide doit être disponible à l'utilisateur qui désire une information plus détaillée lui permettant de déterminer la cause d'un problème le cas échéant, ou plus simplement d'effectuer une requête lui mettant à disposition tables de routages, interfaces et autres données qui caractérisent le routeur.

Ces objectifs induisent évidemment des soucis d'ergonomie et de fonctionnalité. Afin d'y parvenir, des objectifs mineurs ont été fixés, tels que la refonte graphique du site privilégiant la sobriété et la fonctionnalité, et respectant les normes de compatibilité fixées par le W3C<sup>1</sup>.

Enfin, concernant l'écriture de scripts, la généricité et la clarté de l'écriture doivent être privilégiées afin de garantir la meilleure réutilisabilité du travail effectué, par le biais de code clair et commenté et une utilisation optimale de fichiers de configurations.

### 5.2 Choix des outils de développement

#### 5.2.1 Environnement de développement

Il est tout d'abord utile de noter que les machines installées au sein de l'équipe de recherche sont en majorité basées sur FreeBSD version 4, et c'est donc sur ce type de plateforme que s'effectueront les manipulations décrites par la suite, le plus souvent à partir de connections Telnet sur un environnement Windows NT.

---

<sup>1</sup>World Wide Web Consortium

### 5.2.2 Scripts CGI

Un CGI<sup>2</sup> est un programme exécuté *côté serveur*, permettant de cette façon l'affichage de données traitées par le serveur (provenant d'une autre application, comme par exemple un système de gestion de base de données, d'où le nom de *gateway - passerelle*). C'est l'usage le plus courant des programmes CGI.

Un programme CGI peut être écrit dans n'importe quel langage ou presque, pourvu que celui-ci soit :

- Capable de lire le flux de données d'entrée.
- Capable de traiter des chaînes de caractères.
- Capable d'écrire sur le flux standard de sortie
- Exécutable ou interprétable par le serveur.

En effet, un des grands intérêts de l'utilisation de CGI est la possibilité de fournir des pages dynamiques, c'est-à-dire des pages pouvant être différentes selon un choix ou une saisie de l'utilisateur. L'application la plus fréquente de cette technique repose sur l'utilisation de formulaires HTML permettant à l'utilisateur de choisir ou saisir des données, puis à cliquer sur un bouton de soumission du formulaire, envoyant alors les données du formulaire en paramètre du programme CGI...

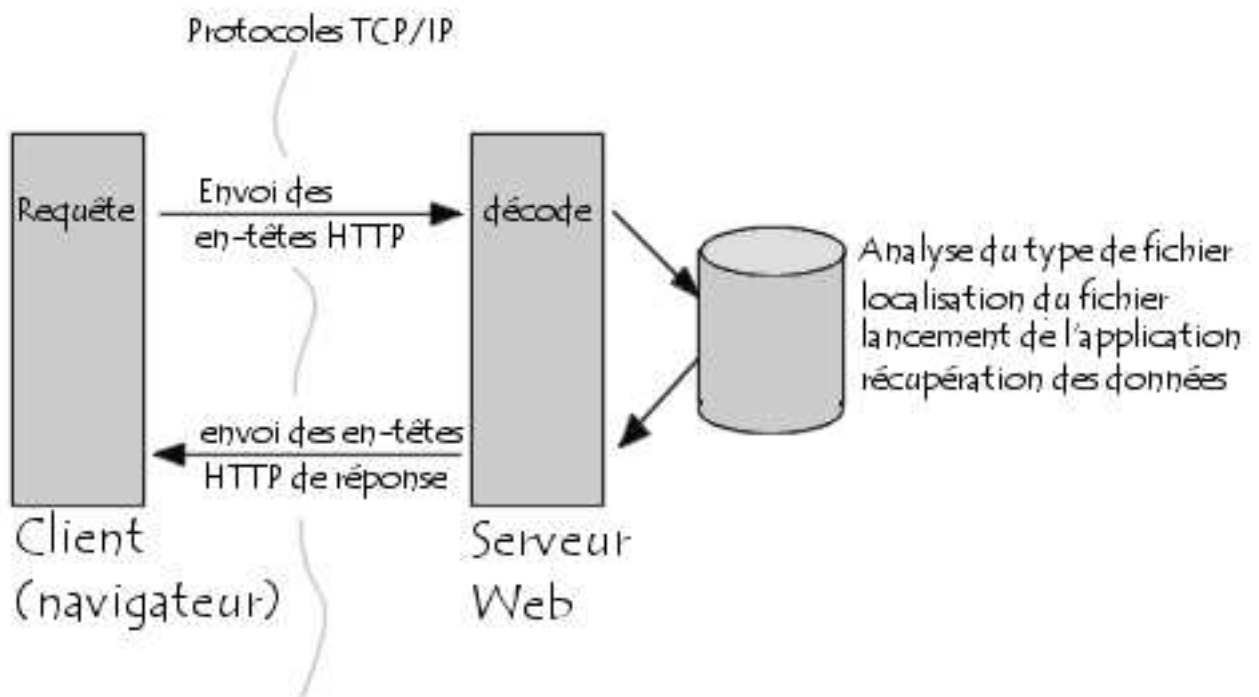


FIG. 5.1 – Exécution *côté serveur*

Le fait que ces scripts soient exécutés du côté du serveur supprime tout souci de compatibilité au niveau des clients (navigateurs Web), et renvoie le problème du temps d'exécution à la machine serveur.

<sup>2</sup>Common Gateway Interface

Lorsqu'une requête est envoyée au serveur, celui-ci examine l'extension du fichier demandé afin de déterminer l'action qu'il va effectuer. Ainsi, par exemple :

- Dans le cas d'un fichier HTML (extension `.htm` ou `.html`), le serveur va générer des en-têtes de réponse HTTP qu'il va envoyer au navigateur demandant un fichier, puis envoyer tout simplement le fichier HTML.
- Dans le cas d'un fichier CGI (extension `.cgi`), le serveur va exécuter l'interpréteur correspondant, et celui-ci va interpréter le code contenu dans le fichier portant l'extension `.cgi` puis envoyer les données au serveur, qui va les envoyer au navigateur, comme dans le cas d'un fichier HTML.

Lorsqu'un navigateur effectue une requête sur un CGI, les étapes suivantes se déroulent, comme indiqué sur la figure 5.1 :

- Les données de requête sont envoyées au serveur sous forme d'en-têtes de requête HTTP.
- Lorsque le serveur reçoit la requête, il analyse les en-têtes HTTP, notamment celui permettant de localiser le fichier demandé.
- Le serveur va ensuite analyser l'entête du fichier (figure 5.2), puis exécuter l'application associée à ce type de fichier (l'interpréteur *Perl* par exemple), en fournissant à cette application les en-têtes de requête HTTP sous forme de variable d'environnement.
- L'application va donc s'exécuter puis fournir des données de sortie au serveur.
- Le serveur va envoyer au client (le navigateur) un en-tête de réponse valide ainsi que les données créées par l'application.
- A la réception du document, le navigateur affiche le résultat à l'écran.

```
#!/usr/bin/perl -w
use lib qw( /lib/perl );
# For parameters passing
```

FIG. 5.2 – Entête d'un CGI écrit en Perl

Ainsi, le rôle d'un programme CGI est de recevoir des données envoyées par le serveur, à les traiter, puis à renvoyer des en-têtes de réponses et des tags HTML valides. En effet, lorsqu'un programme CGI renvoie un fichier, il doit commencer par envoyer un en-tête HTTP permettant de préciser le type de contenu envoyé au navigateur (appelé type MIME), c'est-à-dire :

- Dans le cas d'un fichier HTML, la chaîne suivante : `content-type : text/html`
- Dans le cas d'un fichier GIF (pour faire de la création dynamique d'images par exemple, comme dans le cas d'histogrammes ou de diagrammes divers, ce qui nous sera utile dans la suite), la chaîne suivante : `content-type : image/gif`

Les langages de programmation les plus utilisés pour l'écriture de CGI sont :

- Les langages *C*, *C++*.
- *Java*.
- Le langage *Perl*.

C'est ce dernier langage, d'ailleurs utilisé de façon majoritaire pour l'écriture d'interfaces CGI, qui sera utilisé dans la suite.

### 5.2.3 Perl

Le choix de *Perl*<sup>3</sup> s'explique tout d'abord par le simple fait que l'outil initial, le *Looking Glass* permettant d'effectuer les requêtes sur les routeurs, est écrit avec ce langage.

*Perl* est un langage de programmation dérivé des scripts shell, créé en 1986 par Larry Wall afin de mettre au point un système de *News* entre deux réseaux. Il s'agit d'un langage interprété, possédant de nombreuses fonctions permettant notamment de :

- Manipuler des chaînes de caractères.
- Manipuler des fichiers.

Le langage *Perl* a un typage faible par rapport à des langages comme le langage *C*, de plus l'allocation de mémoire est prise en charge par l'interpréteur. D'autre part, *Perl* possède une caractéristique intéressante : le dimensionnement automatique des expressions, c'est-à-dire que selon le type de variable une expression sera évaluée différemment. D'autre part, sa gratuité et sa présence sur de nombreux systèmes d'exploitation en ont fait, comme il l'a été dit précédemment, le langage de script privilégié pour la création de CGI permettant de générer au vol des fichiers HTML à partir de requêtes effectuées par des clients Web.

C'est également un langage optimisé pour extraire des informations de fichiers textes et imprimer des rapports basés sur ces informations.

Enfin, l'emploi *Perl* présente certaines similitudes avec l'utilisation de langages étudiés dans le cadre de la formation, tels que *PHP* ou *JSP*.

Les versions les plus répandues de *Perl* sont les versions 4 et 5. C'est cette dernière version qui sera utilisée dans le cadre de ce travail.

### 5.2.4 Feuilles de style et Javascript

Ces deux technologies choisies ont toutes deux l'avantage de présenter une compatibilité maximale avec une majorité de plate-formes et de navigateurs ; l'utilisation de Javascript se justifie principalement par des ajouts de fonctionnalités graphiques mineures et l'interaction de l'utilisateur avec la carte dynamique. Toutefois, il est nécessaire de limiter le plus possible son utilisation, puisque quelques problèmes sont toujours envisageables avec des navigateurs moins répandus, et surtout puisque cette technologie n'est pas normalisée par le W3C.

Quand aux feuilles de style, elles sont normalisées par le W3C selon la spécification CSS2, et permettent d'apporter des facilités de mise en page et de dynamisme grâce à l'utilisation d'éléments cachés.

## 5.3 Mise en œuvre

### 5.3.1 Apprentissage de Perl

Une phase nécessaire à la résolution du problème proposé est celle de l'apprentissage du langage *Perl*, grâce à plusieurs ouvrages mis à disposition par l'équipe de recherche et le laboratoire [4; 2; 5; 1].

---

<sup>3</sup>Practical Extraction and Report Language



### 5.3.2 Simplification de l'arborescence

La première priorité pour la réalisation de la carte dynamique, et donc de la refonte du site *w6.loria.fr* est une phase de simplification de l'arborescence et une factorisation des scripts déjà mis en place. En effet, un des objectifs fixés étant la réutilisabilité du travail définitif, il est nécessaire de regrouper au maximum les fichiers dépendant d'une configuration (selon le routeur concerné).

Cette phase aboutit à une arborescence du site plus claire, et surtout l'obtention d'un unique *Looking Glass*, qui, par le biais de fichiers de configuration dépendant de chaque routeur, permet de superviser l'ensemble du backbone géré au Loria.

### 5.3.3 Mise aux normes XHTML 1.0 et CSS 2

XHTML est une famille de types de documents Web qui étend HTML<sup>4</sup> 4, version d'HTML qui constitue l'actuel standard du langage de publication du World Wide Web. Ce standard est basé sur XML<sup>5</sup>, et est conçu pour fonctionner en accord avec les navigateurs supportant XML.

L'apparition de cette norme, aux contraintes très strictes puisque basées sur XML, a permis la définition d'un standard solide auquel tout navigateur Web se doit de répondre. Le choix de l'utilisation de ce langage s'est donc fait en toute logique, puisque cette norme est l'évolution naturelle des versions successives d'HTML. En écrivant des documents répondant à ce standard, on certifie leur validation XHTML et on en garantit dès lors leur affichage de manière universelle sous toute plate-forme possédant un navigateur supportant ce standard : parmi eux, citons Netscape 6.x, Mozilla, Internet Explorer 5.x, Opera...

### 5.3.4 Mise en place d'une charte graphique

L'établissement d'une charte graphique commune au site de supervision se justifie par une volonté de cohérence, de sobriété et d'ergonomie de l'ensemble. En effet, puisqu'il doit constituer un outil utile à la supervision du backbone IPv6, le site mis en place, et plus particulièrement la carte dynamique de cette plateforme de test, se doit de répondre à des exigences graphiques et de facilité d'utilisation. Les informations présentées doivent être claires et la navigation se doit d'être intuitive.

### 5.3.5 Mise en place de la carte dynamique

Cette phase constitue le cœur du développement, puisqu'il s'agit d'écrire les scripts permettant l'affichage sur une partie du site de la carte dynamique de la plateforme de test.

### 5.3.6 Evaluation

### 5.3.7 Améliorations

---

<sup>4</sup>HyperText Markup Language

<sup>5</sup>Extensible Markup Language

## Chapitre 6

# Résultats des réalisations

### 6.1 Description



FIG. 6.1 – *w6.loria.fr* : accueil

Le site Web *w6.loria.fr* présente tout d'abord à l'utilisateur une page d'accueil succincte (figure 6.1), lui rappelant son type de connection (IPv4 ou IPv6) et les quelques contraintes imposées à son navigateur Web (type et version, résolution d'écran). Un menu est disponible à gauche, réparti en 6 rubriques :

- **Platform**, qui présente la plateforme de test du Loria, à savoir le PIO et le PIR mis en place.
- **Project**, résumant les projets en développement au sein du laboratoire, à savoir l'ensemble d'outils de supervision du backbone IPv6 et un projet annexe, *Amarrage*.
- **Software**, qui met à disposition les différents outils logiciels développés par l'équipe de recherche : *Looking Glass*, évidemment, mais aussi le *JDK IPv6*, permettant l'utilisation de

- sockets IPv6 en *Java*, et *ANEP*<sup>1</sup>, protocole spécifique aux réseaux actifs.
- **Publications**, qui dresse un résumé des ouvrages et différentes publications des membres de l'équipe.
- **Links**, l'habituelle section de liens vers des sites utiles et diverses documentations.
- **Running**, qui permet d'accéder à la carte de France de supervision.

Cette dernière rubrique constitue le cœur du site de supervision. L'utilisateur dispose d'une carte de France où est indiqué l'emplacement des différents routeurs constituant le backbone IPv6 (figure 6.2).

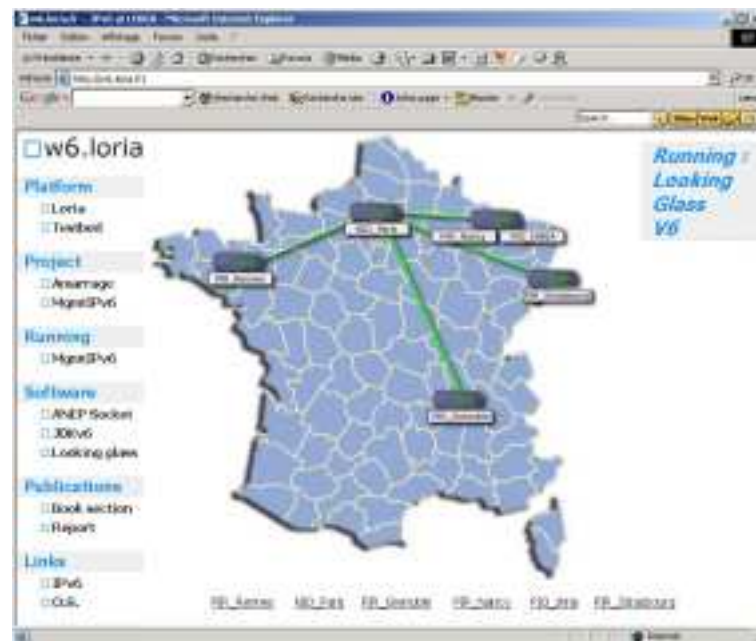


FIG. 6.2 – *w6.loria.fr* : carte de France des routeurs du backbone IPv6

De manière immédiate, l'utilisateur peut visualiser le statut de chaque routeur grâce à un indicateur de couleur, vert si le routeur fonctionne normalement ou rouge dans le cas d'un problème sur le routeur concerné (figure 6.3).



FIG. 6.3 – *w6.loria.fr* : statut d'un routeur

<sup>1</sup>Active Network Encapsulation Protocol

Les liens entre routeurs voisins sont également visibles, par le biais d'un trait de couleur verte si un lien existe.

Lorsque l'utilisateur désire plus d'information concernant un routeur, il lui est possible d'obtenir un résumé des caractéristiques et de statistiques sur chacun d'entre eux, en positionnant la souris sur son symbole. Il apparaît alors un indicateur en surimpression (figure ??) qui donne respectivement :

- Le statut du routeur.
- Le n° d'AS.
- Les adresses d'interface qui lui sont liées (*Global Unicast Addresses*)
- Les statistiques quotidiennes du trafic IPv6, en paquets par seconde et en nombre total de paquets.
- L'accès au *Looking Glass*.
- L'accès au détail des statistiques IPv6, ICMP et UDP.

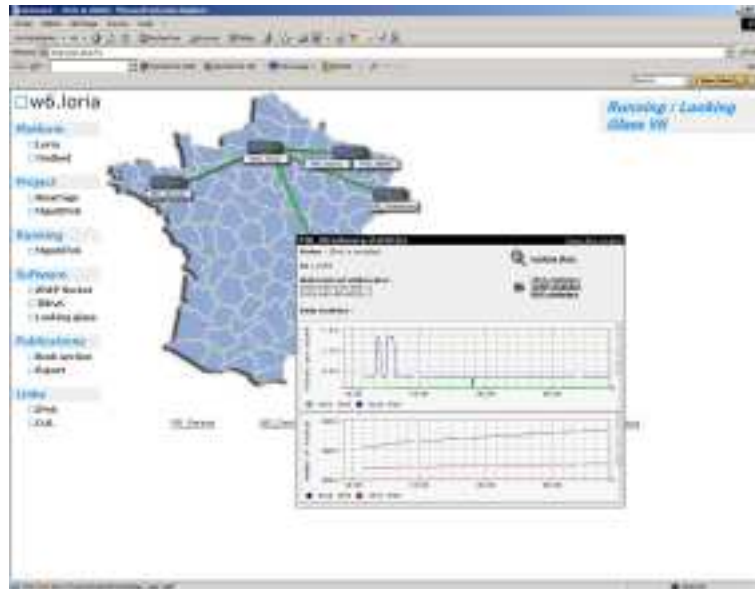


FIG. 6.4 – *w6.loria.fr* : accès aux statistiques d'un routeur

## 6.2 Problèmes rencontrés et limitations

Les premiers problèmes rencontrés furent tout d'abord principalement des problèmes de forme. En effet, la charte graphique et la disposition des différents éléments constituant l'interface ont du être plusieurs fois modifiés afin de convenir à un maximum de types d'affichages, en termes de résolutions, de navigateurs et de type d'utilisations.

Notamment, il est encore à regretter un problème de compatibilité avec des navigateurs tels que Netscape 4.x, qui implémentent mal les CSS. De plus, seul un panel relativement réduit de navigateurs a pu être testé, étant donné la diversité des plate-formes permettant la navigation.

En outre, ce travail connaît des limitations causées par les choix initiaux des outils de développement, comme l'utilisation de Perl, qui ne permet pas un affichage progressif des résultats qu'il génère.

Enfin, il existe des problèmes causés par les limites imposées au départ : l'utilisation d'un outil tel que Flash, par exemple, aurait pu permettre un affichage plus agréable, et aurait nécessité un investissement de développement moindre ; néanmoins, les inconvénients liés à son format propriétaire et à son caractère non-standard en interdisaient l'utilisation.

# Chapitre 7

## Conclusion

### 7.1 Améliorations possibles

Les principales améliorations qu'il sera possible d'apporter à ce travail concernent l'ajout d'interfaçage avec d'autres outils développés au sein de Resedas ; notamment, l'interfaçage de requêtes SNMP<sup>1</sup>, dont l'adaptation vers IPv6 est en cours, permettrait l'ajout de fonctionnalités intéressantes pour la supervision de la plate-forme de test.

D'autre part, des travaux sont actuellement en cours afin de déterminer de manière dynamique la liste des routeurs constituant le backbone. L'intégration d'une liste qui ne serait plus figée pour la génération de la carte des routeurs permettrait de rendre transparente et automatique la modification de la topologie du backbone.

### 7.2 Intérêt du stage

En conclusion, ce stage a permis principalement de mettre en pratique des connaissances acquises au fil de la formation : connaissances en réseau tout d'abord (fonctionnement d'un routeur, principes de routage...), mais également des connaissances de publication sur Internet et des normes en vigueur (XHTML, CSS).

La pratique d'Apache et de ses modules a également été utile, de même que l'enseignement portant sur l'exécution *côté serveur*, même si c'est *Perl*, dont l'apprentissage a été nécessaire, qui a été mis à profit plutôt que des technologies telles que PHP ou JSP qui permettraient d'obtenir un résultat similaire.

Ces contraintes sur l'environnement de travail ont permis l'intégration dans un contexte professionnel, au sein d'une équipe de recherche dynamique et travaillant sur le développement de nouveaux outils, ce qui aurait été impossible en entreprise privée.

Enfin, la découverte d'IPv6, de ses enjeux et de la mise en oeuvre de ce protocole dans un laboratoire travaillant à la mise au point de cette évolution a présenté évidemment un intérêt aussi personnel que professionnel.

---

<sup>1</sup>Simple Network Management Protocol

# Bibliographie

- [1] CHRISTIANSEN Tom et CHRISTIANSEN Nathan. *Perl Cookbook*. O'Reilly, 1998.
- [2] WALL Larry et CHRISTIANSEN Tom et ORWANT Jon. *Programming Perl*. O'Reilly, 2000.
- [3] THOMAS Dave et HUNT Andy. *Programming Ruby : A Pragmatic Programmer's Guide*. Addison-Wesley, 2000.
- [4] SIEVER Ellen et SPAINHOUR Stephen et PATWARDHAN Nathan. *Perl in a Nutshell*. O'Reilly, 1999.
- [5] SRINIVASAN Sriram. *Advanced Perl Programming*. O'Reilly, 1997.