Institutional Repository - Research Portal
Dépôt Institutionnel - Portail de la Recherche
researchportal.unamur.be
University of Namur

THESIS / THÈSE

MASTER EN SCIENCES INFORMATIQUES

Etude expérimentale du protocole BGP

Vandesteene, Christine

Award date:
2003

Link to publication

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Download date: 02. May. 2026

Facultés Universitaires Notre-Dame de la Paix, Namur
Institut d'Informatique

Année académique 2002-2003

Etude expérimentale du protocole BGP

Christine Vandesteene

Mémoire présenté pour l'obtention du grade de Licenciée en Informatique

Juin 2003

Résumé

L'Internet est composé de milliers domaines qui correspondent approximativement à une compagnie ou à un fournisseur d'accès (ISP). Tous ces domaines sont interconnectés pour former l'Internet global. BGP (*Border Gateway Protocol*) est le protocole qui permet de distribuer les informations de routage entre les différents domaines, en y appliquant des règles qui reflètent des accords politiques ou commerciaux. Dans ce mémoire, nous avons évalué les propriétés de convergence d'une implémentation commerciale du protocole BGP au moyen de tests *black-box* : ces tests permettent d'inférer qu'un processus interne s'est déroulé sur base de l'observation d'événements externes. Nous avons commencé par caractériser le processus de décision. Nous avons montré que la durée du processus de sélection de routes pouvait varier en fonction du critère de sélection utilisé, que la durée du filtrage de routes pouvait varier en fonction du nombre de critères examinés, du type de filtre utilisé et par le fait que la politique était appliquée à l'entrée ou à la sortie du routeur. Nous avons montré que l'agrégation de routes influençait fortement le temps de traitement de préfixes spécifiques, et ce d'autant plus que la distance entre la longueur du masque du préfixe spécifique et celle de l'agrégat était grande, mais que ce phénomène n'était pas observé lorsque les préfixes spécifiques appartiennent à un autre espace d'adressage que l'agrégat. Enfin, nous avons évalué l'impact de facteurs externes au processus de décision sur les temps de traitement. Les effets les plus importants ont été observés avec le MRAI et en augmentant le nombre de voisins en aval du routeur.

Abstract

The Internet is a collection of thousands of domains that approximately correspond to a company or an ISP (Internet Service Provider). The Border Gateway Protocol (BGP) is generally used to interconnect all those domains. BGP is able to distribute routing information between those domains and apply them some rules that reflect trading or policy agreements.
In this work, we evaluated convergence properties of a BGP commercial implementation using black-box tests. First of all, we characterized the decision process. We showed that the route selection can take more or less time, depending on the criterion that selects the route. We showed that route filtering duration can vary depending on the number of rules that have to be checked before a route was accepted, depending on the filter's type and depending on the fact that the filter was an input or output policy. We showed that route aggregation was able to dramatically increase the duration of the prefix selection, and that the effect was more important if the distance between specific prefix and aggregate mask length increased, but that this effect was not seen when specific prefixes belong to another address space than aggregate. We finally showed that among other factors, MRAI and number of peers increased the more dramatically the prefix redistribution times.

Avant-propos

Au terme d'un travail de longue haleine, il est de tradition de remercier ceux qui directement ou indirectement y ont participé.

Ce mémoire est l'aboutissement d'une aventure qui a commencé, un peu par hasard lorsque poussée par la curiosité mais aussi par la volonté d'acquérir des compétences nouvelles, j'ai franchi la porte de la LIHD. Les premiers pas en préparatoire m'ont aidée à trouver mes marques dans un domaine qui ne m'était pas familier. Et c'est encouragée tant par l'émulation des autres étudiants que par la satisfaction d'avoir pu surmonter des épreuves à chaque fois plus complexes que je suis arrivée au terme des cours. Je tiens en particulier à remercier Fernando et Laurent, qui en organisant des séances de réflexion et de révisions collectives nous ont permis d'assimiler plus facilement et de manière très agréable les matières vues au cours. Mais la fin des cours ne signifie pas nécessairement l'aboutissement du parcours : le mémoire constitue à lui seul un énorme défi à relever.

Je remercie le Professeur Olivier Bonaventure d'avoir accepté de me superviser alors que je n'avais pas le profil du génie en informatique et de m'avoir incitée à persévérer jusqu'au bout quelles que soient les circonstances ou mes hésitations. Dans le terme supervision, j'englobe à la fois les conseils précieux, la promptitude à répondre à mes questions ou à mes remarques quelles qu'elles soient, la mise à disposition du laboratoire ou encore la recherche des conditions matérielles les plus propices à la réalisation de mon travail. Dans la foulée, je remercie également l'équipe d'Infonet, pour sa bienveillance et son accueil chaleureux, et en particulier Steve et Bruno pour leur soutien spontané, leurs conseils et leur aide efficace, ainsi que Sébastien, qui s'est montré un concurrent loyal pour le partage du labo.

Enfin, je tiens à remercier tous ceux qui, bien que n'étant pas directement impliqués dans ce travail, m'ont soutenu d'une manière ou d'une autre au cours de cette période.

Christine

Table des matières

GLOSSAIRE 1

INTRODUCTION 3

PRÉSENTATION DE BGP 5

1. L'INTERNET AUJOURD'HUI 5

1.1. LES SYSTÈMES AUTONOMES 5
1.2. GESTION DE L'ESPACE D'ADRESSAGE 6
1.2.1. L'adressage dans IP 6
1.2.2. Le CIDR 7
1.2.3. Allocation d'adresses 7
1.3. LA TENDANCE ACTUELLE 7

2. LE ROUTAGE DANS INTERNET 8

2.1. LE PROTOCOLE IP 8
2.2. LES PROTOCOLES DE ROUTAGE 9
2.2.1. Protocoles de routage à vecteurs de distance 10
2.2.2. Protocoles de routage par information d'état des liens 10
2.2.3. Protocole de routage à vecteur de chemin 11

3. LE PROTOCOLE BGP 12

3.1. LES MESSAGES BGP 12
3.1.1. Message OPEN 12
3.1.2. Message UPDATE 13
3.1.3. Message NOTIFICATION 14
3.1.4. Message KEEPALIVE 14
3.2. ETABLISSEMENT D'UNE SESSION BGP 14
3.3. CONTRÔLE DE LA REDISTRIBUTION DES ROUTES PAR BGP 15
3.3.1. Modélisation d'un routeur BGP 16
3.3.2. Les attributs de chemin 17
3.3.3. Le processus de décision 19
3.3.4. Filtrage de routes et manipulation d'attributs 20
3.4. BGP INTERNE OU BGP EXTERNE 20
3.5. AGRÉGATION 21
3.6. PEER GROUP 21

4. MISE EN ŒUVRE DE BGP 21

4.1. DÉPLOIEMENT DE SITES MULTI-CONNECTÉS 21
4.1.1. Site multi-connecté avec un seul fournisseur 22
4.1.2. Site multi-connecté avec plusieurs fournisseurs 24
4.2. DÉTERMINATION DE POLITIQUES DE ROUTAGES 25
4.3. UTILISATION DE BGP PAR UN FOURNISSEUR 26
4.3.1. Agrégation 26
4.3.2. Filtrage du trafic de transit de ses clients 26

MATÉRIELS ET MÉTHODES 27

5. MISES AU POINT DES TESTS BLACK-BOX 27

5.1. PRINCIPE DES TESTS 27
5.2. GÉNÉRATION DE MESSAGES BGP 28
5.2.1. Caractérisation des messages BGP 28
5.2.2. Description générale de l'outil 29
5.2.3. Production d'une description texte des messages 29
5.3. ENVOI DE MESSAGES BGP 30
5.3.1. Gestion des sessions multiples avec sbgp 31
5.3.2. Gestion de l'intervalle de temps entre les UPDATE 32
5.4. MESURE DU TEMPS DE TRAITEMENT 32
5.4.1. Mesure du temps de passage des messages 32
5.4.2. Extraction des informations 32
5.4.3. Calcul de la convergence 32

5.5. Contrôle du déroulement de l'expérience 33
5.5.1. Dispositif de test *33*
5.5.2. Configuration des routeurs et des PC *34*
5.5.3. Conduite des tests *34*
5.6. Conclusion 36

6. PRÉSENTATION DES RÉSULTATS 36

RÉSULTATS ET DISCUSSION 39

7. MESURES DE CONVERGENCE AVEC UN SEUL PAIR EN AVAL DU DUT 39

7.1. Le processus de sélection de routes 39
7.1.1. Redistribution d'annonces à un voisin EBGP *39*
7.1.2. Autres conditions de redistribution *41*
7.2. Effet du filtrage de route et de la manipulation d'attributs 42
7.2.1. Influence du nombre de critères *43*
7.2.2. Comparaison de divers outils *44*
7.3. Effet de l'agrégation de routes 45
7.3.1. Agrégation de son propre espace d'adressage avec un masque long (/21) *46*
7.3.2. Comparaison de différentes conditions d'agrégation *48*
7.4. Influence du MinRouteAdverTimer (advertisement interval) 50
7.5. Influence du Holdtimer 52
7.6. Influence de divers paramètres de configuration 53
7.7. Influence de la taille de la table de routage 56
7.7.1. Initialisation du DUT avec un sous-ensemble des préfixes testés *56*
7.7.2. Initialisation du DUT avec des préfixes différents des préfixes testés *58*
7.8. Discussion 59

8. TESTS AVEC PLUSIEURS PAIRS EN AVAL DU DUT 61

8.1. Influence du nombre de pairs 62
8.2. Effet de la commande peer-group 63
8.2.1. Sur les tests de base *63*
8.2.2. Lors de l'application de politiques de routage *64*
8.3. Discussion 65

CONCLUSION 67

BIBLIOGRAPHIE 69

ANNEXES 71

A. DESCRIPTION DU LABORATOIRE 71

B. COMMANDES RÉSEAU SOUS LINUX 71

C. MRT 72

D. CONFIGURATION D'UN ROUTEUR BGP 75

E. GEN_PREF 78

F. INIT_TABLE 82

G. GREP_CAPTURE 85

H. CALCULE 88

I. CONFIGURATION DE BASE DES ROUTEURS 90

J. SÉQUENCE D'ÉVÉNEMENTS AVEC "AGGREGATE-ADDRESS AS-SET" 91

K. QUANTIFICATION DES PROCESSUS INTERNES 93

Glossaire

AS : *Autonomous system*. Système autonome. Ensemble de réseaux sous une même autorité administrative, qui présentent une politique de routage unifiée par rapport aux autres AS. Un système autonome peut être composé de plusieurs domaines de routage différents.

CIDR : *Classless Inter-Domain*. Routing. Routage inter-domaine sans classes. Fait référence au fait que la longueur du masque d'une adresse IP est indiquée explicitement dans le préfixe et n'est plus déduite implicitement sur base de la valeur du premier octet de l'adresse. Le CIDR a permis de limiter le gaspillage d'adresses IP de classe B en permettant le regroupement d'adresses IP de classe C contiguës pour les réseaux comportant plus de 256 et moins de 65000 hôtes.

Convergence : Fait référence au moment où le réseau entier est mis à jour par rapport au fait qu'une route particulière est apparue ou a disparu.

Domain : Domaine. Collection de routeurs utilisant un même protocole de routage interne (OSPF, RIP).

Forwarding : Transmission. Quand un nœud reçoit un paquet sur une interface, il cherche l'adresse de destination dans sa table de transmission (*forwarding table*) et copie le paquet sur l'interface qui le rapproche de sa destination. Sélection d'un port de sortie sur base de l'adresse de destination et du contenu de la table de transmission. Voir : routing, switching.

Host :Hôte. Ordinateur dont l'usage principal est destiné à l'utilisateur final ou comme serveur (serveur de fichiers, serveur Web, ...). Tout le trafic qui arrive à un hôte lui est destiné. Voir : router, node.

IGP : *Interior Gateway Protocol*. Protocole de routage interne ou intra-domaine.

EGP : *Exterior Gateway Protocol*. Protocole de routage externe ou inter-domaine.

Mask : Masque. Le masque sert à distinguer explicitement la partie réseau de la partie hôte dans les adresses IP. Il est constitué d'une série de 1 contigus, qui servent à représenter le réseau, suivie d'une série de 0 contigus qui servent à représenter l'hôte.

Network : Réseau. Infrastructure de liens et de routeurs. Groupe particulier de nœuds qui ont des adresses IP apparentées. Cette deuxième signification réfère au résultat de l'adressage et est synonyme de « prefix ».

Next-hop : Saut suivant. Réfère au nœud qui rapproche un paquet de la destination souhaitée. Voir : router, node.

NLRI : *Network layer reachability information*. Manière dont BGP représente les adresses réseaux , sous forme de paires <longueur, prefix>, afin de spécifier la longueur du masque de sous-réseau. Cela lui permet de s'adapter au routage inter-domaine sans classe (CIDR).

Node (end node, intermediate node) : Noeud (terminal, intermédiaire). Fait référence au rôle qu'un périphérique joue par rapport à un flux de trafic particulier. Dans un échange unicast, on a deux nœuds terminaux, et n'importe quel nombre de nœuds intermédiaires. A titre d'exemple, un routeur peut terminer une connexion de la couche application, lorsqu'on y travaille par Telnet. Voir : next-hop, router, host.

Prefix : Préfixe. Permet de représenter un bloc d'adresses IP sous la forme d'une adresse 32 bits et d'un masque. Par exemple, le préfixe 192.0.2.0/24 correspond aux adresses IP 192.0.2.0 à 192.2.0.255.

Protocol : Protocole. Ensemble de règles qui permettent de définir le mode de communication entre deux entités paires, matérielles ou logicielles. Les protocoles peuvent servir de support entre applications différentes (par exemple : standards dérivés de XML).

Router : routeur. Machine utilisée principalement pour transmettre le trafic entre réseaux. Le trafic qui arrive sur un routeur est essentiellement destiné à d'autres nœuds ; le travail du routeur est de l'acheminer vers un autre nœud plus proche de sa destination. Voir : next-hop, node, host.
Routing : Routage. Processus de décision qui permet à un nœud intermédiaire de choisir un ou plusieurs chemin pour une destination sur base de l'analyse des informations reçues de sources variées. Ce chemin sera optimal, ou cohérent avec des politiques techniques ou administratives. Le processus de décision utilise une table de routage, qui inclut toutes les informations que le nœud connaît à propos de la topologie du réseau en termes de liens et de localisation de la destination. Processus par lequel la table de transmission est construite. Voir : forwarding.

Subnet : Sous-réseau. Subdivision des réseaux de classe A, B et C, qui utilise une partie de l'espace d'adressage réservée normalement à l'hôte. Cette subdivision est rendue possible en utilisant un masque de sous-réseau qui permet de distinguer explicitement la partie réseau de la partie hôte.

Supernet : Super-réseau. Ensemble de blocs d'adresses contigus qui peut être représenté sous la forme d'une seule adresse. Le masque qui permet de séparer la partie réseau de la partie hôte doit être indiqué explicitement.

Switching : Fait référence à la transmission au niveau de la couche 2, utilisée dans les réseaux à relais de trame (*frame relay*) et ATM. Utilisé dans le cadre du protocole IP, ce terme indique une augmentation de vitesse. Voir : Forwarding.

Introduction

En quelques années, Internet est devenu un composant essentiel du système de communication : un nombre croissant d'applications reposent sur l'accessibilité des réseaux et la fiabilité de leur infrastructure. Dans ce contexte, il est important de comprendre le fonctionnement des protocoles et acteurs principaux qui assurent le maintien de la connectivité du système.

L'Internet est composé de milliers de domaines qui correspondent approximativement à une compagnie ou à un fournisseur d'accès (ISP). Tous ces domaines sont interconnectés pour former l'Internet global. BGP (*Border Gateway Protocol*) est le protocole qui permet de distribuer les informations de routage entre les différents domaines, en y appliquant des règles qui reflètent des accords politiques ou commerciaux.

L'objectif de ce mémoire est d'essayer de comprendre et caractériser le fonctionnement du protocole BGP par la pratique. Théoriquement, BGP est un protocole assez simple à comprendre : il est caractérisé par le format de ses messages et un ensemble d'attributs, et les décisions de routage sont basées sur la valeur de ces attributs. Cependant, en pratique, on remarque que des décisions de conception prises par un domaine, telles que la volonté de bénéficier de solutions de redondance ou de répartition de charge, peuvent avoir un impact important sur le fonctionnement de l'Internet entier. De plus, chaque domaine peut appliquer ses propres politiques de routage indépendamment des autres domaines, ce qui peut être source de conflit et de contradictions.

L'analyse d'un protocole de routage peut consister à mesurer les performance de la transmission des paquets de données (*forwarding*) ou à évaluer le temps nécessaire pour retrouver une vue stable et cohérente du réseau après un changement de topologie (*convergence*). La caractérisation du fonctionnement de BGP peut se faire au niveau de l'Internet entier, d'un seul domaine ou d'un routeur isolé et impliquer des mesures réelles ou l'utilisation de simulateurs. La caractérisation d'un protocole de routage peut se faire au moyen de techniques *white-box*, qui nécessitent un marquage temporel à l'intérieur du routeur, ou au moyen de tests *black-box*, qui infèrent qu'un événement interne s'est produit sur base de l'observation d'un événement externe qui résulte normalement de la réalisation de l'événement interne.

Nous avons choisi d'évaluer les performances de convergence d'une implémentation commerciale de BGP au moyen de tests *black-box*. Ce choix a été guidé par le constat que même si les spécifications du fonctionnement interne d'un routeur BGP sont bien connues, certaines sont exprimées de manière ambiguë et leur implémentation est libre. L'implémentation testée étant propriétaire, il ne nous était pas possible de modifier le code pour réaliser des tests *white-box*.

Pour réaliser ce travail, il a été nécessaire, dans un premier temps, d'acquérir un certain nombre de compétences techniques : manipulation et configuration de routeurs, utilisation de systèmes d'exploitation de type Unix (Linux), apprentissage d'un langage adapté à l'extraction d'informations à partir de fichiers textes et à l'écriture de scripts pour automatiser les procédures (Perl). Nous avons ensuite recherché les outils les plus appropriés à l'analyse du trafic réseau (`tethereal`), à la production de séquence d'événements propres à BGP (MRT), et à la configuration des routeurs en mode non interactif à partir de scripts (modules du CPAN) afin d'automatiser les procédures, et nous avons adapté ces outils à nos propres besoins. Enfin, nous avons essayé d'utiliser au mieux les diverses possibilités de configuration de BGP pour pouvoir tirer le maximum d'informations possibles à propos du fonctionnement interne du routeur sur base des seules observations *black-box*.

Les grandes lignes de ce document sont les suivantes. Dans la partie présentation de BGP, nous décrirons le fonctionnement de BGP. Le chapitre 1 décrit les caractéristiques générales du fonctionnement actuel de l'Internet. Le chapitre 2 rappelle le fonctionnement des protocoles de routage. Le chapitre 3 détaille le fonctionnement du protocole BGP au niveau d'un routeur. Le chapitre 4 donne quelques exemples de déploiement du protocole BGP.

Dans la partie matériel et méthode, nous présenterons le dispositif que nous avons mis au point pour réaliser les tests black-box, ainsi que la méthode suivie pour présenter les résultats. Le chapitre 5 décrit la mise au point des tests black-box. Le chapitre 6 décrit la méthode que nous

avons suivie pour présenter les résultats de manière synthétique et comparative, afin d'en extraire les enseignements ad-hoc.

Enfin, la partie résultats et discussion nous permet de présenter les enseignements que l'on peut tirer de l'utilisation des tests black-box. Le chapitre 7 présente les résultats obtenus dans une configuration simple, où le routeur testé ne dispose que d'un seul pair en aval. Le chapitre 8 nous montre que l'augmentation du nombre de pairs dans une configuration peut avoir des effets inattendus.

Présentation de BGP

1. L'Internet aujourd'hui

En quelques années, Internet est devenu un composant essentiel du système de communication : il permet de connecter à grande échelle de multiples réseaux et ordinateurs et un nombre croissant d'applications reposent sur l'accessibilité des réseaux et la fiabilité de leur infrastructure. Dans ce contexte, il est important de comprendre le fonctionnement des protocoles et acteurs principaux qui assurent le maintien de la connectivité du système.

La plupart des réseaux informatiques se basent sur les protocoles TCP/IP, dont le cœur se situe dans les couches réseau et transport. La fonction principale de la couche réseau est l'acheminement des paquets de données entre une source et une destination en passant si nécessaire par une série de nœuds intermédiaires. Le routage est un processus de décision qui permet à chaque nœud de choisir un chemin vers une destination sur base des informations qu'il connaît à propos de la topologie du réseau.

L'Internet est composé de milliers de domaines qui correspondent approximativement à une compagnie ou à un fournisseur d'accès (ISP). Chaque domaine peut utiliser un protocole de routage interne différent (RIP[1], IS-IS[2], OSPF[3]), afin de constituer une table de routage qui permet de déterminer le chemin le plus court pour atteindre chaque destination connue dans le domaine. Tous ces domaines sont interconnectés pour former l'Internet global. BGP (*Border Gateway Protocol*) est le protocole qui permet de distribuer les informations de routage entre les différents domaines, en y appliquant des règles qui reflètent des accords politiques ou commerciaux. Depuis son introduction, au début des années 1990, BGP a fortement évolué, pour s'adapter à la croissance de l'Internet. Ainsi, BGP-4, le standard actuel en matière de routage interdomaine, a introduit l'utilisation des masques de longueur variables, ce qui a permis de ralentir la demande d'adresses IP de classe B et de contenir la croissance des tables de routage de l'Internet.

Actuellement, Internet est confronté à de nouveaux défis. Le premier consiste à contenir son taux de croissance pour ne pas dépasser la capacité de traitement de ses routeurs. Le second consiste à pouvoir répartir au mieux le trafic de données au moyen d'un protocole qui a été conçu initialement pour assurer une connectivité de type « best-effort ». Enfin, il doit fournir un routage stable, qui s'adapte aux défaillances dans un délai raisonnable, et qui fournit des performances acceptables.

1.1. Les systèmes autonomes

Internet est organisé en un ensemble de systèmes autonomes, ou AS (*Autonomous Systems*), qui sont sous l'autorité administrative de différentes organisations, par exemple des universités, des entreprises, un réseau backbone. En général, ces entités sont politiquement et techniquement indépendantes les unes des autres. Chaque AS gère son réseau et ses relations avec les autres AS (clients, fournisseurs d'accès). Un AS possède un ensemble d'adresses IP non forcément contiguës et est identifié par un nombre de 16 bits, le numéro d'AS, ou ASN (*AS Number*).

De la structuration des réseaux en systèmes autonomes découle une hiérarchie à deux niveaux pour la propagation des routes. Chaque AS utilise son propre protocole de routage interne (IGP) à l'intérieur de ses limites. Un protocole de routage de plus haut niveau, ou protocole de routage externe (EGP), permet le routage entre les systèmes autonomes. Les protocoles de routage interdomaine reflètent les accords entre AS, et permettent de gérer des quantités d'informations importantes.

Les protocoles de routage interdomaine ont été créés pour un Internet hiérarchisé. Ils se focalisent sur l'accessibilité. Les routes ne sont pas nécessairement optimales, mais répondent à certaines politiques.

---

[1] RIP : *Routing Information Protocol*
[2] IS-IS : *Intermediate system to intermediate system*
[3] OSPF : *Open shortest path first*

On distingue plusieurs types de systèmes autonomes, selon le type de trafic qu'ils acceptent de transporter [RFC1772]. L'AS souche (*stub AS*) a une seule connexion avec un autre AS. Il transporte le trafic local uniquement, c'est-à-dire, le trafic qui soit a son origine dans l'AS, soit est destiné à l'AS. L'AS souche multi-connecté (*multihomed stub AS*) a des connexions avec plus d'un AS. Il refuse de transporter le trafic de transit. Le trafic de transit est tout trafic qui a une source et une destination à l'extérieur de l'AS. L'AS de transit (*transit AS*) a des connexions avec plus d'un AS. Il transporte le trafic local et le trafic de transit. Un AS peut accepter le trafic de transit vers ses clients, mais refuser le trafic de transit vers d'autres AS.

1.2. Gestion de l'espace d'adressage

IP est un protocole qui fonctionne depuis des années et qui aurait été victime de son succès s'il n'avait pas pu s'adapter. Aux environs de 1992, la communauté Internet a relevé trois problèmes fondamentaux dans le schéma d'allocation d'adresses originel. En premier lieu, les adresses de classe B, utilisées principalement par les organisations de taille moyenne (qui comptent plus de 256 hôtes) commençaient à se raréfier. Ensuite, la taille des tables de routage dans les backbone Internet montrait une croissance qui ne correspondait plus à la capacité de la mémoire disponible sur les routeurs. Enfin, les adresses IP étaient attribuées sur une base premier arrivé premier servi, sans référence géographique. De ce fait, des adresses adjacentes pouvaient se retrouver à l'intérieur de la même organisation ou sur des continents différents.

Pour résoudre ces problèmes, deux solutions ont été proposées. La solution à long terme, IPv6, est une révision du protocole IP qui étend l'espace d'adresses sur 16 octets, enlève des caractéristiques d'IP peu intéressantes, ce qui rend le protocole plus rapide et plus facile à implémenter, qui intègre des caractéristiques de sécurité et d'authentification et élimine la fragmentation sur les routeurs intermédiaires. La solution à court terme, le CIDR, devait permettre d'utiliser plus efficacement l'espace d'adresses existant et de soulager les tables de routage.

1.2.1. L'adressage dans IP

Dans la version 4 du protocole IP (IPv4), qui est la version la plus couramment utilisée à l'heure actuelle, les adresses IP sont des nombres de quatre octets. Elles sont représentées en notation décimale pointée comme des nombres de quatre chiffres, un pour chaque octet, séparés par un point.

Une adresse IP est divisée en une portion réseau, qui représente le réseau logique auquel l'adresse fait référence, et une portion hôte, qui identifie la machine sur ce réseau. Le routage dans Internet est basé uniquement sur la partie réseau de l'adresse.

Historiquement, les adresses IP étaient groupées en classes sur base des premiers bits de l'octet de poids fort de l'adresse. La classe détermine quels octets de l'adresse se trouvent dans la portion réseau et dans la portion hôte. Les classes A, B et C contiennent les adresses IP régulières. Ce modèle d'adressage est appelé classful.

Actuellement, les systèmes de routage utilisent un masque explicite pour spécifier le nombre de bits assignés à la portion réseau de l'adresse. La séparation ne doit pas nécessairement apparaître à une frontière d'octet : le masque permet de poser une limite arbitraire entre la portion réseau et la portion hôte de l'adresse. Un préfixe est un réseau qui porte un masque explicite. Le modèle d'adressage qui nécessite l'utilisation d'un masque de réseau explicite est appelé classless.

L'utilisation de masques explicites a permis de définir un schéma d'adressage plus fin, appelé le *subnetting*, dans lequel une partie de la portion réservée à l'hôte sert à étendre la partie réseau. Cela a pour effet d'augmenter le nombre de réseaux et de diminuer le nombre d'hôtes possibles dans chaque réseau. En règle générale, les adresses de sous-réseau n'ont qu'une utilisation locale : seul le réseau « normal » est annoncé à l'extérieur.

[Tan99] donne une description détaillée des principes de l'adressage et du *subnetting* dans IP.

1.2.2. Le CIDR

Le routage interdomaine sans classes, ou CIDR (*classless interdomain routing*) élimine le système de classes qui déterminait auparavant les portions réseau et hôte des adresses IP. Tout comme le *subnetting*, dont il est une extension directe, il repose sur un masque explicite pour définir les frontières entre les parties hôte et réseau d'une adresse IP. Mais contrairement au *subnetting*, il permet à la portion réseau d'être plus petite que ce qu'elle aurait été dans la classe naturelle pour faciliter le routage.

L'utilisation d'un masque de sous-réseau plus petit permet d'agréger plusieurs réseaux. Dès lors, le CIDR est parfois appelé *supernetting*.

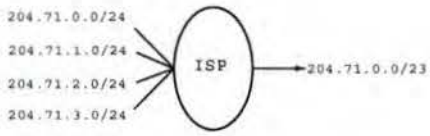

Figure 1-1 : Agrégation d'adresses contiguës avec le CIDR

Avec le CIDR, plusieurs réseaux de classe C contigus peuvent être alloués à un site, sans qu'il soit nécessaire d'avoir une entrée séparée pour chacun. La figure 1-1 nous montre comment un ISP choisit d'annoncer un seul préfixe (avec un masque 23 bits) qui résume quatre préfixes de classe C plutôt que d'annoncer séparément les préfixes spécifiques. Un fournisseur peut ainsi fournir du routage pour des centaines de clients en une seule annonce. Le CIDR permet de résumer les informations de routage de telle sorte que la taille des tables de routage sera réduite tout en maintenant le même niveau de connectivité.

Etant donné que le CIDR permet à des préfixes de longueurs différentes d'être envoyés dans l'Internet, il doit avoir un moyen de gérer les préfixes qui se recouvrent. Le CIDR a formalisé l'idée de la correspondance la plus grande : lorsque la table de transmission d'un routeur contient plusieurs préfixes qui correspondent à une destination, le routeur doit obligatoirement faire suivre le paquet vers le préfixe qui a le plus grand nombre de bits en commun avec cette destination. Par exemple, si la table de transmission d'un routeur contient les entrées <138.48.32.0, eth1> et <138.48.0.0, eth2>, il enverra les paquets de données destinés à 138.48.32.1 via son interface eth1, et ceux destinés à 138.48.33.1 via eth2.

D'après Huston [Hus01], le CIDR a permis de résoudre les problèmes de taille et de taux de croissance des tables de routage, en introduisant une sorte de hiérarchie dans le routage interdomaine. Cela a permis au routage de continuer à se faire avec la même génération de routeurs et d'allonger la durée de vie de l'espace d'adresses, en diminuant la demande d'adresses de classe B.

1.2.3. Allocation d'adresses

A l'heure actuelle, seuls les ISP qui doivent allouer une part importante de leur espace d'adressage peuvent en faire la demande à un registre régional. Les autres doivent demander à leur ISP. Les blocs d'adresses sont délégués à des registres régionaux, responsables de la distribution de blocs aux ISP de leur région. Les ISP divisent à leur tour leurs blocs et les distribuent à leurs clients.

La politique de délégation d'adresses a permis d'utiliser l'agrégation de manière plus efficace.

1.3. La tendance actuelle

Fin 2001, Huston [RFC3221] a constaté que le modèle hiérarchique de connectivité et de routage commençait à montrer des signes d'affaiblissement. Cet affaiblissement est caractérisé par une reprise de la croissance exponentielle de la table de routage (avec un taux de croissance de 42% par an), qui ne s'explique pas seulement par la seule augmentation de la consommation des adresses (7% par an). Il est également caractérisé par une augmentation de la granularité des entrées dans la table (plus de 55% d'adresses

avec un masque de 24 bits), une augmentation de la longueur moyenne des préfixes, une diminution du nombre de préfixes par annonces, et une plus grande dispersion du trafic à travers un plus grand nombre d'entrées, ce qui indique qu'un niveau plus fin de détails de routage est annoncé à l'Internet entier.

Les causes qui pourraient expliquer cette diminution de l'efficience du routage hiérarchique sont une augmentation de l'interconnectivité, provoquée par une augmentation du multihoming à différents niveaux et caractérisée par la perte de la structure hiérarchique, ainsi que la nécessité d'avoir des informations de routage plus spécifiques pour pouvoir réaliser la répartition de charge au niveau interdomaine. [RFC3221]

2. Le routage dans Internet

Le routage est un processus de décision qui permet à chaque nœud de choisir un chemin vers une destination sur base des informations qu'il connaît à propos de la topologie du réseau. Le routage est dynamique : les routeurs mettent constamment à jour leurs tables de routage. Afin d'éviter des pertes de paquets ou des boucles de routage, l'information concernant l'accessibilité des différents réseaux doit être la plus récente possible. On considère que l'information a convergé lorsque tous les routeurs ont en leur possession la même information d'accessibilité pour un réseau donné.

L'architecture de communication TCP/IP utilisée dans l'Internet peut être représentée par un modèle en cinq couches, dans lequel des protocoles pairs peuvent communiquer sans se soucier des détails d'implémentation des couches adjacentes. Chaque couche a une fonctionnalité bien précise.

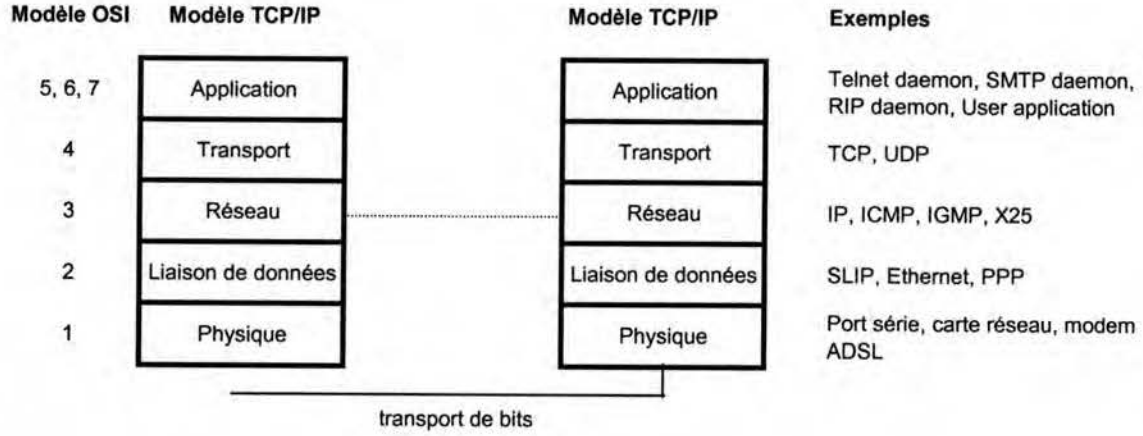

Figure 2-1 : Représentation du modèle de communication en couches

2.1. Le protocole IP

La fonction principale de la couche réseau est d'assurer le transport des paquets de données entre une source et une destination. Les réseaux IP fonctionnent en mode non connecté, ou mode datagramme. Chaque paquet est envoyé indépendamment de son prédécesseur. Il contient toute l'information nécessaire pour être amené à bon port. Un paquet de données peut traverser de nombreux nœuds intermédiaires : la couche réseau est concernée par la transmission de bout en bout du trafic. Pour ce faire, elle connaît la topologie du sous-réseau de communication et est capable de choisir des chemins appropriés à travers ce réseau.

Lorsqu'un hôte (*host*) doit envoyer un paquet de données, il vérifie si le destinataire est accessible directement ou non. Si les deux hôtes sont directement connectés, l'émetteur envoie le trafic directement à la destination. Si le destinataire n'est pas directement connecté, la transmission des paquets se fait saut par saut entre la source et la destination. Quand un nœud reçoit un paquet sur une interface, il cherche l'adresse de destination dans sa table de transmission (*forwarding table*) et copie le paquet sur l'interface qui le rapproche de sa destination.

La figure 2-2 illustre le principe de fonctionnement du protocole IP. PC1 est directement connecté à PC2 : il peut lui envoyer directement le trafic qui lui est destiné. Par contre, s'il veut envoyer du trafic à PC3, il doit d'abord consulter sa table de routage pour vérifier vers quel nœud intermédiaire l'envoyer, en l'occurrence, RTR1.

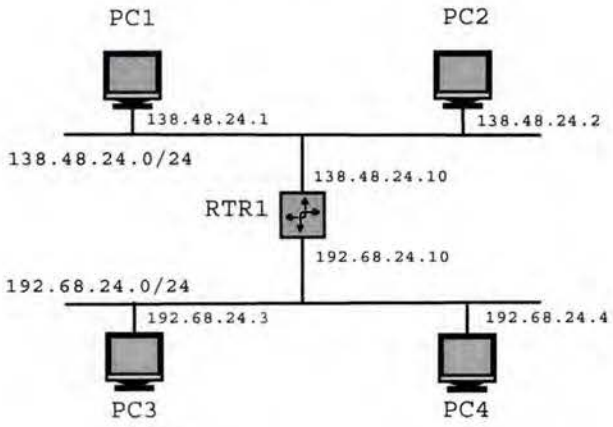

Figure 2-2 : Principe de fonctionnement du protocole IP

Chaque nœud sait à quelle situation il a affaire sur base de l'adressage : les adresses IP de deux hôtes qui sont directement connectés partagent un certain nombre de bits. Dans l'exemple ci-dessus, les trois premiers octets de l'adresse IP de PC1, PC2 et d'une des interfaces de RTR1 sont identiques : ces trois hôtes sont directement connectés. Ce n'est pas le cas de PC3 et PC4. Le nombre de bits qui doivent être en commun dépend du masque du réseau sur lequel le nœud est connecté. La configuration de chaque interface d'un nœud donné comprend au minimum l'adresse et le masque du réseau sur lequel l'interface se connecte.

2.2. Les protocoles de routage

Le rôle joué par les protocoles de la couche réseau est d'assurer que l'information peut s'échanger entre des ordinateurs connectés au réseau. Dans le protocole IP, les données sont acheminées saut par saut entre la source et la destination. Les routeurs sont des machines utilisées principalement pour transmettre le trafic entre réseaux différents. Le trafic qui arrive sur un routeur est destiné essentiellement à d'autres nœuds.

Pour diriger le trafic qu'ils reçoivent vers un nœud qui le rapproche de sa destination finale, les routeurs utilisent une table de transmission (*forwarding table*). Chaque routeur construit sa propre table de transmission en sélectionnant, parmi les informations qu'il a reçu de ses voisins, le meilleur chemin pour une destination donnée. Ce processus de décision est appelé le routage (*routing*).

De manière simplifiée, on peut considérer que le choix du saut suivant (*next-hop*) s'effectue :

- par configuration statique vers un routeur par défaut. Ce routeur directement connecté sera utilisé pour toutes les destinations qui ne sont pas directement attachées à l'hôte. Le routage statique ne tient pas compte des modifications de routes en temps réel.
- par un protocole de routage dynamique. Le routage dynamique permet de mettre à jour les tables de routage en temps réel.

Les protocoles de routage dynamique peuvent être comparés selon la manière dont ils représentent la topologie du réseau qui est routé. On retrouve deux grandes catégories

traditionnelles : les protocoles à vecteur de distance et les protocoles à information d'état des liens.

2.2.1. Protocoles de routage à vecteurs de distance

Les protocoles à vecteur de distance sont basés sur l'algorithme de Bellman-Ford. [Tan99]. Chaque nœud maintient un ensemble de trois nombres (destination, coût, next-hop). Il connaît également le coût de la ligne qui le sépare de ses voisins.

Les mises à jour de la table de routage se font entre voisins directement connectés. Elles contiennent une liste de paires <destination, coût>, ou vecteurs. Sur base de l'information reçue de ses voisins, le routeur calcule le coût pour atteindre tous les autres points du réseau.

Le routeur fait une mise à jour locale s'il reçoit une meilleure route vers une destination (route de coût moins élevé). Les vecteurs sont mis à jour régulièrement, par la réception de paquets des routeurs voisins.

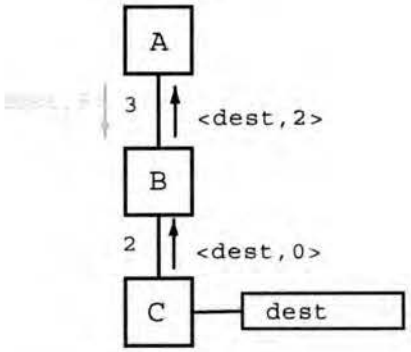

Figure 2-3 : Fonctionnement du protocole à vecteur de distance

Dans la figure 2-3, le routeur C est directement connecté à la destination « dest ». Il annonce à ses voisins qu'il connaît une route pour « dest » avec un coût de 0. Le routeur B calcule le coût réel de la route pour « dest » en ajoutant à l'information qu'il a reçue de C le coût de la ligne qui les sépare. Il installe cette information dans sa table de routage et annonce à ses voisins qu'il connaît une route pour « dest » avec un coût de 0. A son tour, le routeur A procède de la même manière : il ajoute le coût de la ligne qui le sépare de B à l'information qu'il en a reçue et annonce à ses voisins une route pour « dest » avec un coût de 5. En temps normal, B n'utilise pas cette route, parce qu'elle est moins bonne que celle qu'il a apprise directement de C.

Les protocoles à vecteurs de distance sont faciles à comprendre et à implémenter. Mais ils souffrent de désavantages qui limitent leur utilisation à des petits réseaux. En effet, les messages passés entre voisins sont des tables de routage ; ces tables de routage doivent être envoyées à intervalle régulier, ce qui peut devenir gênant lorsque le nombre de préfixe annoncés est important. De plus, en cas de défaillance de lignes, un routeur peut croire qu'une destination reste accessible (comptage à l'infini). Pour pallier à ce problème, on utilise de faibles valeurs pour l'infini, ce qui limite la taille des réseaux.

2.2.2. Protocoles de routage par information d'état des liens

Dans les protocoles de routage par information d'état des liens, les routeurs s'échangent des LSP (*link state packet*). Chaque LSP contient l'identification du routeur qui l'a généré, les routeurs et réseaux auxquels ce routeur est connecté ainsi que le coût pour les atteindre.

Un routeur génère un LSP pour lui-même et l'envoie à ses voisins quand il s'active, quand il connaît un changement de topologie parce que l'état d'une ligne change, ou périodiquement pour rafraîchir des LSP plus anciens. Le mécanisme utilisé pour envoyer

les LSP est l'inondation. Un algorithme vérifie que chaque LSP du routeur est envoyé à chaque routeur du réseau.

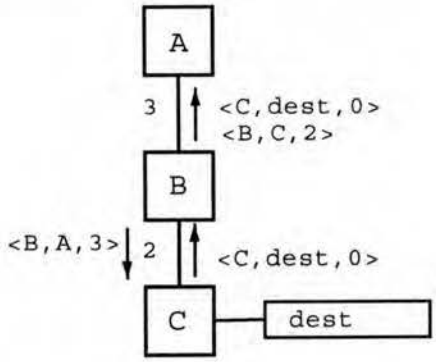

Figure 2-4 : Fonctionnement du protocole de routage par information d'état des liens

La figure 2-4 illustre le fonctionnement du protocole de routage par information d'état des liens. Le routeur C envoie, à destination de tous les routeurs du réseau, un LSP qui indique son identité et qui annonce qu'il connaît une route pour la destination « dest » avec un coût de 0. Par ailleurs, le routeur B envoie à tous ses voisins des LSP qui l'identifient et qui annoncent les routes pour ses voisins directs (A et C), ainsi que le coût pour les atteindre. Chaque routeur reçoit un ensemble de LSP pour le réseau. Il peut alors construire une carte du réseau entier en calculant le meilleur chemin vers chaque destination du réseau.

Avec les protocoles à état des liens, chaque nœud a une information complète sur la topologie du réseau et tous les chemins possibles. L'information qui est transmise est le LSP et non la table de routage complète. Ces protocoles reflètent la topologie : quand il y a une modification, la quantité d'information qui est transmise est proportionnelle au changement de topologie, pas à la taille de la table de routage.

2.2.3. Protocole de routage à vecteur de chemin

BGP utilise un protocole de routage basé sur les vecteurs de distances. Chaque mise-à-jour porte l'indication de tous les systèmes autonomes (AS) qu'un préfixe a traversés depuis son origine, le chemin ou *as-path*.

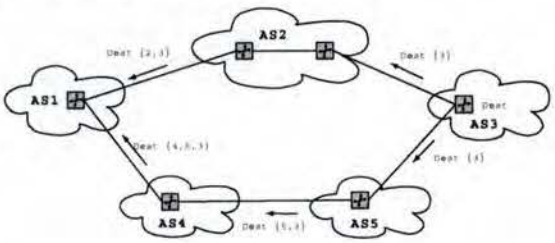

Figure 2-5 : Fonctionnement du protocole à vecteur de chemin

La figure 2-5 illustre le fonctionnement du protocole à vecteur de chemin. Bien que diverses métriques soient possibles, le choix du meilleur chemin est généralement basé sur la longueur de l'*as-path*. L'utilisation de l'*as-path* permet de détecter plus rapidement des boucles de routage.

BGP utilise TCP comme protocole de transport. Cela assure un transport fiable, où les retransmissions sont prises en charge par TCP et ne doivent pas être implémentées dans BGP. Un routeur BGP transmet donc une seule fois sa table de routage à ses voisins, puis transmet les changements quand ils se produisent. Cette amélioration permet de diminuer substantiellement la quantité de trafic.

3. Le protocole BGP

BGP est le protocole de routage inter-domaine, utilisé dans l'Internet pour interconnecter différents systèmes autonomes. C'est le ciment qui permet d'assurer la connectivité globale de l'Internet. Contrairement aux protocoles de routage internes, dont le rôle est de transmettre les datagrammes aussi efficacement que possible d'une source vers une destination, les protocoles de routage interdomaine doivent se préoccuper de stratégies. Les stratégies de routage s'appuient sur des considérations politiques, stratégiques ou économiques. Par exemple, le trafic sortant d'IBM ne doit pas transiter par chez Microsoft, ou le trafic du Pentagone ne doit pas passer par l'Irak.

BGP est un protocole à vecteur de chemin : fondamentalement, il s'agit d'un protocole à vecteur de distance amélioré, qui garde la trace des différents systèmes autonomes qu'une route a traversés depuis son origine. De plus, au lieu de communiquer périodiquement à ses voisins son estimation du poids vers chaque destination possible, chaque routeur indique à ses voisins le chemin exact qu'il utilise. [Tan99].

BGP construit un graphe des systèmes autonomes (AS) sur base de l'information échangée entre voisins. Chaque AS est identifié par un numéro (ASN, ou *AS Number)*. Les connexions entre deux AS forment un chemin, et l'ensemble des informations de chemin forme une route pour atteindre une destination spécifique. BGP assure que le routage inter-domaine est sans boucles.

BGP utilise TCP comme protocole de transport. Cela assure un transport fiable, où les retransmissions sont prises en charge par TCP et ne doivent pas être implémentées dans BGP. Un routeur BGP transmet donc une seule fois sa table de routage à ses voisins, puis transmet les changements quand ils se produisent. Cette amélioration permet de diminuer substantiellement la quantité de trafic.

BGP a connu diverses phases et améliorations depuis sa première version en 1989. Le déploiement de BGP4 a commencé en 1993. C'est la première version qui gère l'agrégation d'adresses, les super-réseaux (supernetting) et le routage interdomaine sans classes (CIDR). Le CIDR est critique pour le fonctionnement d'Internet. Si BGP4, qui a été créé principalement pour supporter le CIDR, n'existait pas, la possibilité de croissance d'Internet aurait été ralentie.

3.1. Les messages BGP

Le protocole BGP peut être décrit par les types et formats des messages qui s'échangent. Quatre types de messages sont possibles : OPEN, UPDATE, NOTIFICATION et KEEPALIVE.

Tous les messages BGP possèdent une en-tête commune, composée de 3 champs. Le champ « Marker » est utilisé, si nécessaire, pour authentifier des messages BGP entrants ou pour détecter une perte de synchronisation entre deux pairs. Le champ « Length » spécifie la longueur du message BGP entier, y compris l'en-tête. Le champ « Type » précise quel message BGP est envoyé.

3.1.1. Message OPEN

Le message OPEN est le premier message envoyé après l'établissement de la connexion TCP. L'objectif de ce message est de permettre aux deux partenaires de la session BGP de s'identifier l'un à l'autre et de se mettre d'accord sur les paramètres du protocole qui seront utilisés. Les champs principaux du message OPEN sont les suivants :

- Version : indique la version du protocole BGP utilisée par l'annonceur.
- My Autonomous System : indique le numéro d'AS du routeur. Internet est une collection de systèmes autonomes identifiés de manière unique par un numéro. Chaque annonceur BGP est configuré avec son propre numéro de système autonome, ainsi que celui de son voisin.
- Holdtime : C'est le temps maximum qui peut séparer la réception de deux messages successifs avant de considérer que la connexion est défaillante. Le Holdtime choisi pour la session sera le minimum entre la valeur configurée localement et la valeur annoncée par le voisin. Si la valeur est 0, la connexion est considérée comme toujours fonctionnelle.

- BGP Identifier : Cette valeur indique l'identité de l'émetteur. En général, il s'agit de la plus haute des adresses IP attribuées au routeur, avec une préférence pour les adresses virtuelles (*loopback*).

Le message OPEN peut également comporter des paramètres optionnels, qui contiennent par exemple des informations d'authentification.

3.1.2. Message UPDATE

Le message UPDATE sert à échanger les informations de routage entre deux partenaires BGP. Ces informations concernent aussi bien l'annonce de nouvelles routes, que le retrait de destinations devenues inaccessibles. L'absence de rafraîchissement périodique impose ce retrait explicite.

Les mises à jour de routage contiennent toute l'information nécessaire pour que BGP puisse construire une image de l'Internet sans boucle. Un message UPDATE comprend trois blocs de base.

UNFEASIBLE ROUTE .

Ce bloc fournit la liste des préfixes IP qui doivent être retirés des tables de routage BGP, soit parce qu'ils ne sont plus accessibles, soit parce que l'annonceur a appris un meilleur chemin vers ce préfixe. Le bloc est composé d'un champ « Unfeasible Routes Length », qui représente la longueur en octets de toutes les routes enlevées, et d'une liste de préfixes représentés sous la forme d'une paire « Length, Prefix », où la longueur indique le nombre de bits du masque de réseau.

PATH ATTRIBUTES.

Ce bloc contient une liste d'attributs BGP associés avec l'information d'accessibilité de la couche réseau. Les attributs de BGP sont une de ses caractéristiques les plus importantes. Ils permettent de décrire la manière dont un préfixe a été routé par BGP, le chemin des AS à travers lesquels il est passé, des métriques qui expriment son degré de préférence ou donnent des informations d'agrégation. Les valeurs prises par les différents attributs permettent de contrôler quels préfixes seront échangés dans une session BGP (filtrage de route) et quel chemin sera utilisé pour atteindre un préfixe particulier (processus de décision).

Chaque attribut est encodé sous la forme d'un triplet <attribute type, attribute length, attribute value>.

Le champ « Attribute Type » permet notamment de définir à quelle catégorie appartient l'attribut. Les attributs de chemin se retrouvent dans l'une des quatre catégories suivantes :

- well-known mandatory : les attributs bien connus et obligatoires doivent être reconnus par toutes les implémentations de BGP. Si un tel attribut manque, un message NOTIFICATION d'erreur sera généré. On retrouve dans cette catégorie l'attribut AS-PATH.
- well-known discretionary : les attributs bien connus et discrétionnaires sont reconnus par toutes les implémentations de BGP, mais ils peuvent ou non être envoyés dans les messages UPDATE. On trouve dans cette catégorie l'attribut LOCAL-PREF.
- optional transitive : les attributs optionnels ne doivent pas être reconnus par toutes les implémentations de BGP. Un attribut transitif qui n'est pas reconnu par une implémentation de BGP sera néanmoins passé aux autres voisins du routeur.
- optional non-transitive : un attribut optionnel non-transitif qui n'est pas reconnu par une implémentation de BGP sera ignoré, et ne sera pas transmis aux autres pairs.

NETWORK LAYER REACHABILITY INFORMATION (NLRI)

Le NLRI décrit l'accessibilité d'une ou plusieurs destinations sous la forme d'une liste de paires « length, prefix ». Cette représentation permet de supporter le routage

interdomaine sans classes (CIDR), étant donné que la longueur représente le nombre de bits du masque qui correspond à la partie réseau de l'adresse.

Les attributs précisés dans un message UPDATE s'appliquent à tous les préfixes du champ NLRI. Si un pair BGP veut annoncer plusieurs préfixes dans un seul message UPDATE, ceux-ci devront avoir tous leurs attributs en commun. A cause de la surcharge de traitement pour envoyer des messages individuels, l'envoi de messages contenant de multiples préfixes est encouragé.

Un message UPDATE peut contenir uniquement des retraits de routes, ou uniquement des annonces de routes ou les deux. Une annonce, ou un retrait, peut concerner 0, 1 ou plusieurs préfixes. Le nombre de préfixes dans la liste est limité seulement par la taille du paquet qui peut être envoyé.

3.1.3. Message NOTIFICATION

Un message NOTIFICATION est émis lorsqu'une erreur est détectée dans le traitement d'un message reçu d'un voisin. Après l'envoi de ce type de message, la connexion TCP est immédiatement arrêtée.

Le message NOTIFICATION est composé d'un champ qui identifie l'erreur rencontrée (expiration d'un timer, type de message dans lequel l'erreur a été détectée), d'un champ , qui donne plus de précisions sur la cause de l'erreur, et d'un champ « Data » optionnel. Il permet aux administrateurs de réseau de comprendre la nature des erreurs du protocole de routage.

3.1.4. Message KEEPALIVE

Les messages KEEPALIVE sont échangés périodiquement entre les pairs pour confirmer que la session BGP entre eux est toujours active. La fréquence d'émission des messages KEEPALIVE dépend du Holdtime négocié lors de l'ouverture de la session BGP et de la fréquence d'émission des UPDATE. Le protocole requiert que des données doivent être envoyées entre les voisins avant l'expiration du Holdtimer.

Les messages KEEPALIVE contiennent uniquement l'en-tête commune.

3.2. Etablissement d'une session BGP

Deux routeurs BGP forment une connexion TCP entre eux. Ces routeurs sont appelés voisins ou pairs (neighbors, peers). Les routeurs pairs échangent de nombreux messages pour ouvrir ou confirmer les paramètres de la connexion, tels que la version de BGP utilisée entre les deux pairs. En cas de désaccord, un message de NOTIFICATION est envoyé, et la connexion entre les pairs n'est pas établie.

Initialement toutes les routes choisies par chacun des pairs sont échangées. Des mises à jour incrémentales sont envoyées lorsque les informations de réseau changent. L'approche incrémentale montre une amélioration énorme par rapport aux mises à jours périodiques complètes, pour la charge du CPU et l'utilisation de la bande passante.

Les routes sont annoncées dans des messages UPDATE, qui contiennent une liste de préfixes accessibles par le système. Le message UPDATE contient également les attributs de chemin, qui incluent des informations telles que le degré de préférence pour une route particulière, l'indication des différents AS qu'une route a traversés depuis son origine,....

Lorsqu'un changement d'information survient (préfixe inaccessible ou réception d'un meilleur chemin), les routes invalides sont retirées, et de nouvelles informations de routage sont injectées.

Une des étapes de base de BGP est l'établissement de la session entre deux pairs. Tant que cette étape n'est pas complètement réalisée, aucun échange d'informations de routage ne peut prendre effet. L'établissement de la session entre voisin est basée sur la réalisation d'une connexion TCP, le traitement du message OPEN, et la détection périodique des messages KEEPALIVE.

La négociation entre voisins passe par plusieurs étapes avant que la session soit établie. Ces étapes peuvent être vues sous la perspective d'une machine à état fini. Les états clés sont les suivants :

- *Idle* (inactif) : est l'état qui précède une tentative d'établissement de connexion. BGP attend un événement qui est normalement initié par un opérateur et le fait transitionner dans l'état *Connect*.
- *Connect* : BGP attend que la connexion TCP soit établie. Si c'est le cas, soit il envoie un message d'ouverture de session BGP et transitionne vers l'état *OpenSent*, soit il reçoit un message d'ouverture de session de son voisin et transitionne vers l'état *OpenConfirm*. Si la connexion TCP n'a pas pu être établie, il passe à l'état *Active*.
- *Active* : BGP essaie d'initier une connexion TCP. Si la tentative réussit, il envoie un message d'ouverture de session BGP et transitionne dans l'état *OpenSent*. Sinon, il retourne à l'état *Connect*, pour écouter après une connexion initiée par un pair. En général, si l'état d'un routeur oscille entre *Connect* et *Active*, cela indique un problème de connexion TCP. (Incapacité d'atteindre l'adresse IP d'un voisin).
- *OpenSent* : BGP attend un message OPEN de son pair. C'est l'étape des négociations et comparaison des numéros d'AS. Si le message est correct, BGP commence à envoyer des messages KEEPALIVE et redémarre le compteur. Sinon, il envoie un message de notification d'erreur, et retourne à l'état *Active*.
- *OpenConfirm* : BGP attend un message KEEPALIVE ou NOTIFICATION. Dans le premier cas, il passe à l'état *Established* : la négociation entre voisins est terminée. Sinon, il repasse à l'état *Idle*.
- *Established* : C'est l'étape finale de la négociation. BGP peut commencer à échanger des messages UPDATE et KEEPALIVE. Si un message NOTIFICATION est reçu ou que le timer expire, il retourne dans l'état Idle.

L'objectif de ces étapes est tout d'abord d'établir la connexion TCP, puis une session BGP et enfin d'échanger les messages UPDATE.

3.3. Contrôle de la redistribution des routes par BGP

BGP est un protocole de routage relativement simple et flexible. Les routes sont échangées entre pairs dans des messages UPDATE. Le routeur peut accepter ou non les préfixes reçus d'un voisin sur base d'une série de critères. Si un préfixe est accepté, et qu'il existe plusieurs routes pour la même destination, le routeur BGP choisit la meilleure route et l'installe dans sa table de routage. Le routeur peut alors l'annoncer à ses voisins : il n'inonde pas ses voisins avec toutes les routes qu'il connaît. Les protocoles de routage externes comme BGP ont été conçus pour permettre d'appliquer diverses stratégies de routage au trafic interdomaine : un routeur BGP ne redistribue pas obligatoirement toutes ses routes à tous ses voisins.

Les clés pour contrôler les informations de routage sont le filtrage de route et la manipulation d'attribut. Nous allons donc examiner chaque attribut BGP pour déterminer ce qu'il fait et comment l'utiliser. Les attributs sont également utilisés par le processus de décision pour sélectionner la meilleure route.

3.3.1. Modélisation d'un routeur BGP

Pour modéliser le fonctionnement de BGP, il faut imaginer que chaque annonceur BGP dispose de différents pools de routes et de différentes machines pour les politiques à appliquer aux routes. [Hal97], [RFC1771]

Les composants du modèle sont décrits à la figure 3-1.

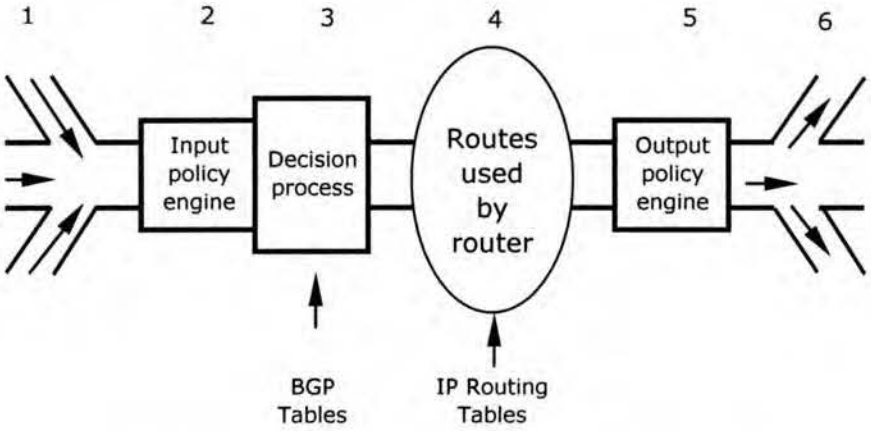

Figure 3-1 : Composants d'un routeur BGP selon Halabi

Chaque routeur BGP maintient trois types d'informations de routage :

- Les routes reçues des pairs [1]. Chaque routeur BGP maintient une base d'information de routage des routes reçues pour chacun de ses pairs (Adj-RIB-In)
- Les routes utilisées par le routeur [4]. Il s'agit des meilleures routes sélectionnée par le processus de décision. Ces routes sont maintenues dans une base d'information des routes locales (Loc-RIB). Elles sont également sont placées dans la table de transmission du routeur. (FIB)
- Les routes annoncées aux pairs [6]. Chaque routeur maintient une base d'information de routage des routes qu'il annonce pour chacun de ses pairs. (Adj-RIB-Out)

Chaque routeur applique des politiques de routage, qui lui permettent de filtrer des route ou de manipuler des attributs d'une route. Ces politiques sont appliquées :

- A l'entrée du routeur [2]. Selon ce qui est configuré comme politique d'entrée, certaines ou toutes les routes reçues d'un pair pourront s'ajouter à la table BGP du routeur. BGP utilise ces politiques pour influencer son propre processus de décision et donc affecter les routes qu'il va réellement utiliser pour une certaine destination.
- A la sortie du routeur [5]. L'application de politiques à la sortie du routeur permet de sélectionner, parmi les routes utilisées par le routeur, celles qui seront réellement annoncées aux pairs. Ce composant est capable de faire la distinction entre pairs internes et externes : ainsi, les routes apprises d'un pair interne ne seront pas passées à un autre pair interne.

Enfin, chaque routeur applique un processus de décision [4] pour choisir les routes qu'il va utiliser pour atteindre une destination particulière. Le fonctionnement du processus de décision est précisé à la section 3.3.3.

3.3.2. Les attributs de chemin

Les attributs de chemin constituent une des caractéristiques les plus importantes de BGP. Ils contiennent la plus grande partie de l'information qui décrit les préfixes routés. Les attributs de chemin sont encodés d'une manière qui permet d'ajouter de nouvelles caractéristiques à BGP sans changer le protocole de base.

3.3.2.1 ORIGIN

Cet attribut bien connu et obligatoire décrit comment l'AS d'origine a appris l'existence du préfixe routé par BGP. Les préfixes n'apparaissent pas au hasard dans BGP mais sont injectés à partir d'une autre source, telle qu'une interface directement connectée, une route statique, un protocole de routage dynamique interne ou un protocole de routage dynamique externe.

Les valeurs possibles de cet attribut sont :

- IGP si l'origine du préfixe est interne à l'AS qui le génère.
- EGP si l'information est apprise via EGP.
- INCOMPLETE si l'information est apprise par d'autres moyens.

3.3.2.2 NEXT-HOP

Il s'agit d'un attribut bien connu et obligatoire qui indique le nœud vers lequel envoyer les paquets de données pour les rapprocher de leur destination.

- Dans le cas des sessions EBGP, l'adresse du NEXT-HOP est celle du routeur qui a envoyé le message BGP.
- Dans le cas des sessions IBGP, si la route a son origine à l'extérieur de l'AS, le NEXT-HOP est l'adresse IP du voisin EBGP qui a annoncé la route. (La distinction entre IBGP et EBGP est explicitée au chapitre 3.4).
- Pour les routes annoncées sur un milieu à accès multiple de type Ethernet, l'adresse du NEXT-HOP est l'adresse du premier routeur du réseau par lequel la route est passée. Sur ce type de média, les routeurs ont la capacité d'échanger des données dans une relation many-to-many. Les routeurs partagent un même sous-réseau IP et peuvent accéder physiquement aux autres routeurs en un seul saut. Dans ce cas, le routeur doit toujours annoncer la source réelle de la route.

3.3.2.3 AS-PATH

Cet attribut bien connu et obligatoire est constitué d'une liste des numéros des systèmes autonomes à travers lesquels une annonce pour un préfixe est passée. Le numéro d'AS de l'annonceur est ajouté à l'AS-PATH chaque fois qu'un préfixe est passé à un pair externe.

BGP utilise l'attribut AS-PATH pour détecter et éviter les boucles de routage. Si un préfixe revient dans un AS qu'il a déjà traversé, le numéro de cet AS se trouve déjà dans l'AS-PATH, et la route sera rejetée.

L'AS-PATH peut se présenter sous deux formes. L'AS-SEQUENCE est un ensemble ordonné qui indique l'ordre dans lequel le préfixe a traversé les différents AS depuis l'origine du chemin. L'AS-SET est un ensemble non ordonné qui inclut l'ensemble des AS qu'une route a traversés. Ce type de liste est utilisé en cas d'agrégation de routes pour conserver l'information relative aux AS d'origine et éviter les risque de boucles de routage. Dans la pratique, les AS-SET sont assez rares : la plupart du temps, l'agrégation est faite sur des préfixes plus spécifiques qui ont des attributs AS-PATH identiques.

L'attribut AS-PATH est encodé comme une séquence de segments d'AS-PATH. Chaque segment peut être soit un ensemble non ordonné d'ASN (AS-SET), soit une séquence ordonnée d'ASN. Les segments d'AS-PATH sont encodés sous la forme de triplets <type, length, value>.

L'information de l'AS-PATH peut être manipulée pour modifier le comportement du routage interdomaine. Etant donné que BGP donne la préférence aux chemins plus

courts, il suffit d'augmenter la longueur du chemin en insérant des numéros d'AS, généralement celui du routeur, pour influencer la trajectoire du trafic.

3.3.2.4 LOCAL-PREF

La préférence locale est un attribut bien connu et discrétionnaire. Elle permet de choisir parmi de multiples chemins pour une même destination, même si ces chemins sont appris d'AS différents.

La préférence locale est habituellement utilisée dans les AS multi-connectés pour choisir le point de sortie qui permet d'atteindre une destination particulière. Cet attribut est le premier critère de décision de BGP : en utilisant la préférence locale, un AS peut contrôler lui-même les points de sortie de son trafic et surcharger d'autres critères de décision comme l'AS-PATH ou le MED.

Cet attribut n'a une signification qu'à l'intérieur d'un AS et n'est donc pas passé aux pairs EBGP.

3.3.2.5 MULTI-EXIT-DISCRIMINATOR (MED)

Le MED est un attribut optionnel non transitif, également connu comme la métrique externe d'une route. Quand deux AS sont connectés par plus d'un lien, il peut être utile de pouvoir choisir le lien optimal pour atteindre un préfixe particulier. Le MED est utilisé pour porter une métrique qui exprime le degré de préférence pour ce lien particulier.

Ce qui caractérise le MED est qu'il est utilisé par un AS pour influencer la décision de sortie d'un autre AS. De cette façon, il est capable favoriser des pratiques non équitables, lorsqu'un AS tente d'utiliser la bande passante d'un autre AS pour atteindre une destination. Pour cette raison, il est plutôt utilisé dans des situations client/fournisseur, pour aider à choisir parmi les chemins multiples appris d'un AS.

On choisit de baser les décisions BGP sur des facteurs externes, comme le MED, pour simplifier la configuration de ses propres routeurs. Mais il est toujours possible pour un AS de surcharger le MED en fixant une valeur de LOCAL-PREF à l'intérieur de l'AS.

L'attribut MED est non transitif : il est échangé entre les AS, mais un attribut MED qui entre dans un AS n'en sort pas.

3.3.2.6 ATOMIC-AGGREGATE

Il s'agit d'un attribut bien connu et discrétionnaire. Il est utilisé pour permettre aux annonceurs de signaler les décisions qu'ils ont prises à propos des routes qui se recouvrent dans un agrégat. Si un routeur reçoit une route avec l'attribut ATOMIC-AGGREGATE attaché, il sait que les chemins vers certains sous-ensembles de l'espace d'adresse décrit par l'agrégat pourraient traverser des AS qui ne sont pas listés dans l'AS-PATH. Cet attribut indique une perte d'information pour des routes agrégées ; il n'est pas utilisé si l'agrégat donne des informations complémentaires sur l'origine des routes agrégées, par exemple avec un AS-SET.

3.3.2.7 AGGREGATOR

Cet attribut optionnel et transitif spécifie le système autonome et l'adresse IP du routeur qui a réalisé une agrégation de préfixes.

3.3.2.8 COMMUNITY

L'attribut COMMUNITY est un attribut optionnel et transitif. Dans le contexte de BGP, une communauté est un groupe de destinations qui partagent des propriétés communes. Cet attribut est de plus en plus largement utilisé pour simplifier les configurations de politiques de routages complexes en identifiant les routes sur base d'une propriété logique plutôt qu'un préfixe IP ou un numéro d'AS. Il offre la possibilité d'associer un identifiant à une route. Ainsi, un ensemble de routes à qui on doit appliquer les mêmes politiques de routage peuvent recevoir le même identifiant.

Certaines valeurs de communautés sont réservées et ont une signification globale dans l'Internet. Les routes qui utilisent la valeur NO-EXPORT ne seront pas annoncées en-

dehors de la confédération ou de l'AS. Les routes qui portent la valeur de communauté NO-ADVERTISE ne seront annoncées à aucun pair BGP.

D'après Quoitin et al. [QB02], l'attribut COMMUNITY est utilisé principalement soit pour marquer les routes reçues d'un type de pair particulier (client, fournisseur...) ou d'une zone géographique spécifique (villes, pays ou continents), soit pour influencer la redistribution des routes par des routeurs en aval. Dans ce cas, l'attribut COMMUNITY peut être utilisé pour indiquer qu'une route ne doit pas être annoncée à un pair spécifique, soit pour indiquer que l'AS-PATH doit être allongé artificiellement (AS-PATH prepending) lorsqu'une route est annoncée à des voisins spécifiques, ou encore pour fixer la préférence locale sur le routeur qui reçoit la route.

3.3.3. Le processus de décision

Une des tâches principales d'un routeur BGP est d'évaluer les différents chemins vers les destinations couvertes par un préfixe, choisir le meilleur, appliquer diverses politiques et l'annoncer à ses voisins. La question est de savoir comment les chemins sont évalués et comparés. Dans les protocoles à vecteurs de distance traditionnels, comme RIP, il y a une seule métrique associée avec un chemin, par exemple, le nombre de sauts. Ainsi, la comparaison de deux chemins se résume à comparer deux nombres. Le problème qui se pose dans le cas du routage entre systèmes autonomes différents est qu'il n'y a pas de métrique universelle qui peut être utilisée pour évaluer les chemins externes. Au contraire, chaque AS peut avoir son ensemble de critères pour évaluer un chemin.

Un annonceur BGP maintient une base de données qui consiste en une liste de destinations accessibles et des différents chemins pour y accéder. La plupart du temps on trouvera un seul chemin pour un préfixe donné. Cependant, si ce n'est pas le cas, tous les chemins accessibles doivent être maintenus, ce qui accélère l'adaptation à la perte du chemin primaire. BGP suit un ensemble de règles pour choisir le meilleur chemin vers un préfixe et l'installer dans sa table de transmission. Le processus de décision se déclenche quand un routeur reçoit un message UPDATE qui contient une annonce ou un retrait de préfixe.

Selon le [RFC1771], le processus de décision peut être découpé en trois phases distinctes. La première sert à calculer un degré de préférence pour la nouvelle route en tenant compte des politiques d'acceptation de routes pré-configurées, la deuxième sert à sélectionner la meilleure route pour un préfixe et à l'installer dans la table de routage BGP, la troisième sert à propager les routes sélectionnées vers les pairs voisins. Si le degré de préférence n'est pas suffisant pour sélectionner une seule route, la deuxième phase prévoit des règles d'arbitrage supplémentaires : l'utilisation de la valeur du MED, la route dont le coût vers le NEXT-HOP selon les métriques internes est le plus bas ou la route pour laquelle l'identifiant du routeur est le plus bas. Le [RFC1772] définit un ensemble de critères qui pourraient être utilisés comme critères de sélection, mais sans spécifier l'ordre dans lequel ils doivent être utilisés. La seule condition à vérifier est que l'application de la fonction qui sert à calculer le degré de préférence doit permettre de définir un ordre strict sur les routes.

Selon [Ste01, Hal97], le processus de sélection de BGP utilise comme entrée l'ensemble de toutes les routes qui ont été acceptées par le système local, qu'elles aient été apprises d'un voisin BGP ou redistribuées à partir d'un autre protocole. S'il y a une seule route vers un préfixe, c'est elle qui sera choisie. Par contre, si le système connaît plusieurs routes vers le même préfixe, il doit utiliser un système d'arbitrage qui permet de décider quelle route utiliser. BGP base son processus de décision sur la valeur des attributs des messages UPDATE. Il vérifie qu'il existe une route vers le NEXT-HOP, et examine les critères suivants l'un à la suite de l'autre jusqu'au moment où il ne lui reste qu'une seule route. Les principaux critères de sélection de routes sont les suivants :

1. La préférence locale (LOCAL-PREF) la plus élevée
2. Une origine locale au routeur (la route générée par la commande `network`, `redistribute` ou `aggregate` est préférée à une route apprise d'un voisin)
3. L'AS-PATH le plus court
4. Le type d'origine le plus petit (IGP<EGP<INCOMPLETE)
5. Le MED le plus bas
6. Une origine externe (EBGP)

7. La métrique de routage interne vers le NEXT-HOP BGP la plus basse
8. Apprise du voisin BGP dont l'identifiant (*RID*) est le plus bas

On peut trouver une version détaillée du processus de décision sur le site de Cisco. [Cisco-DP]. Le processus de décision de Cisco contient en outre des critères de sélection propriétaires, comme le WEIGHT, que nous n'avons pas mentionné ici.

3.3.4. Filtrage de routes et manipulation d'attributs

Le filtrage de routes et la manipulation d'attributs sont des éléments essentiels dans la définition de politiques de routage. Le filtrage de route s'applique en entrée ou en sortie. Un AS annonce à ses voisins les routes pour lesquelles il accepte du trafic entrant. Il choisit les routes que son trafic sortant va utiliser en acceptant les annonces de ses voisins. Le filtrage est également utilisé au niveau du protocole pour limiter les mises à jour de routage qui passent d'un protocole à l'autre lors de la redistribution.

Différents critères permettent de différencier les routes : le préfixe IP, une information contenue dans l'AS-PATH, une valeur spécifique d'un attribut...

Dès qu'une route a été identifiée plusieurs actions peuvent lui être appliquées: la rejeter, l'accepter telle qu'elle ou la soumettre à des modifications. La manipulation d'attributs est utilisée pour influencer le processus de décision. C'est la clé qui permet d'établir des politiques de routage, la répartition de charge et la symétrie de routes.

3.4. BGP interne ou BGP externe

BGP est utilisé en premier lieu comme protocole de routage entre les systèmes autonomes. Reste à savoir comment les préfixes appris par un routeur via une session BGP sont redistribués aux autres routeurs de l'AS.

Une solution possible serait d'injecter dans le protocole de routage interne les préfixes appris d'autres AS via BGP. L'inconvénient d'une telle solution est que les ISP qui doivent transporter des tables de routage complètes ne peuvent le faire, parce que le volume des routes est trop important et le taux de changement trop fréquent pour pouvoir être supporté par un IGP. De plus, la redistribution dans l'IGP peut provoquer une perte des attributs BGP, ce qui pose des problèmes dans le cas des AS de transit.

L'alternative est d'utiliser BGP à l'intérieur du système autonome. Cette variante de BGP appelée IBGP (pour internal BGP) est utilisée entre deux routeurs du même système autonome et répond à l'exigence de distribuer les routes apprises via EBGP (pour external BGP) aux autres routeurs de l'AS.

EBGP et IBGP constituent un même protocole : ils partagent les mêmes types de messages, les mêmes attributs, la même machine à état fini.

Cependant, EBGP et IBGP ont des règles différentes pour ré-annoncer les préfixes : les préfixes appris d'un voisin IBGP ne peuvent pas être annoncés à un autre voisin IBGP. La raison de cette règle est d'empêcher les boucles de routage des annonces à l'intérieur d'un système autonome : le mécanisme qui évite les boucles de routage repose sur l'attribut AS-PATH, qui est mis à jour uniquement quand un préfixe sort d'un AS. Mais elle impose la nécessité d'avoir une connexion directe entre chaque paire de routeurs IBGP à l'intérieur d'un AS. Le maillage complet (*full mesh*) est indépendant des liaisons physiques : les voisins IBGP ne doivent pas nécessairement être directement connectés.

Une des caractéristiques importantes des routeurs IBGP est qu'ils sont capables d'établir un degré de préférence pour les routes qu'ils injectent dans l'AS. Le degré de préférence est inclus dans l'attribut LOCAL-PREF. Tous les routeurs à l'intérieur de l'AS ont donc connaissance du degré de préférence pour les routes que chacun d'eux a injecté dans IBGP. Seule la route avec le degré de préférence le plus élevé sera gardée pour l'ensemble de l'AS.

3.5. Agrégation

Une des améliorations principales apportées par BGP4 est sa capacité à gérer le CIDR et de former des super-réseaux, ce qui permet de contrôler la croissance des tables de transmission IP (*forwarding table*) et la déplétion de l'espace d'adressage IP.

L'agrégation s'applique aux routes qui existent dans la table de routage BGP. Elle peut être réalisée si au moins une route plus spécifique se trouve dans la table de routage BGP.

Selon les besoins, l'annonce de l'agrégat peut s'accompagner ou non de l'annonce des routes spécifiques, ou d'un sous-ensemble des routes spécifiques. Par exemple, lorsqu'un client est multi-connecté à un seul fournisseur, celui-ci peut vouloir utiliser les routes spécifiques du client pour prendre de meilleures décisions de routage quand il lui envoie du trafic. Tout en annonçant seulement l'agrégat au reste de l'Internet, pour minimiser le nombre de routes propagées.

3.6. Peer group

On appelle peer group un groupe de routeurs BGP voisins auxquels on veut appliquer les mêmes politiques. Au lien de définir ces politiques individuellement pour chaque voisin, on définit un nom de peer group et on applique ces politiques au groupe lui-même.

Les peer group permettent d'éviter les configurations répétitives de chaque voisin. Ils permettent également de former une seule fois l'UPDATE, et de l'envoyer ensuite à tous les voisins qui appartiennent au même groupe. Des politiques, telles que le filtrage de routes ou la manipulation d'attributs peuvent s'appliquer aux peer groups. Une fois que les politiques ont été définies, elles sont appliquées à tous les voisins qui appartiennent au groupe.

4. Mise en œuvre de BGP

Il est important que les développeurs comprennent les formats, attributs et règles des messages BGP. Par contre, il est plus important pour les opérateurs de comprendre la manière dont les réseaux qui utilisent BGP sont configurés et la manière dont BGP interagit avec les autres parts du système.

Dans cette section, nous allons illustrer la manière dont les attributs et fonctionnalités de BGP permettent de déployer des stratégies de routage particulières. Les stratégies de routage ne font pas partie du protocole BGP mais sont paramétrables manuellement.

4.1. Déploiement de sites multi-connectés

Le succès croissant d'Internet s'est traduit par une augmentation des besoins en connectivité, ainsi que par une dépendance de plus en plus grande de nombreuses applications vis-à-vis de la connectivité globale.

Une des raisons majeures de l'émergence de sites multi-connectés est la volonté de pouvoir conserver la connectivité Internet même en cas de défaillance d'une ligne ou d'un routeur. De plus, dans un contexte international, le système autonome d'une compagnie peut être vu comme une entité logique qui se répartit sur plusieurs entités physiques, pour lesquelles il peut être intéressant d'avoir des points de sortie locaux.

Si un client dispose de plusieurs liens vers Internet, il pourrait souhaiter que son trafic utilise toutes ses connexions, même si une seule aurait suffi à le transporter. La répartition de charge vise à réaliser une distribution du trafic qui utilise au mieux les liens multiples qu'apporte la redondance. Pour réaliser cet objectif, il est nécessaire de savoir quel trafic on souhaite répartir, entrant ou sortant, et d'estimer l'importance relative de ces deux flux de trafic. En effet, si un site est composé essentiellement de clients HTTP, il y a peu d'intérêt de déployer des configurations complexes pour optimiser le trafic sortant : le client envoie une petite quantité de données au serveur (requête HTTP) et reçoit potentiellement de grandes quantités de données (pages Web avec des graphiques, du texte,...).

A partir du moment où le trafic peut utiliser plusieurs chemins possibles pour atteindre une destination, des questions annexes sont à envisager : la symétrie et le réarrangement des paquets. Le routage symétrique fait référence au fait que le trafic qui quitte un AS par un

point de sortie va revenir par le même point. Dans la réalité, le trafic asymétrique est le plus fréquent, en particulier pour les réseaux qui sont éloignés géographiquement l'un de l'autre, mais les clients souhaitent voir revenir leur trafic à proximité du point où il a quitté l'AS pour minimiser les délais potentiels qu'ils risquent de rencontrer autrement. La question du réarrangement doit être envisagée avec sérieux dans les cas où plusieurs chemins sont utilisés pour atteindre une même destination, si on ne veut pas avoir affaire à une diminution des performances. En effet, un des protocoles de la couche transport, TCP, utilise un certain nombre d'algorithmes pour éviter la congestion et maximiser les performances. L'un d'entre eux, fast retransmit, est déclenché par une livraison hors séquence des paquets de données. Il faut donc faire attention à ce que les paquets d'un flux TCP ne se mélangent pas à cause d'une mauvais conception de l'infrastructure réseau.

Les critères de symétrie et de répartition de charge sont généralement contradictoires. Le déploiement concret de BGP sera différent si on souhaite utiliser un lien principal et un lien de secours, pour maintenir sa connectivité en toutes circonstances tout en conservant un seul point d'entrée et de sortie (symétrie), ou si on souhaite répartir le trafic sur l'ensemble des liens dont on dispose (pour augmenter la bande passante). C'est pourquoi, lorsqu'une solution de redondance est déployée, il est important de définir quels autres critères de conception doivent être pris en compte.

Les attributs BGP sont des outils qui permettent de réaliser les objectifs désirés. Il appartient à l'opérateur de choisir parmi ces objectifs et de configurer les attributs corrects pour y arriver. Les sites multi-connectés sont intrinsèquement compliqués. Ils ne sont pas bien gérés par BGP. Le multihoming interagit de manière complexe avec les politiques d'allocation d'adresses et d'agrégation. Cette complexité doit être reconnue par les organisations qui souhaitent multiplier leurs connexions pour qu'un plan de routage rigoureux soit développé. On distingue deux types possibles de multihoming : avec un seul fournisseur ou avec plusieurs fournisseurs. Dans ce dernier cas, le multihoming sera connu de l'Internet entier. Nous allons montrer, sur base de quelques scénarios que le paramétrage de BGP dépend des critères de conception que l'on souhaite implémenter.

4.1.1. Site multi-connecté avec un seul fournisseur

UTILISATION DE DEUX LIGNES PARALLELES ENTRE LES DEUX MEMES ROUTEURS

Dans ce type de scénario, l'objectif de symétrie est rencontré quoi qu'il arrive, puisqu'on n'a qu'un seul point d'entrée et de sortie de l'AS. On pourrait imaginer utiliser une des connexions comme lien primaire et l'autre comme solution de dépannage : c'est le comportement par défaut de BGP. En effet, en temps normal, lorsqu'un routeur BGP connaît plusieurs chemins pour une même destination, il choisit la meilleure route et l'installe dans sa table de routage IP. Lorsque les routes viennent du même routeur, BGP choisit celle qui vient du voisin avec l'adresse IP la plus basse.

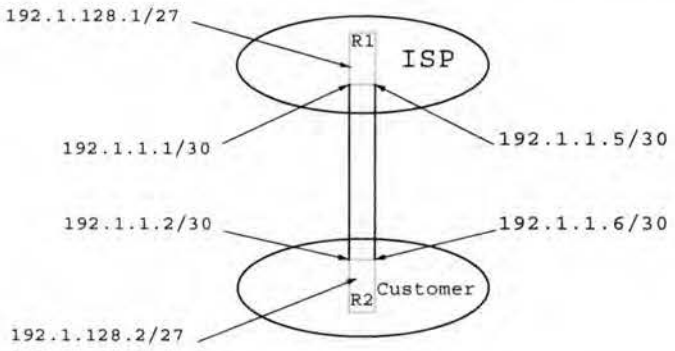

Figure 4-1 : utilisation de liens parallèles entre deux routeurs

Mais il est plus plausible d'imaginer que ce scénario vise plutôt à augmenter la bande passante entre les routeurs, et à répartir le trafic entre les deux liens de la manière la plus équitable possible. La figure 4-1 illustre la solution la plus générale à cette question, qui consiste à utiliser une interface virtuelle (*loopback*) sur chacun des routeurs et à ouvrir une seule session BGP entre ces deux interfaces virtuelles. L'interface virtuelle n'étant pas directement connectée, elle n'est pas accessible directement. Il convient donc de configurer sur chacun des routeurs une route pour chacun des liens que l'on veut utiliser pour atteindre l'interface virtuelle du voisin. Les messages reçus du voisin porteront comme NEXT-HOP l'adresse de son interface loopback. Pour pouvoir installer le préfixe dans sa table de transmission, chaque routeur procédera à un examen récursif de sa table de routage, qui donnera un résultat pour chacune des routes statiques configurées. Si le routeur choisit au hasard un des chemins pour chacun des préfixes qu'il va installer dans sa table de transmission, on peut supposer que les deux lignes seront utilisées.

Cette solution est intéressante si le nombre de préfixes annoncés est élevé et si la même quantité de trafic est échangée pour chacun des préfixes.

En ce qui concerne le trafic client-fournisseur, le degré de répartition de charge dépend du nombre de routes que le client apprend de l'ISP. Si l'ISP annonce uniquement une route par défaut, il est difficile de réaliser du partage de charge pour ce trafic. Le client peut demander à l'ISP d'envoyer sa table de routage complète, ou un sous-ensemble de ses tables de routage qui prend en compte les capacités du client.

CONNEXION DU CLIENT ET DU FOURNISSEUR SANS PARTAGE D'EQUIPEMENT

Ce scénario est supposé être assez fiable étant donné qu'aucun équipement n'est partagé. Une fois encore, la configuration de BGP dépendra des critères de conception que l'on veut implémenter. D'une manière générale, on peut considérer que plus le nombre de préfixes annoncés augmente, plus on a de possibilités de répartir le trafic sur les différents liens qui existent.

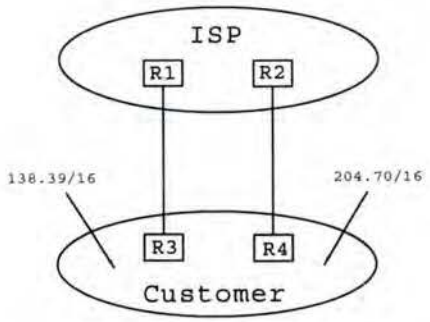

Figure 4-2 : Répartition de charge avec des équipements différents

La répartition du trafic entrant (fournisseur-client) repose sur plusieurs hypothèses. Si le client annonce un seul préfixe, la charge respective de chacun des liens dépend de la proximité de la source du trafic avec chacun des routeurs d'accès du fournisseur.

Si le client annonce plusieurs préfixes, la répartition de charge sera contrôlée plus finement si chaque préfixe est bien localisé à proximité d'un des point de sortie de l'AS du client. Le client annonce ses préfixes sur ses deux points de sortie, pour bénéficier d'une ligne de secours. Il peut contrôler la ligne qui sera utilisée pour le trafic entrant en utilisant un attribut MED différent sur chaque occurrence de chaque route, avec la valeur du MED reflétant la distance qui sépare le préfixe du point d'entrée dans l'AS. Une autre solution consiste à augmenter artificiellement la longueur de l'AS-PATH pour le chemin de secours de chaque préfixe.

Le fournisseur peut, de son côté, configurer des valeurs de préférences locales différentes selon la ligne sur laquelle il apprend les différents préfixes de son client.

Le contrôle du trafic sortant (client-fournisseur) s'effectue de manière inverse. Si le fournisseur annonce un seul préfixe, par exemple une route par défaut, la charge respective de chacun des liens dépendra de la proximité de la source du trafic avec chacun des routeurs d'accès. Si le client souhaite effectuer un contrôle plus fin de la ligne qui sera utilisée par le trafic quittant l'AS, il pourra demander à son fournisseur de lui annoncer sa table de routage complète, ou un sous-ensemble de sa table de routage, en fonction de la capacité de ses routeurs d'accès.

4.1.2. Site multi-connecté avec plusieurs fournisseurs

Les motivations principales pour déployer ce type de connectivité sont les besoins de redondance ainsi que des considérations géographiques.

Selon la manière dont le client obtient son espace d'adressage, on aura des effets variés en terme d'agrégation, de répartition du flux de trafic entre les fournisseurs et de possibilités d'utilisation de lignes de secours.

Prenons l'exemple d'un client connecté à deux fournisseurs, ISP1 et ISP2, qui s'interconnectent entre eux. Ces deux fournisseurs sont également connectés à un troisième fournisseur, ISP3. Le client peut utiliser des préfixes qui lui sont délégués par un seul de ses fournisseurs, par chacun de ses fournisseurs ou obtenus de manière indépendante. Chaque option a ses avantages et ses inconvénients pour différentes parties de l'Internet. Les cas décrits ci-dessous nous serviront à illustrer la manière dont le client peut gérer son trafic entrant.

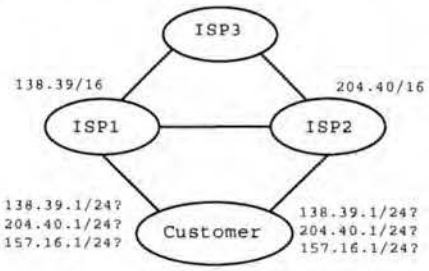

Figure 4-3 : Exemple de site multi-connecté

LE CLIENT UTILISE DES ADRESSES QUI LUI ONT ETE ATTRIBUEES PAR UN SEUL DE SES FOURNISSEURS

Supposons que ISP1 dispose du bloc d'adresses 138.39/16 et que le client utilise le préfixe 138.39.1/24 qui lui a été délégué par ISP1. Etant donné que ISP1 est autorisé à agréger l'espace d'adresses dont il dispose, le préfixe du client sera inclus dans l'agrégat 138.39/16 que ISP1 annonce à ses voisins.

On peut raisonnablement supposer qu'un site connecté à plusieurs fournisseurs souhaite répartir son trafic entre eux. Le client annonce donc également son préfixe 138.39.1/24 à ISP2, bien que celui-ci ne puisse pas l'agréger. ISP2 annonce à son tour le préfixe 138.39.1/24 à ISP1 et ISP3.

Puisque le routage sans classes (CIDR) suit la règle de la correspondance le plus grande pour trouver le meilleur chemin vers une destination, le préfixe le plus spécifique devient une sorte d'aimant pour le trafic de ISP2 et ISP3. Cela peut résulter en une bonne répartition de charge si le client reçoit autant de trafic de ISP1 que de ISP2 et ISP3 réunis.

L'inconvénient de ce genre de pratique est qu'elle permet la diffusion dans l'Internet entier de préfixes spécifiques, bien que ce ne soit pas nécessaire. Pour éviter ce genre de situation et pour encourager une meilleure agrégation dans l'Internet, certains fournisseurs implémentent des politiques de filtrage qui rejettent les routes sur base de la longueur du masque, par exemple, les préfixes avec un masque de plus de 24 bits, ou pour les adresses de classe B, les préfixes avec un masque de plus de 16 bits, ou encore les préfixes qui répondent à des critères définis par les registres Internet régionaux (RIR).

Huston [Hus01], ainsi que Bellovin [BBG+01], ont cependant montré que le filtrage est loin d'être généralisé, et que de nombreux préfixes spécifiques se retrouvent encore dans les tables de routage.

LE CLIENT UTILISE DES ADRESSES QUI LUI ONT ETE ATTRIBUEES PAR CHACUN DE SES FOURNISSEURS

Dans cette deuxième situation, on peut imaginer que le client se contente d'annoncer à chacun de ses fournisseurs les seuls préfixes qu'il en a reçus. Dans ce cas, on obtient de bons résultats en terme d'agrégation, puisque chaque fournisseur peut se contenter d'annoncer son agrégat au reste de l'Internet. Si la quantité de trafic destinée à chacun des préfixes est à peu près égale, on obtient également une bonne répartition de charge.

Le désavantage de cette solution est son manque de fiabilité : en cas de défaillance d'une ligne entre le client et un de ses fournisseurs, le deuxième fournisseur ne sait pas prendre la relève, puisqu'il n'est pas informé des autres préfixes que le client utilise. Le premier espace d'adressage du client devient donc inaccessible.

LE CLIENT UTILISE UN ESPACE D'ADRESSES INDEPENDANT DE CELUI DE SES FOURNISSEURS

Cette solution offre un meilleur contrôle des flux de trafic, puisque l'Internet entier verra le même préfixe, mais cela se fait au détriment de l'agrégation. Pour cette raison, il faut s'assurer que l'espace d'adresse obtenu satisfait aux critères de sélection des nouvelles politiques de routage, sinon le client risque de se retrouver sans connectivité.

Si le client a reçu un espace d'adresse qui lui permet de traverser les filtres de routes, il peut vouloir contrôler le chemin que chacun de ses fournisseurs utilise pour l'atteindre. Par exemple, il peut souhaiter diriger le trafic venant de ISP1 sur la ligne ISP1-client et le trafic venant de ISP2 et ISP3 sur la ligne ISP2-client. Cela peut être réalisé en manipulant l'AS-PATH, qui est allongé artificiellement dans toutes les annonces envoyées à ISP1. L'utilisation d'une valeur de préférence locale appropriée permettra à ISP1 de continuer à utiliser le lien ISP1-client pour son trafic.

4.2. Détermination de politiques de routages

Quand un annonceur BGP reçoit un message UPDATE contenant un certain nombre de routes, il n'est pas obligé de les accepter toutes. De plus s'il accepte une de ces routes, cela n'influence pas la préférence qu'il lui donnera par rapport aux autres routes pour le même préfixe qu'il a appris d'autres sources. Cela signifie que la décision à propos des routes à accepter d'un voisin BGP et la décision à propos des routes à annoncer est locale au routeur.

Cette décision a un impact profond sur le trafic qui traverse un réseau. Les politiques de routage sont utilisées pour refléter les accords commerciaux entre deux ou plusieurs parties.

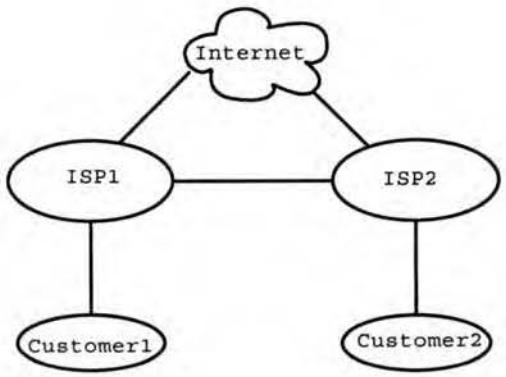

Figure 4-4 : Politique de routage et transit

BGP peut être utilisé pour empêcher le trafic de transit. Le trafic de transit est tout trafic qui a son origine et sa destination en dehors du système autonome qu'il traverse. C'est le type

de service qu'un ISP fournit normalement à ses clients, qui paient pour recevoir ce service. Par contre, un ISP n'est pas obligé d'accepter qu'un autre ISP utilise son infrastructure pour atteindre le reste de l'Internet.

Pour fournir de la connectivité Internet à son client, ISP1 doit faire en sorte que les autres ISP (et leurs clients) puissent atteindre son client, Customer1, en leur annonçant les préfixes de son client. Par ailleurs, il doit s'assurer que son client peut atteindre toutes les autres parties. Pour ce faire, il annonce à son client toutes les routes qu'il a appris de ISP2 et du reste de l'Internet. Par contre, ce qu'il doit absolument éviter de faire, c'est d'annoncer les routes qu'il a appris de ISP2 au reste de l'Internet et inversement, sinon le client de ISP2 pourrait utiliser l'infrastructure de ISP1 pour atteindre l'Internet, ce qui n'est pas souhaité.

4.3. Utilisation de BGP par un fournisseur

Les gros ISP ont des configurations BGP particulièrement compliquées. Les questions suivantes s'appliquent particulièrement à eux.

4.3.1. Agrégation

L'agrégation est un élément critique pour la survie du système de routage de l'Internet. Pour cette raison, il est important que les clients et fournisseurs veillent à configurer leurs réseaux pour permettre le maximum possible d'agrégation.

Les fournisseurs réalisent deux types d'agrégation :

- à l'attention des AS voisins. Seuls les agrégats sont annoncés aux systèmes autonomes voisins, et les routes spécifiques sont filtrée.
- purement interne au réseau du fournisseur. L'agrégat est divisé en sous-agrégats, qui sont assignés à des routeurs d'accès. Quand un client demande une connexion au fournisseur, il reçoit un espace d'adresses basé sur le routeur d'accès auquel il se connecte, et le routage vers les clients à l'intérieur de l'AS du fournisseur se fait sur base de ces sous-agrégats.

4.3.2. Filtrage du trafic de transit de ses clients

Etant donné qu'un fournisseur ré-annonce les routes qu'il apprend de ses clients, il devra faire attention à ne pas accepter n'importe quelle annonce de ses clients. Au contraire, il devra appliquer un filtre aux routes qu'il apprend de ses clients. (routes privées, annonces illégales, routes venant d'autres fournisseurs ...).

Matériels et méthodes

5. Mises au point des tests black-box

5.1. Principe des tests

Il existe plusieurs manières d'étudier un protocole de routage.

Selon Berkowitz, [BHR+02], l'étude peut porter sur l'une des deux fonctionnalités du routage : la transmission des données (*forwarding*), au niveau du plan de données, ou la distribution des informations de routage (*convergence*) au niveau du plan de contrôle.

La caractérisation de la convergence de BGP est un phénomène extrêmement complexe qui peut s'effectuer dans l'Internet entier (domaine de BGP), dans un AS ou sur une machine « isolée » qui reçoit des input sur une ou plusieurs de ses interfaces, et génère des output sur une ou plusieurs de ses interfaces. Dans le contexte des tests, cette machine est appelée le DUT (Device under test).

Ces mesures peuvent être réalisées de manière interne (test *white-box*) en appliquant un marquage temporel à l'intérieur du DUT, ou de manière externe (test *black-box*).

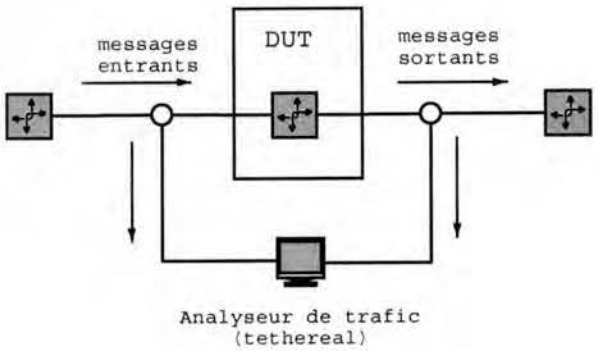

Figure 5-1 : Principe de fonctionnement des tests black-box

La figure 5-1 illustre le principe de fonctionnement des tests *black-box*. Ils consistent à mesurer le temps qui s'écoule entre l'entrée d'une annonce dans le routeur et sa sortie. Les mesures ne doivent pas perturber le fonctionnement normal du routeur : nous utiliserons un analyseur de trafic, **tethereal** [Eth, Teth], qui permet d'écouter le trafic qui s'échange sur le réseau. L'écoute se fera en amont et en aval du routeur testé.

Les mesures externes sont plus facilement applicables que les mesures internes, même si elles incluent des excédents tels que le temps de propagation de l'annonce sur le média utilisé et à l'intérieur du DUT, ou le temps pour annoncer une nouvelle route.

Les mesures externes limitées à la propagation de route facilitent les comparaisons du sous-système de routage sur des machines avec des architectures et des fonctions internes diverses : on peut ainsi négliger l'interaction avec d'autres sous-systèmes de la machine testée.

Nous limiterons notre étude à la re-convergence incrémentale : en effet, la convergence initiale, qui consiste en l'échange des tables de routage complètes lors de l'établissement de la session BGP, est un événement qui se produit rarement pendant la durée de vie d'un routeur, alors que la re-convergence incrémentale se produit fréquemment.

5.2. Génération de messages BGP

La caractérisation de la convergence de BGP au moyen de tests *black-box* nécessite un échange de messages : le DUT reçoit des UPDATE d'un de ses pairs en amont, accepte les préfixes qu'il contient, choisit ces préfixes et les annonces à ses pairs en aval.

Nous utiliserons **sbgp**, un des outils de la suite MRT [Merit] , pour établir une session BGP avec le DUT et lui envoyer des UPDATE BGP stockés dans un fichier. Cet outil a été conçu dans le but de capturer des sessions BGP et les rejouer dans des conditions différentes, mais nous avons préféré générer nos propres messages pour avoir un meilleur contrôle des préfixes annoncés au DUT.

Les messages sont générés sous forme de descriptions ASCII, transformés au format binaire MRT grâce à l'outil **route_atob** [Merit], et envoyés dans une session BGP avec l'outil **sbgp**. La mise au point de l'outil de génération de messages BGP s'est faite en deux étapes. La première consiste à déterminer le format de message utilisable par **sbgp**. La deuxième à reproduire des messages qui possèdent ces caractéristiques.

5.2.1. Caractérisation des messages BGP

Nous avons commencé par vérifier que **sbgp** était capable de capturer des messages BGP comportant les différents types d'attributs (bien connus obligatoires ou discrétionnaires, optionnels).

La figure 5-2 schématise le dispositif que nous avons utilisé pour caractériser les messages capturés par **sbgp**, et nous donne à titre indicatif quelques exemples d'annonces échangées.

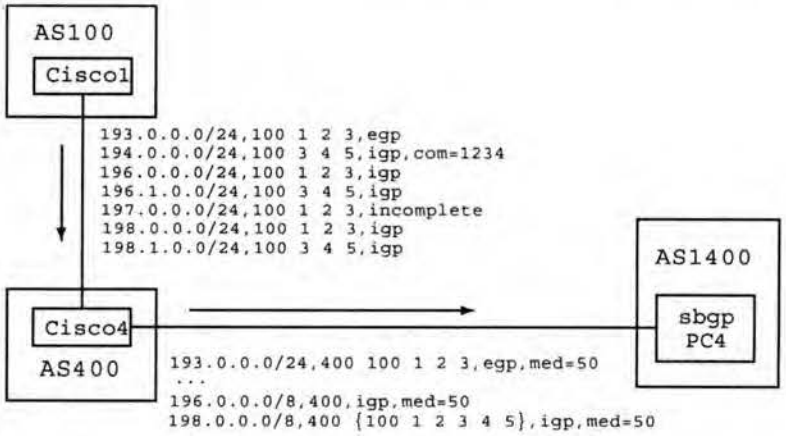

Figure 5-2 : Dispositif utilisé pour caractériser les messages capturés par sbgp

Une session BGP est établie entre Cisco1 et Cisco4 ; Cisco1 annonce à Cisco4 plusieurs préfixes porteurs des attributs qui nous intéressent (différents types d'origine, AS-PATH, COMMUNITY, ...). Cisco4, agrège une partie des préfixe, en utilisant ou non l'option AS-SET.

Une deuxième session BGP est établie entre un PC sur lequel tourne un processus **sbgp** et Cisco4. **sbgp** reçoit les UPDATE de Cisco4 et les sauve dans un fichier binaire. Nous avons exécuté la commande **route_btoa** sur le fichier binaire capturé par **sbgp**, afin d'obtenir les descriptions ASCII de tous ces messages. (voir annexe C3).

L'objectif de cette mise au point est de mettre en évidence la syntaxe exacte des UPDATE capturés par **sbgp** pour les attributs les plus conventionnels (AS-PATH, avec ou sans AS-SEQUENCE, ORIGIN, MED, COMMUNITY, NEXT-HOP, ATOMIC-AGGREGATE, AGGREGATOR, LOCAL-PREF), dans différents cas de figures (session EBGP, IBGP, annonce de routes, retrait de routes).

Le dispositif de test n'avait pas pour objectif de mettre en évidence l'ordre des attributs, mais bien de définir la syntaxe qui permet de les décrire dans un langage compréhensible par les outils MRT. Le [RFC1771] recommande d'ordonner les attributs selon la valeur du code qui correspond à leur type. Nous utiliserons les valeurs décrites dans [Hal97] pour ordonner les attributs de nos messages.

5.2.2. Description générale de l'outil

Les outils de la suite MRT nous permettent de capturer les UPDATE BGP échangés au cours d'une session BGP (**sbgp**), et de les transformer en une description ASCII (**route_btoa**).

La figure 5-3 nous montre qu'il également est possible de transformer une description ASCII « human-readable » de messages BGP en un fichier MRT binaire au moyen de la commande **route_atob**. Ces fichiers binaires sont exploités par **sbgp** pour rejouer ou simuler une suite d'événements qui se sont produits au cours d'une session BGP.

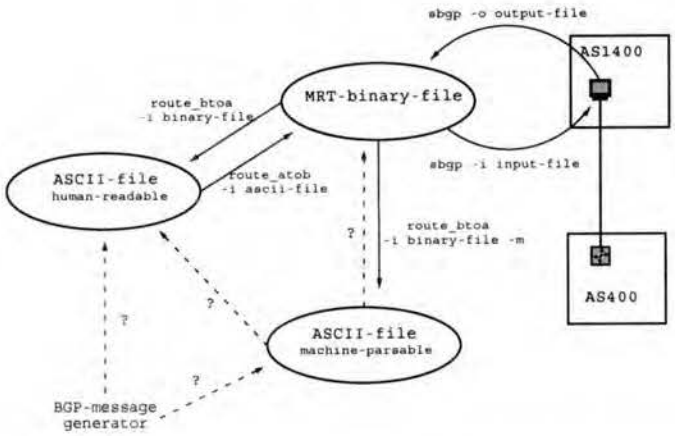

Figure 5-3 : Possibilités de transformation de descriptions textuelles de messages BGP

L'outil de génération de messages BGP comporte deux fonctionnalités distinctes. La première (**gen_pref**) produit des descriptions ASCII « machine-parseable » de messages BGP. Ce format permet de décrire de façon synthétique les attributs de base des UPDATE BGP et ainsi de manipuler facilement de grandes quantités de données. La seconde (**init_table**) transforme ces descriptions au format « human-readable », qui est utilisé par **route_atob**. En ayant séparé ces deux fonctionnalités, nous avons la possibilité d'exploiter le résultat éventuel d'autres outils, tels que RTG [Maennel], qui dans ses premières versions générait des tables de routage dans le format « machine-parseable » à des fins d'analyse.

5.2.3. Production d'une description texte des messages

Les UPDATE BGP sont générés sous le format MRT « machine-parseable ». Chaque UPDATE s'affiche sur une ligne et comprend 14 champs délimités par le caractère « | ». Ils répondent à un certain nombre d'exigences qui doivent nous permettre de mesurer leur temps de traitement par le routeur testé.

TOUT MESSAGE QUI ENTRE DANS LE DUT DOIT EN RESSORTIR

Cette condition est nécessaire pour pouvoir mesurer le temps de traitement des messages au moyen de tests black-box, pour lesquels le dispositif de mesure est situé en amont et en aval du routeur.

Les préfixes appartiendront au pool des adresses privées de classe A (10.0.0.0 à 10.255.255.255). Ces adresses ne sont pas diffusées dans l'Internet : elles ne risquent donc pas de se trouver par hasard dans la table de routage du DUT, et ainsi de perturber les mesures. Par ailleurs, les réseaux de classe A comportent un grand

nombre d'hôtes potentiels (plus de 16 millions), et peuvent être utilisés pour constituer un nombre important de sous-réseaux.

TRAITEMENT IDENTIQUE DES ANNONCES

Cette condition nous permet de faire l'hypothèse que dans chaque série de tests, chaque message BGP pourra être considéré comme une répétition du même événement.

Les préfixes annoncés auront tous une même longueur de masque. Nous utiliserons un masque de 24 bits, étant donné que c'est la longueur la plus représentée dans les tables de routage de l'Internet. [Potaroo] De plus, les UPDATE qui contiennent ces préfixes partageront certaines caractéristiques : même longueur d'AS-PATH, mêmes attributs (NEXT-HOP, ORIGIN, MED, LOCAL-PREF...).

TRAITEMENT INDEPENDANT DES ANNONCES

Les préfixes annoncés seront tous différents : une annonce répétée du même préfixe pendant le déroulement du test pourrait être considérée comme une oscillation de route et déclencher un mécanisme qui sert à empêcher la propagation des routes instables, le route-flap dampening [Ste01], ce que nous voulons éviter. En pratique, il y a moyen de générer potentiellement 65536 sous-réseaux de longueur 24 à partir du réseau 10/8.

Les messages BGP seront mélangés, afin que la convergence ne soit pas influencée par l'ordre des routes.

SIMPLICITE DES CALCULS

Chaque message BGP produit ne contiendra qu'un seul préfixe : nous pourrons ainsi mesurer le temps de traitement du message sans devoir tenir compte d'éventuels regroupements de préfixes. Pour ce faire, il suffit que le contenu de l'AS-PATH soit différent pour chaque préfixe.

De plus, un AS marqueur sera ajouté dans l'AS-PATH des préfixes testés afin vérifier si le message qui sort du routeur est bien celui qui y est entré, en particulier dans le cas où le DUT connaîtrait plusieurs chemins pour le même préfixe

Ces critères nous permettent de générer des UPDATE BGP pour lesquels on peut mesurer le temps de traitement par un routeur. Ces UPDATE sont suffisamment réalistes pour pouvoir être acceptés et traités par le routeur ; ils ne sont toutefois pas le reflet exact de la réalité dont ils diffèrent, par exemple par l'absence systématique de regroupement des préfixes dans une seule annonce BGP, ou par le fait que des préfixes proches peuvent avoir des AS d'origine différents.

5.3. Envoi de messages BGP

Parmi les outils de la suite MRT [Merit], deux avaient des prédispositions pour envoyer des séquences de préfixes à un ou plusieurs voisins BGP. `bgpsim` est capable d'envoyer une suite de préfixes compris dans une fourchette fixée à une fréquence élevée, puis de les retirer. Il peut effectuer cette opération vis-à-vis de plusieurs voisins.

`bgpsim` présente trois inconvénients. Premièrement, les préfixes sont annoncés en séquences, et donc triés. Il n'est pas possible de les envoyer dans un ordre aléatoire. Ensuite, les simulations concernent des rangs de préfixes, par exemple tous les préfixes entre 10.0.1.0/24 et 10.10.255.0/24. Les attributs de chemins sont identiques pour cette liste de préfixe : s'il est possible de prévoir des changements au cours du temps des attributs appliqués à une séquence d'annonce, par contre, il n'est pas possible de les appliquer individuellement à chaque préfixe. Enfin, s'il est possible de fixer l'intervalle qui sépare l'annonce de l'ensemble des préfixes, puis son retrait et une nouvelle annonce de la séquence, il n'est pas possible de fixer l'intervalle entre les différents préfixes d'une même séquence. De ce fait, les différents préfixes sont groupés dans un seul message UPDATE, ce que nous voulions éviter.

bgpsim présente des caractéristiques intéressantes, surtout pour étudier les oscillations de routes, mais ne nous permet pas de répondre à nos objectifs, et en particulier de s'adapter à la méthodologie proposée par Berkowitz [BHR+02].

sbgp, de son côté, a été conçu pour capturer des séquences d'événements et les rejouer. Il est capable de tenir compte de l'intervalle de temps qui sépare des annonces, et de traiter les annonces individuellement. Il suffit de pouvoir générer une séquence d'annonce avec les caractéristiques que l'on souhaite et de l'envoyer avec **sbgp** pour pouvoir jouer un scénario. Reste à savoir comment gérer les diverses sessions avec **sbgp**.

5.3.1. Gestion des sessions multiples avec sbgp

sbgp a la capacité de capturer les messages UPDATE venant de plusieurs sessions BGP. De même, il peut rejouer des suites d'événements avec plusieurs pairs : il suffit qu'à chaque NEXT-HOP utilisé dans le fichier d'UPDATE corresponde une des interfaces, réelle ou virtuelle, de la machine où **sbgp** tourne.

Chaque UPDATE du fichier contient des renseignements qui sont propres à BGP (préfixe, annonce ou retrait, AS-PATH, NEXT-HOP), mais également des renseignements propres à la couche IP (adresse IP de l'émetteur).

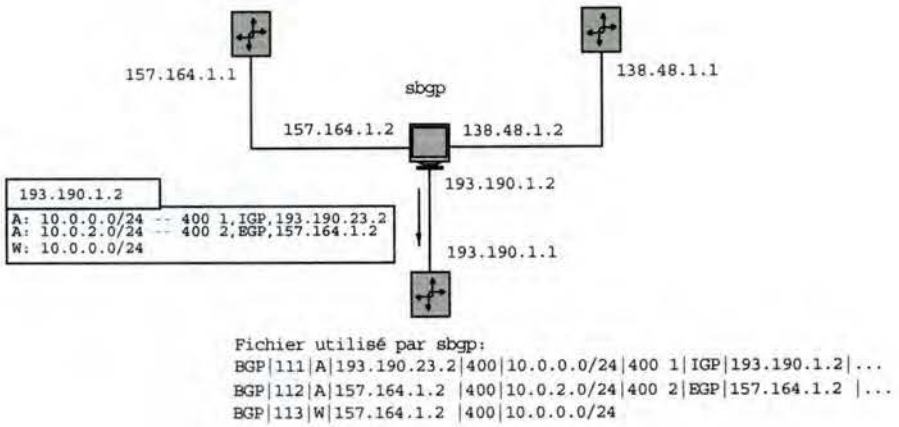

Figure 5-4 : Gestion des sessions multiples avec sbgp

La figure 5-4 illustre le fonctionnement d'une session multiple avec **sbgp**. En pratique, **sbgp** envoie tous les UPDATE du fichier à chacun de ses voisins. Dans l'annonce qui est envoyée, l'adresse IP réelle de l'annonceur (**sbgp**) se substitue à l'adresse IP qui se trouve dans le fichier d'UPDATE. Par contre, la valeur du NEXT-HOP reste inchangée.

Tout fonctionne sans problème pour les annonces : chaque session BGP est formée entre deux voisins qui se trouvent sur un même réseau, et par conséquent, le NEXT-HOP qui correspond à ce réseau est accessible directement. Les voisins acceptent ou rejettent l'UPDATE sur base de l'accessibilité du NEXT-HOP.

Par contre, les retraits de route posent problème dans ce cas : ils sont annoncés à tous les voisins, en substituant à l'adresse IP du fichier d'UPDATE l'adresse IP de l'interface sur laquelle la route est annoncée, et le voisin ne peut qu'accepter ce retrait. Supposons que le fichier d'UPDATE ait prévu de retirer un préfixe d'une seule des sessions BGP. Ce retrait est envoyé à tous les voisins. Si le préfixe avait été annoncé précédemment à chacun des voisins de **sbgp**, il sera retiré correctement dans une session, et indûment dans toutes les autres.

C'est une des imperfections de **sbgp**. Pour y remédier, il suffit soit de ne pas envoyer de messages de retrait de routes, soit d'utiliser un fichier d'UPDATE et un processus **sbgp** pour chacune des sessions BGP que l'on veut établir.

5.3.2. Gestion de l'intervalle de temps entre les UPDATE

L'intervalle de temps qui sépare deux annonces successives n'est pas trop compliqué à gérer dans le cas où un seul fichier d'UPDATE est utilisé pour jouer une session d'événements avec un seul voisin : dans ce cas, l'intervalle de temps qui sépare deux annonces est calculé sur base du timestamp des messages.

Là où cela se complique, c'est lorsque l'on souhaite envoyer des UPDATE synchronisés à plusieurs voisins : la première annonce vers chaque voisin est toujours envoyée à un moment qui dépend du temps d'établissement de la session BGP et que l'on ne peut pas prévoir avec précision. S'il est donc possible de définir une séquence d'événements avec un seul voisin, il est difficile de synchroniser des événements entre plusieurs sessions BGP, que l'on utilise un ou plusieurs processus `sbgp` pour chacune des sessions BGP.

5.4. Mesure du temps de traitement

5.4.1. Mesure du temps de passage des messages

Les mesures sont effectuées au moyen d'un sniffer, qui capture tout le trafic qui s'échange sur le réseau. L'outil utilisé est `tethereal` [Eth, Teth]. Un filtre est appliqué pour conserver exclusivement les messages BGP.

Les messages qui sont envoyés au DUT et ceux qui en sortent sont capturés sur le même PC : cela permet d'éviter les problèmes de synchronisation. Pour la même raison, on s'arrange pour que tous les messages passent par le même réseau physique et soient capturés sur le même hub.

5.4.2. Extraction des informations

Les mesures de convergence étudient le temps de traitement des messages UPDATE, mais les autres types de messages (OPEN, NOTIFICATION, KEEPALIVE) apportent des informations qui ont leur intérêt, par exemple pour détecter des problèmes ou anomalies de la session, ou pour vérifier que toutes les sessions ont bien été établies. Pour estimer le temps mis par un UPDATE BGP pour traverser un routeur au moyen des tests black-box, il faut pouvoir reconnaître cet UPDATE avant et après son passage dans le routeur.

Les fichiers capturés par `tethereal` ne contiennent que des messages BGP, mais ils comportent le détail des différents protocoles qui encapsulent ces messages (TCP, IP, Ethernet). Pour chaque trame Ethernet du fichier, les informations sont organisées en sections qui correspondent aux différentes couches de la pile de protocoles IP.
Nous avons donc découpé le fichier en trames et nous avons été rechercher dans les différentes sections de chaque trame les informations les plus pertinentes pour la suite de ce travail. De la section BGP, nous avons retenu le type de message. Pour les UPDATE, nous avons retenu le préfixe et l'information indiquant s'il s'agit d'une annonce ou d'un retrait de route. Pour les annonces, nous avons retenu en plus les attributs obligatoires (NEXT-HOP, AS-PATH, ORIGIN) et l'attribut COMMUNITY : ils sont utilisés pour distinguer les annonces lorsque le même préfixe est annoncé au DUT par plusieurs de ses pairs. De la section IP, nous avons retenu les adresses IP source et destination de chaque message, pour distinguer les messages entrants des messages sortants. De la section Ethernet, nous avons retenu le numéro de la trame, relatif au début de la capture par le processus `tethereal`. Enfin, nous avons retenu le moment de la capture.

L'extraction est réalisée au moyen du script **grep_capture** ; le résultat est présenté en annexe G.

5.4.3. Calcul de la convergence

La durée de la convergence est calculée pour chaque préfixe individuellement. Dans les cas où les UPDATE BGP annoncent un seul préfixe, le temps de traitement d'un UPDATE est équivalent au temps de traitement d'un préfixe.

L'utilisation de `tethereal` combinée à l'extraction des informations pertinentes par **grep_capture** nous fournit un fichier qui établit le temps de passage de chaque préfixe soit en amont, soit en aval du routeur testé.

La première action du script **calcule** consiste donc à déterminer le sens de propagation de l'annonce : pour ce faire, nous utilisons l'adresse IP de destination de l'UPDATE. Ce choix nous permet de distinguer plusieurs flux en sortie, dans le cas où le DUT possède plusieurs voisins en aval, mais ne permet pas de distinguer les flux en entrée. Nous basons nos calculs sur l'hypothèse qu'il existe une séparation temporelle entre les annonces faites par chacun des pairs en amont du DUT. Si ce n'était pas le cas, il faudrait trouver un autre critère pour reconnaître les différentes instances du même préfixe et établir leur temps de passage, par exemple utiliser un marqueur dans l'AS-PATH.

Nos calculs de convergence sont basés sur l'hypothèse que chaque préfixe reçu pas le DUT est traité puis annoncé à ses pairs en aval. Il est donc important de configurer le DUT pour qu'il ne rejette pas les préfixes testés (choix des filtres, annonce des préfixes spécifiques en cas d'agrégation). Il est aussi important de générer des messages qui seront acceptés (AS-PATH sans boucle) et sélectionnés par le DUT (leurs attributs de chemins sont les meilleurs).

5.5. Contrôle du déroulement de l'expérience

5.5.1. Dispositif de test

Les mesures sont effectuées à l'extérieur du routeur. Si une route est annoncée à un pair en aval, c'est qu'elle a été choisie par le routeur testé. Pour faire fonctionner le processus de décision, il faut que le routeur ait à choisir entre deux chemins pour le même préfixe : sa table de routage doit être initialisée. Quel que soit le test effectué, le préfixe testé est toujours préféré : il faut que la comparaison entre le nouveau préfixe et celui qui se trouve dans la table BGP du routeur donne la préférence au nouveau préfixe. Nous contrôlons le processus de décision au moyen des attributs de chemin des préfixes.

Le dispositif de test retenu comprend 4 routeurs et 2 PC pour les tests de base. Il est représenté sur la figure 5-5.

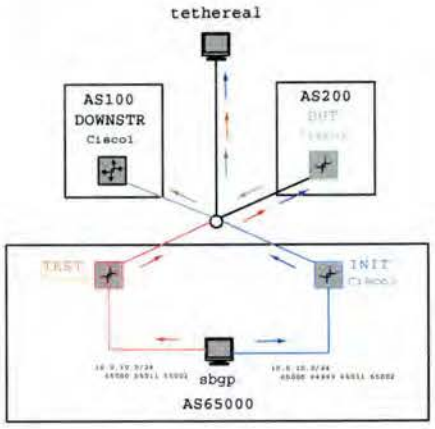

Figure 5-5 : Dispositif de test - Schéma basé sur la topologie du laboratoire

Le routeur pour lequel on mesure la convergence est appelé DUT (device under test).

Deux routeurs Cisco sont utilisés en amont du DUT. Le premier, appelé INIT (Cisco3), sert à initialiser la table de routage du DUT. Le second, appelé TEST (Cisco4), sert à envoyer les messages dont on mesure la convergence.

Un routeur Cisco (DOWNSTREAM) est utilisé en aval du DUT. Il a pour objectif d'ouvrir une session BGP qui permet aux préfixes acceptés d'être ré-annoncés. On peut donc mesurer leur convergence, sans qu'il soit nécessaire d'interférer avec le DUT.

Les routeurs situés en amont du DUT ne font qu'envoyer des messages UPDATE, ils n'en reçoivent jamais. Le routeur situé en aval du DUT ne fait que recevoir des messages UPDATE du DUT, il ne lui en transmet jamais. Les messages OPEN, KEEPALIVE et NOTIFICATION sont envoyés dans les deux sens pour toutes les sessions.

Un PC sert à alimenter les deux routeurs en amont du DUT, au moyen de deux processus **sbgp** (un par session). Les messages qui servent à initialiser le DUT sont envoyés en premier lieu. On attend un temps suffisant afin de s'assurer que la table de routage du DUT est stable avant de commencer les tests proprement dits.

Un PC sert à effectuer les mesures : nous utilisons l'analyseur de trafic **tethereal** , qui capture tout le trafic qui s'échange sur le réseau et nous lui appliquons un filtre pour conserver exclusivement les messages BGP. Pour éviter les problèmes de synchronisation, les messages BGP qui entrent dans le DUT et ceux qui en sortent passent par le même réseau physique et sont capturés sur le même hub au moyen d'un seul processus **tethereal**. Malgré le fait que tout le trafic passe par le même endroit, les messages sont suffisamment espacés pour éviter que des collisions potentielles n'interfèrent fortement avec les mesures.

Ce dispositif est proche de celui proposé par Berkowitz [BHR+02], excepté que pour lui, les tests de base ne mettent en jeu qu'un seul routeur en amont du DUT. Dans ces conditions, un préfixe est sélectionné et annoncé aux pairs en aval parce que le DUT ne connaît pas d'autre route pour la même destination.

5.5.2. Configuration des routeurs et des PC

Etant donné que nous n'utilisons pas le lien physique, mais bien un lien virtuel, pour établir la session BGP entre les routeurs en amont du DUT et le PC sur lequel tourne **sbgp**, il est nécessaire de configurer deux interfaces virtuelles sur le PC qui sert à alimenter les routeurs (**ifconfig**)

Les configurations de base des routeurs se trouvent en annexe I. Elles visent à établir une session BGP entre les différents intervenants, et à imposer des conditions qui nous permettent de contrôler le déroulement de l'expérience.

Le timer MinRouteAdverTimer, qui permet de fixer un laps de temps minimum entre deux mises-à-jour pour un même préfixe, est fixé à 0 sec, afin que les UPDATE ne soient pas mis en attente et groupés dans une même trame après avoir été sélectionnés.

Afin d'éviter d'éventuelles interférences des messages KEEPALIVE avec les UPDATE, les valeurs de Keepalive et Holdtime de tous les routeurs sont fixées à 3600 et 10800 respectivement.

Un filtre est appliqué empêcher que les messages reçus d'un routeur en amont du DUT soient retransmis vers l'autre routeur en amont du DUT. Cela permet d'assurer que le DUT n'envoie jamais de messages UPDATE aux voisins INIT et TEST.

Pour permettre aux UPDATE TEST d'être sélectionnés même en cas d'égalité des autres critères, nous utilisons une adresse loopback (identifiant du routeur) plus basse sur Cisco4 que sur Cisco3.

Tous les messages envoyés au DUT par le voisin TEST portent une valeur pour l'attribut COMMUNITY. Cette caractéristique n'est indispensable que dans le cas des tests réalisant le filtrage de route sur base de l'attribut COMMUNITY. Cependant, en l'imposant dès le départ, nous avons pu réaliser l'ensemble de nos tests de manière similaire.

En règle générale, les configurations de Cisco1 (DOWNSTREAM), Cisco3 (INIT) et Cisco4 (TEST) restent inchangées pour l'ensemble des tests. Seule la configuration de Cisco2 change en fonction des paramètres étudiés. Nous gardons néanmoins inchangés l'adresse IP des interfaces loopback, l'adresse IP des différents voisins BGP, le numéro d'AS des différents voisins BGP, et les routes statiques.

5.5.3. Conduite des tests

L'idée des tests est de mesurer les temps de convergence d'annonces indépendantes. Pour avoir la certitude que les mesures effectuées ne sont pas influencées par des événements extérieur, nous répétons la mesure un certain nombre de fois, 1000 en l'occurrence. En pratique, 1000 préfixes différents sont envoyés au DUT, chaque préfixe

étant annoncé avec un intervalle d'une seconde par rapport au précédent. Ces préfixes sont stockés dans un fichier. L'expérience complète comprend 3 répétitions des tests, afin de vérifier qu'ils sont bien reproductibles.

Une série de tests comporte des mesures de convergence effectuées avec plusieurs configurations du DUT. Chaque série comprend un test de référence, qui nous permet de comparer les résultats obtenus entre les diverses expériences.

Les mêmes opérations sont répétées pour chaque série de tests. L'envoi des messages est géré par un PC, les mesures proprement dites sont gérées par un autre. La séquence des événements est décrite ci-dessous.

ENVOI DES MESSAGES

T0[4] : réinitialisation du DUT (commande **reload**).
T3 : chargement des fichiers de configuration sur le routeur avec **cisconf**. Ce délai est nécessaire pour permettre aux routes d'être réinstallées après la réinitialisation et aux routeurs d'être à nouveau accessibles.
T4 : vérification des configurations de tous les routeurs (commande **show running-config**)
T5 : lancement du processus **sbgp** qui servira à initialiser le DUT.
T10 : lancement du processus **sbgp** qui servira à envoyer au DUT les annonces testées (les routes initiales ont convergé).
T29 : vérification, pour chaque routeur, de l'état de tous les voisins (commande **show ip bgp neighbor**).
T29 : vérification que Cisco1 a bien reçu un préfixe particulier (commande **show ip bgp <prefix>**)
T30 : arrêt des processus **sbgp** et réinitialisation des routeurs (=T0 de la série suivante)

L'utilisation des commandes **show** (T4, T24) permet de faciliter le diagnostic et l'interprétation des résultats : en photographiant l'état des routeurs à des moments particuliers, elle permet de vérifier que le test s'est déroulé correctement.

MESURES

Les opérations à effectuer pour les mesures sont beaucoup plus simples que celles pour l'envoi de messages, puisque nous capturons l'ensemble des messages échangés, messages d'initialisation et messages de test. Ces opérations sont synchronisées avec celles qui permettent d'envoyer les messages.

T4 : lancement du processus **tethereal** qui permet de capturer les messages échangés.
T29 : arrêt du processus **tethereal**. Le processus est tué 1 min avant la fin de la session BGP entre **sbgp** et les routeurs en amont du DUT : cela permet d'éviter de capturer tous les messages de retrait de routes qui suivent la fin de la session.

Dans la mesure du possible, le fichier de préfixes « test » sera réutilisé d'une expérience à l'autre, afin de pouvoir comparer les résultats des diverses expériences.

[4] Temps 0, exprimé en minutes.

5.6. Conclusion

Dans cette première partie, nous avons mis au point une méthodologie qui permet de caractériser le fonctionnement du protocole BGP au niveau d'un routeur isolé, qui reçoit des input sur une ou plusieurs de ses interfaces, et envoie des output sur une ou plusieurs de ses interfaces. Cette méthodologie, inspirée de [BHR+02], a pour objectif de calculer le temps de convergence des UPDATE BGP au moyen de tests *black-box*, c'est-à-dire sur base de leur temps de passage à un point de contrôle situé avant l'entrée dans le routeur étudié et après sa sortie. Les mesures ne concernent que le plan de contrôle (routing) proprement dit, et ne font pas intervenir d'autres processus internes du routeur, comme la transmission des paquets de données (*forwarding*). Le temps de traitement calculé inclut toutefois divers excédents, tels que le temps de propagation du message sur le media de transmission ou à l'intérieur du routeur, et le temps de redistribution des préfixes. Nous avons minimisé le temps de redistribution des préfixes (MRAI), afin de pouvoir étudier plus finement le fonctionnement interne de BGP, et nous avons supposé que les autres excédents étaient constants.

Nous avons d'abord défini les conditions générales qui permettent à tout UPDATE entrant dans le routeur d'en ressortir, que ce soit la configuration du routeur, ou les caractéristiques des UPDATE. Sur base de ces conditions, nous avons mis au point un système qui permet d'envoyer de manière contrôlée des UPDATE vers le routeur que l'on étudie. Ce système combine l'utilisation d'un script de génération d'UPDATE BGP et de l'outil `sbgp` [Merit].

Nous avons ensuite établi un système de mesure du temps de traitement des UPDATE BGP par un routeur. Pour ce faire, nous estimons le temps de passage des UPDATE BGP avant l'entrée et après la sortie du routeur étudié au moyen de l'analyseur de trafic `tethereal`. Pour éviter les problèmes de synchronisation, les temps de passage sont mesurés au même endroit, et au moyen d'un seul processus. La première action du script de calcul des temps de traitement sera donc de déterminer à quelle session il faut rattacher chacune des mesures effectuées. Les calculs ont été simplifié par le fait que chaque UPDATE annonçait un et un seul préfixe, qu'on empêchait les regroupements de préfixes en s'assurant de l'unicité des attributs de chemin, et que l'intervalle qui sépare deux UPDATE successifs est suffisant pour considérer chaque UPDATE comme un événement indépendant.

Nous avons finalement établi un scénario qui permet d'étudier le protocole BGP : choix des différents pairs impliqués et définition de leur rôle respectif, planification des événements.

Si les outils développés et la méthodologie déployée nous permettent effectivement de répondre à nos objectifs de départ, à savoir étudier le fonctionnement interne du processus de convergence BGP au niveau d'un routeur isolé, on peut toutefois regretter une complexité excessive et inutile du dispositif mis en place, ainsi que l'absence d'automatisation de certaines opérations. En résumé, nous aurions pu obtenir des résultats identiques avec un dispositif plus simple. Mais l'objectif principal était de conserver des conditions de travail similaires pour l'ensemble des tests. Ces objections pourraient toutefois être levées en établissant un nouveau scénario qui tient compte des observations recueillies au cours d'une première évaluation du fonctionnement interne d'un routeur BGP.

6. Présentation des résultats

Un examen rapide de la littérature scientifique nous montre que, jusqu'à présent, les mesures de convergence qui se rapportent à BGP ont été réalisées soit sur des réseaux entiers ou des parties de réseau [LAB-TR-00] [LAB+00] [LAW+01], soit sur des simulateurs [NC02], mais jamais sur des routeurs isolés. Ce mémoire constitue donc une première, puisque les mesures de convergence de BGP seront effectuées sur un routeur isolé et qu'elles utiliseront le principe des tests black-box.

L'objectif de ce paragraphe est de définir une méthode qui permet de présenter les résultats de manière comparable et synthétique : l'interprétation des résultats consiste à comparer les temps de traitement obtenus lorsqu'on modifie un des paramètres de configuration du DUT par rapport à un test de référence.

Un premier examen de la distribution des temps de traitement des UPDATE obtenus au cours du déroulement de l'expérience nous montre qu'il existe une certaine variabilité entre les mesures, mais que généralement, les résultats ne sont pas aléatoires. Nous nous basons sur le fait que

chaque test représente 1000 répétition d'un événement élémentaire et que chaque test est répété trois fois pour faire la distinction entre les temps de traitement caractéristiques du paramètre étudié, les temps de traitement qui apparaissent de manière aléatoire, et éventuellement, les temps de traitement qui apparaissent indépendamment du paramètre étudié.

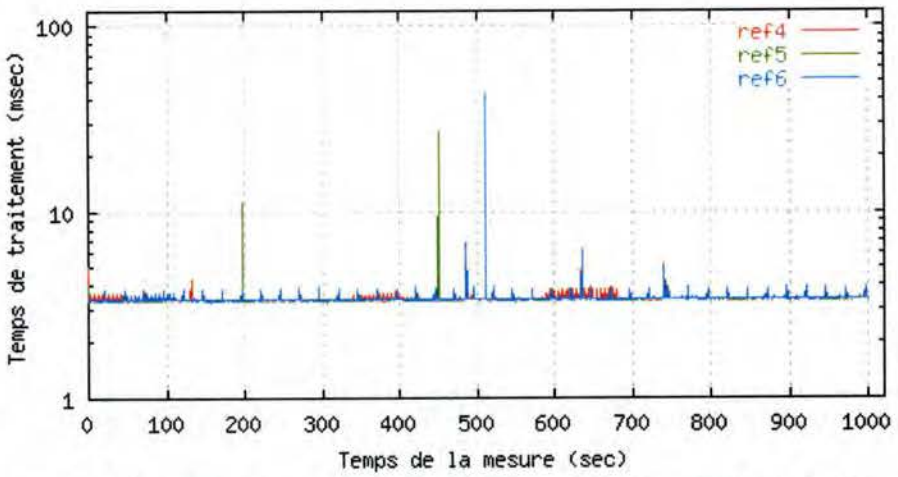

Figure 6-1 : Temps de traitement pour une décision basée sur la longueur de l'AS-PATH (échelle logarithmique)

La figure 6-1 nous montre, à l'échelle logarithmique, la distribution des temps de traitement pour trois répétitions (ref4, ref5 et ref6) d'un même test, qui consiste en la redistribution par le DUT de préfixes choisis sur base de la longueur de l'AS-PATH. Nous constatons que certaines valeurs particulièrement élevées (jusque 45 msec) apparaissent de manière non reproductible entre les trois séries de mesures. Expérimentalement, il ne nous a pas été possible de rattacher ces valeurs élevées à des événements particuliers ; toutefois, on peut supposer qu'IOS[5] est occupé à autre chose qu'à traiter des messages BGP (transmission de paquets, réponse à des requêtes d'une autre machine, ...). Ces événements potentiels sont malgré tout assez rares dans le cadre du test, et sont donc considérés comme non représentatifs du paramètre étudié.

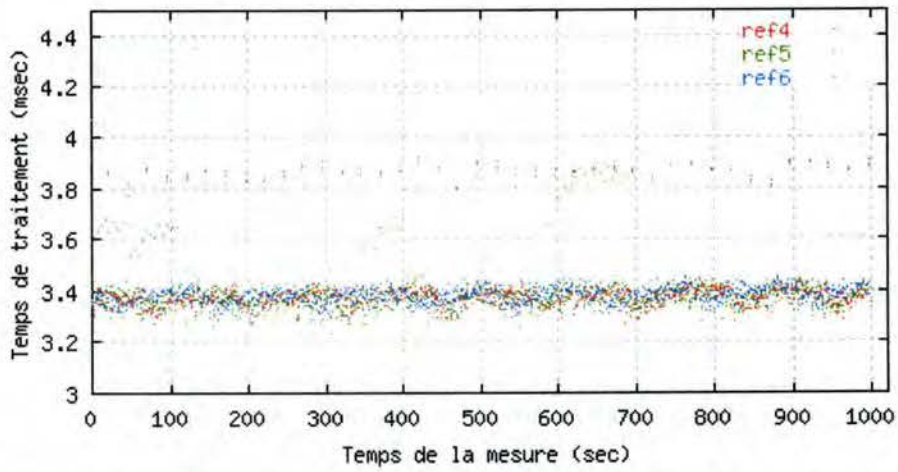

Figure 6-2 : Temps de traitement mesurés pour une décision basée sur la longueur de l'AS-PATH (échelle décimale)

La figure 6-2 nous montre que, à l'échelle décimale, les temps de traitement des messages BGP se superposent pour les trois répétitions d'un même test, et qu'il est assez difficile de distinguer les courbes ainsi obtenues. Les temps de traitement sont situés entre 3 et 4.5 msec et sont distribués sur deux niveaux de valeurs. Le niveau sur lequel sont situées la majorité des valeurs (plus de 90% des mesures) représente bien le test que l'on fait : les n répétitions du même test

[5] IOS : Internetworking Operating System. Système d'exploitation des routeurs Cisco.

montrent des profils très reproductibles. Les valeurs qui semblent apparaître de manière accidentelle ne sont pas prises en compte.

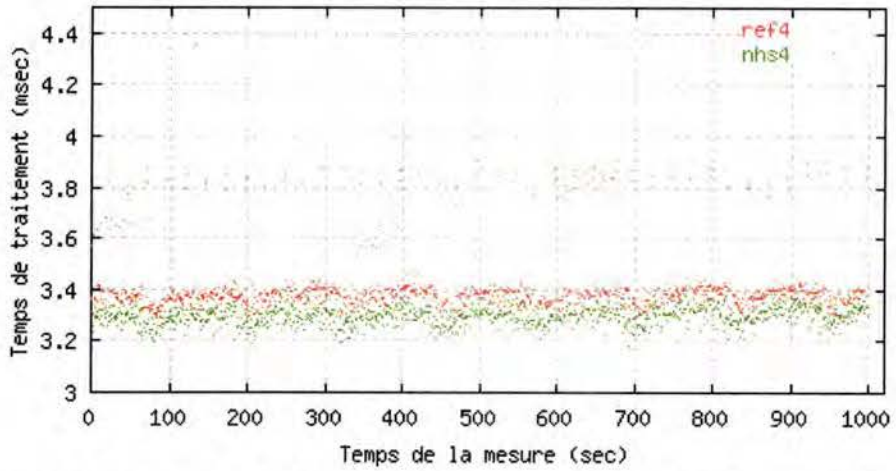

Figure 6-3 : Comparaison des temps de traitement obtenus avec deux configurations différentes du DUT pour une décision basée sur l'AS-PATH (référence et utilisation de l'option NEXT-HOP-SELF)

La figure 6-3 nous montre les temps de traitement obtenus avec deux configurations différentes du DUT (configuration de référence, ref4, ou utilisation de l'option NEXT-HOP-SELF, nhs4). Dans ces conditions, on obtient des courbes distinctes, même si une partie des valeurs se chevauchent. L'analyse de la distribution des temps de traitement nous permet de mettre en évidence des différences de 0.1 msec entre les mesures, soit environ 3% de la valeur de référence.

La vue qualitative offerte par les graphiques de distribution des temps de traitement ne nous permet pas de comparer de manière synthétique les caractéristiques de convergence obtenues avec diverses configurations, ni de quantifier les effets observés. Pour ce faire, il est plus facile de travailler sur des moyennes ; or, nous avons vu que des temps de traitement particulièrement élevés pouvaient apparaître, sans que l'on puisse contrôler ni leur nombre, ni leur valeur, et que ces valeurs ne semblent pas représenter une caractéristique du test réalisé.

Pour éviter de prendre en compte de telles valeurs, nous avons trié les 1000 mesures de convergence obtenues dans chaque test par ordre croissant, ce qui nous permet de rassembler les valeurs anormalement faibles ou anormalement élevées au début et à la fin de la série de valeurs. Nous avons ensuite calculé la moyenne de la 25e à la 925e mesure pour chaque test, afin permet d'éliminer les valeurs de convergence qui ne sont pas représentatives du test réalisé ; cette façon de procéder nous permet en outre de traiter de la même manière les résultats de tous les tests (les calculs sont réalisés sur 900 mesures). Nous avons également utilisé, pour être complet, d'autres indicateurs, tels que le minimum, maximum et médiane pour la plage concernée, ainsi que l'écart-type.

Cette manière de procéder nous permet d'obtenir des valeurs très reproductibles entre les différentes répétitions d'un même test. L'écart-type obtenu dans ces conditions se situe aux alentours de 0.03 msec (soit environ 1% de la moyenne des valeurs) ; la médiane et la moyenne sont à peu près égales, ce qui indique que la moyenne n'est plus influencée par des valeurs extrêmes. Dans la majorité des cas, la moyenne seule permet de caractériser les tests. Toutefois, cette méthode ne permet pas d'interpréter tous les résultats : dans certains cas, l'analyse de l'évolution des temps de traitement au cours du déroulement de l'expérience apporte un complément d'informations.

Résultats et discussion

7. Mesures de convergence avec un seul pair en aval du DUT

Le dispositif mis en place a pour objectif d'essayer de décortiquer le fonctionnement interne d'un routeur sans en connaître l'implémentation, et d'évaluer l'impact de différents facteurs sur le temps de convergence des UPDATE BGP. Au niveau d'un routeur, le terme de convergence inclut l'application de politiques locales aux routes apprises de pairs, la sélection de la meilleure route par le processus de décision, l'application de politiques locales à la sortie du routeur dans le but d'influencer la décision de routage des pairs en aval.

Les tests black-box donnent une estimation des temps de convergence légèrement surévaluée : les mesures incluent le temps de convergence proprement dit, le temps de propagation des UPDATE BGP sur le média de transmission et à l'intérieur du routeur et le temps de redistribution des routes. Pour cette raison, nous préférerons le terme « temps de traitement » pour décrire le résultat de nos mesures.

On fait l'hypothèse que, pour des tests réalisés dans les mêmes conditions matérielles, la somme des excédents est constante. Par conséquent, si un paramètre a été modifié entre deux tests et qu'on obtient des temps de traitement différents entre ces deux tests, on supposera qu'il y a un lien de cause à effet entre les deux événements. Le chapitre consacré à la présentation des résultats nous a donné un aperçu du fait que plusieurs répétitions de la même expérience donnaient des mesures très semblables, alors que la modification d'un des paramètres expérimentaux affectait la mesure.

La plupart de nos tests sont basés sur le même schéma :

- Le critère de sélection est la longueur de l'AS-PATH
- La valeur du MRAI est fixée à 0 sec (pour minimiser le temps de redistribution des annonces).
- La valeur de Holdtime est fixée à 10800 (valeur supérieure à la durée des tests)

Nous mentionnerons uniquement les cas où ces conditions ne sont pas remplies.

7.1. Le processus de sélection de routes

Le rôle d'un routeur BGP consiste à choisir la meilleure route parmi toutes les routes qu'il connaît vers une destination. Nous avons vu dans la section 3.3.3 que le fonctionnement du processus de décision pouvait être subdivisé en trois phases : la première consiste à calculer un degré de préférence pour une route après lui avoir appliqué les politiques d'acceptation. La deuxième consiste à sélectionner la meilleure route pour un préfixe et à l'installer dans la table de routage. La troisième sert à propager les routes vers les pairs voisins, à pratiquer l'agrégation et la réduction d'information et à appliquer des politiques diverses.

Nous allons, dans cette première partie du travail, nous intéresser à la deuxième phase, c'est-à-dire à la sélection de routes proprement-dite. Pour ce faire, nous avons appliqué aux UPDATE une politique minimaliste ACCEPT-ALL, REDISTRIBUTE-ALL (accepte tous les préfixes, à condition qu'il n'y ait pas de boucles ; redistribue tous les préfixes), qui consiste à n'utiliser aucun filtre à l'entrée et à la sortie du routeur. Nous avons également configuré le MRAI à 0 sec, afin de minimiser les temps de redistribution.

7.1.1. Redistribution d'annonces à un voisin EBGP

L'objectif de cette série de tests est d'essayer de comprendre le fonctionnement du processus de sélection de routes, et de vérifier si les temps de traitement des préfixes sont indépendants du critère utilisé dans le processus de décision.

Nous avons donc sélectionné une série de critères de décision pour lesquels on peut mesurer le temps de convergence au moyen de tests black box. Les tests black box nécessitent qu'un événement externe puisse être mesuré en amont du routeur (signal déclencheur) et en aval (résultat du processus de décision). Par ailleurs, nous avons limité notre étude aux seuls cas où le pair en amont du routeur pour lequel on étudie la convergence est un pair externe (EBGP).

En raison de ces exigences, la préférence locale, qui n'est pas échangée entre voisins EBGP, l'origine locale, pour laquelle on ne peut pas trouver un événement déclencheur externe ou l'ancienneté de l'annonce, qui ne produit pas une nouvelle annonce ne seront pas examinés dans cette série de tests.

Nous retiendrons les critères suivants :

WEIGHT : L'attribut weight est spécifique à Cisco. Il est local au routeur et n'est pas contenu dans les messages échangés entre voisins BGP. C'est le premier des critères de décision examinés. La préférence est donnée aux préfixes avec le weight le plus élevé.

AS-PATH : Si le weight et la préférence locale pour une route sont les mêmes et qu'aucune route n'a une origine locale au routeur, la route comprenant le plus petit nombre de systèmes autonomes dans son chemin est choisie (AS-PATH le plus court).

ORIGIN : A égalité des critères précédents, la route avec le code d'origine le plus bas est choisie (IGP<EGP<INCOMPLETE).

MED : Lorsque l'examen des critères de sélection précédents dans le processus de décision n'a pas permis de choisir une seule route vers un préfixe, le MED (Multi-Exit-Discriminator) est pris en considération. L'étude portera uniquement sur le cas où les deux voisins entre lesquels il faut choisir la meilleure route appartiennent au même système autonome. La route avec le MED le plus bas est choisie.

L'ordre d'arrivée des messages peut influencer le choix du meilleur chemin. La commande `bgp deterministic-med` (**MED-DET**) permet de comparer les annonces de manière plus précise et reproductible en groupant les annonces reçues par système autonome, en appliquant l'algorithme du processus de décision au sein de chaque groupe et en comparant les vainqueurs de chaque groupe entre eux. Cette commande doit être utilisée pour tous les routeurs au sein d'un même système autonome ou pour aucun. [Cisco]

RID : Lorsque les critères de sélection basés sur les attributs BGP n'ont pas permis de sélectionner une route, et qu'il n'est pas possible de discriminer les routes sur base de l'ancienneté de l'annonce, en dernier recours, la route apprise du voisin avec l'identifiant du routeur (router ID[6]) le plus bas est choisie.

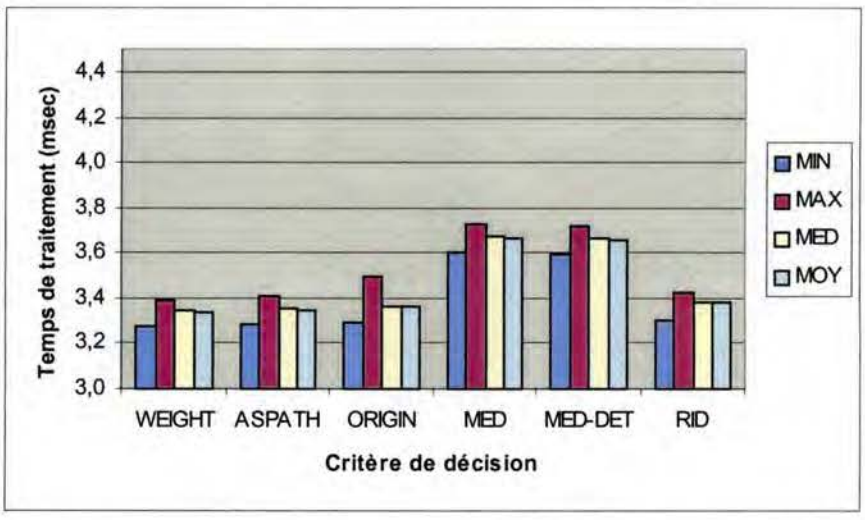

Figure 7-1 : Temps de traitement obtenus pour différents critères du processus de décision

[6] Le router ID est l'adresse IP la plus élevée, avec une préférence pour les adresses loopback (virtuelles). Il peut être configuré manuellement.

La figure 7-1 nous montre que les temps de convergence sont équivalents pour les critères de décision WEIGHT, AS-PATH, ORIGIN et RID : environ 3.35 msec. Par contre, on obtient des valeurs plus élevées lorsque la décision est basée sur le MED, que le DUT utilise ou non l'option **`bgp deterministic-med`**. La différence observée, de l'ordre de 0.32 msec, est supérieure à l'écart-type (0.03 msec), et correspond à une augmentation des temps de convergence supérieure à 9.5%.

7.1.2. Autres conditions de redistribution...

L'étude de la redistribution des annonces de préfixes (A-EBGP).nous a permis de mettre en évidence un certain nombre de caractéristiques du fonctionnement du routeur. Pour affiner notre compréhension du fonctionnement du processus de décision, nous répété ces tests dans les cas de l'annonce d'un préfixe à un voisin de type IBGP (A-IBGP), du retrait d'un préfixe vers un voisin de type EBGP (W-EBGP), et du retrait d'un préfixe vers un voisin de type IBGP (W-IBGP)

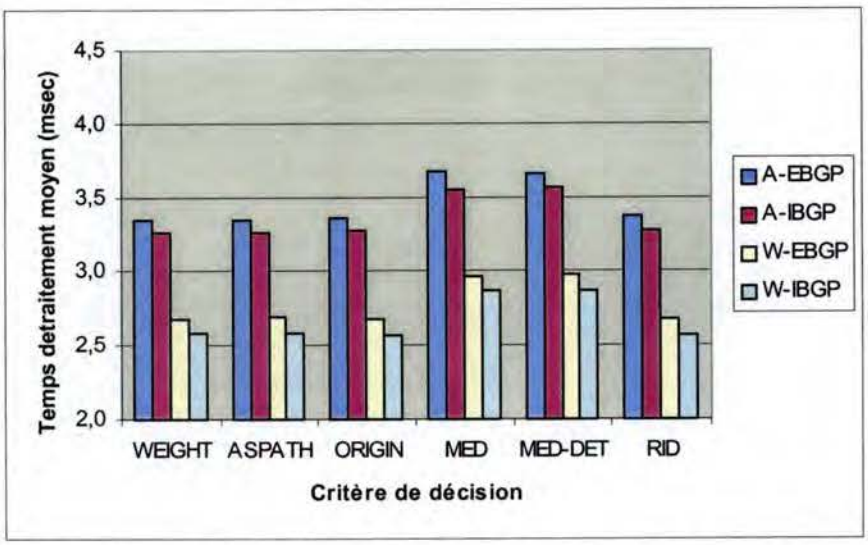

Figure 7-2 : Processus de décision – Comparaison des temps de traitement moyen selon la nature de l'UPDATE et le type de pair en aval du DUT.

La figure 7-2 nous montre que pour chaque situation étudiée, le processus de décision semble fonctionner de façon semblable. Par exemple, dans le cas d'un retrait de préfixe vers un voisin IBGP[7] (W-IBGP), les temps de traitement moyens sont équivalents pour une décision basée sur le WEIGHT, l'AS-PATH, l'origine de la route (ORIGIN) et l'identifiant du routeur (RID). Par contre, ils sont plus élevés pour une décision basée sur le MED, que l'on utilise ou non l'option `bgp deterministic-med`. L'augmentation observée est d'environ 0.30 msec. La même constatation peut être faite dans les autres situations.

De plus, quel que soit le critère de décision examiné, les temps de traitement des retraits de route sont systématiquement plus bas que ceux des annonces de route, que le voisin en aval du DUT soit de type externe (EBGP) ou interne (IBGP). L'écart est d'environ 0.69 msec. Nous supposons que cet écart pourrait s'expliquer par le fait qu'il n'est pas nécessaire de calculer un degré de préférence dans le cas de retrait de route, mais nous n'avons pas pu vérifier cette hypothèse.

Enfin, quel que soit le critère de décision examiné, les temps de traitement vers un voisin de type IBGP sont plus bas que ceux vers un voisin de type EBGP, aussi bien pour les retraits de route (W) que pour les annonces (A). L'écart est d'environ 0.09 msec. Cette différence pourrait s'expliquer par le fait que l'AS-PATH ne doit pas être modifié en cas d'annonce à un voisin IBGP. Pour vérifier cette hypothèse, on pourrait comparer le temps

[7] Dans la pratique, comme la table BGP du DUT contient deux instances du même préfixe, le retrait de route par le pair TEST a pour conséquence l'annonce une route moins bonne.

de traitement obtenu en cas d'annonce à un voisin IBGP lorsque l'option `as-path prepend`[8] est utilisée, avec celui obtenu en cas d'annonce à un voisin EBGP.

Nos tests nous ont donc permis de mettre en évidence le fait que, quelle que soit la situation, les temps de convergence des préfixes étaient semblables lorsque le critère de décision utilisé pour sélectionner une route était le WEIGHT, l'AS-PATH, l'ORIGIN et le RID et qu'ils étaient plus élevés lorsque le critère de décision était le MED. Nos mesures ne nous ont pas permis de déterminer si le processus de décision des routeurs Cisco fonctionne en plusieurs phases, comme prévu dans le [RFC1771], ou plutôt en examinant les critères l'un à la suite de l'autre jusqu'à ce qu'un critère permette de sélectionner une et une seule route pour un préfixe.

Nous avons également mis en évidence le fait que les temps de traitement mesurés pour une sélection basée sur le MED étaient supérieurs à ceux mesurés pour une sélection basée sur le RID, qui est la dernière étape du processus de décision. Ce phénomène pourrait s'expliquer par le fait qu'en cas d'inégalité des MED, le processus doit en plus vérifier si les deux UPDATE proviennent du même AS. Et nous faisons l'hypothèse que cette vérification prend 0.32 msec. Deux tests complémentaires nous permettraient de valider cette hypothèse. Le premier consiste à effectuer une sélection basée sur le MED lorsque l'option `bgp always-compare-med`[9] est configurée. Le second consiste à effectuer une sélection basée sur le RID, lorsque les MED sont inégaux et que les UPDATE ont été annoncés par des AS différents.

Nos mesures nous permettent déjà de faire quelques hypothèses sur la durée de certains processus internes. Les phases 1 et 3 ont été minimisées par l'application des politiques ACCEPT-ALL/ADVERTISE-ALL. Nous estimons néanmoins la durée du calcul du degré de préférence (phase 1) à 0.69 msec. L'application des règles d'arbitrage (phase 2) se fait de manière transparente lorsque la décision est basée sur le WEIGHT, l'AS-PATH, l'ORIGIN ou le RID. Elle prend 0.30 msec lorsque la décision est basée sur le MED, et ce délai peut être attribué à la comparaison des AS qui ont annoncé le préfixe. La modification de l'AS-PATH (phase 3) lorsqu'une route est redistribuée à un pair externe est estimée à 0.10 msec. Le reste du temps de traitement (2.56 msec) n'a pas pu être attribué à un processus particulier, mais il est probable qu'il se répartit également sur les trois phases du processus de décision.

7.2. Effet du filtrage de route et de la manipulation d'attributs

Le filtrage de route et la manipulation d'attributs sont deux outils extrêmement puissants qui permettent aux routeurs BGP de refléter des décisions stratégiques dans la manière dont sont routés les préfixes. Ces techniques sont habituellement utilisées pour contrôler le trafic qui traverse un système autonome ou pour modifier les comportements de routage. Elles peuvent s'appliquer aussi bien à l'entrée qu'à la sortie du routeur.

Une stratégie de routage peut s'envisager selon deux axes : soit elle vise à accepter ou refuser des préfixes en fonction de certains critères, soit elle a pour objectif de manipuler certains attributs afin d'influencer une décision de routage.

Dans cette série de test, nous chercherons à mettre en évidence l'impact de l'application de diverses politiques de routage à l'entrée ou à la sortie du routeur sur les temps de convergence de préfixes choisis selon le critère du chemin le plus court. Les mesures de convergence effectuées au moyen de tests black-box impliquent que les messages qui entrent dans le DUT doivent en ressortir : nous utiliserons dès lors des politique de routage qui permettent aux préfixes testés d'être acceptés puis redistribués aux pairs en aval.

La figure 4-1 nous a montré que les composants qui permettent d'appliquer les politiques à l'entrée ou à la sortie du routeur sont séparés. Ces deux composants ont été étudiés séparément, mais comme nous avons utilisé la même méthodologie dans les deux cas, nous allons représenter les résultats obtenus sur le même schéma.

[8] Il s'agit d'un route-map que l'on utilise habituellement pour modifier la longueur de l'AS-PATH afin d'influencer la décision de routeurs dans d'autres AS.
[9] Par défaut, le MED permet de sélectionner une route uniquement lorsque les UPDATE à comparer proviennent du même AS

7.2.1. Influence du nombre de critères

Dans un premier temps, nous allons vérifier si le nombre de critères examinés avant d'accepter une route influence les temps de traitement des préfixes. Nous en déduirons le nombre nécessaire et suffisant de critères à utiliser pour mettre en évidence un effet dans les tests comparatifs sans avoir à utiliser des configurations trop contraignantes.

Deux cas de figures seront étudiés : le filtrage de route, pour lequel la route est acceptée après avoir examiné un certain nombre de critères (clauses `match`), et la manipulation d'attributs, pour lequel un attribut d'une route est modifié après avoir examiné un certain nombre de critères(clauses `match` et `set`).

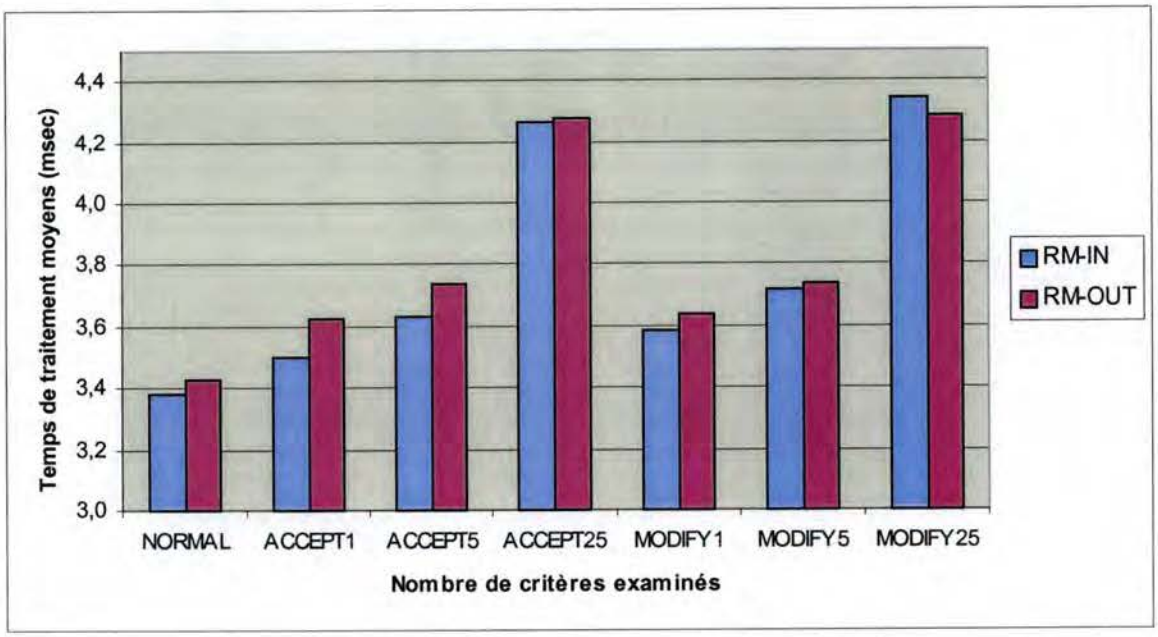

Figure 7-3 : Comparaison de l'effet du nombre de critères examinés sur les temps de traitement pour le filtrage de routes (ACCEPT) et la manipulation d'attributs (MODIFY) appliqués à l'entrée (RM-IN) ou à la sortie du routeur(RM-OUT).

La figure 7-3 nous montre que, globalement, la redistribution des préfixes adopte un comportement semblable à l'entrée (***RM-IN***) et à la sortie (***RM-OUT***) du routeur : la moyenne du temps de traitement des préfixes augmente en fonction du nombre de critères examinés, aussi bien pour le filtrage de route (***ACCEPT***), que pour la manipulation d'attribut (***MODIFY***). Quel que soit le cas considéré, l'effet le plus important est observé lorsque 25 critères sont examinés : l'augmentation du temps de traitement par rapport aux tests de référence (de l'ordre de 0.85 msec) est alors proche de 25%. On obtient déjà un effet de l'ordre de 9% lorsque 5 critères ont été examinés : nous utiliserons donc ce nombre de critère pour comparer différents outils de filtrage de route.

On notera toutefois quelques différences selon que la politique est appliquée à l'entrée ou à la sortie du routeur. Tout d'abord, les temps de traitement calculés pour le filtrage de route appliqué à la sortie du routeur (***ACCEPT***, ***RM-OUT***) semblent légèrement plus élevés que ceux calculés pour le filtrage de route appliqué à l'entrée du routeur (***ACCEPT***, ***RM-IN***). La différence est plus nette lorsque 1 et 5 critères sont examinés : elle vaut respectivement 0.13 et 0.11 msec, ce qui correspond à une augmentation d'environ 3.5%. Cette différence entre l'entrée et la sortie du routeur n'est pas observée dans le cas de la modification de routes. On peut toutefois déduire de ces observations que les machines qui permettent d'appliquer des politiques à l'entrée et à la sortie du routeur ne fonctionnent pas de manière absolument identique, mais les tests n'ont pas été poussés assez loin pour en conclure qu'il s'agit bien effectivement de modules différents. Nous n'avons pas non plus d'hypothèse pour expliquer cette différence de comportement.

Lorsqu'on applique une politiques de routage à l'entrée du routeur, pour un même nombre de critères examinés, la moyenne des temps de traitement est plus élevée si on modifie un attribut d'une route (***MODIFY***, ***RM-IN***) que si on accepte simplement une route (***ACCEPT***, ***RM-IN***). Nous avons attribué la différence (0.082 msec) à l'application de la clause `set` dans le cas de la modification d'attribut ; pour vérifier cette hypothèse, il aurait été intéressant de réaliser la manipulation d'attribut de manière inconditionnelle, c'est-à-dire en n'utilisant pas de clause `match` dans le `route-map`. Par contre, si on applique une politique de routage à la sortie du routeur, les moyennes des temps de traitement obtenues pour le filtrage de route (***ACCEPT***, ***RM-OUT***) ou la manipulation d'attribut (***MODIFY***, ***RM-OUT***) sont équivalentes.

Sur base de nos différentes mesures, nous avons essayé d'établir la durée de traitement de certaines opérations impliquées dans le filtrage de route et la manipulation d'attributs. A l'entrée du routeur, nous avons une partie fixe de 0.090 msec que nous ne pouvons pas attribuer à un processus particulier. L'examen de chaque clause match est estimé à 0.031 msec ; la durée du filtrage de route dépendra du nombre de clauses `match` à examiner. L'application de la clause `set` lors de la manipulation d'attribut est estimée à 0.082 msec. A la sortie du routeur, la partie fixe est estimée à 0.195 msec. L'examen de chaque clause `match` est estimé à 0.029 msec. Nous supposons que l'application de la clause `set` est transparente à la sortie du routeur.

7.2.2. Comparaison de divers outils

Plusieurs outils permettent d'appliquer des politiques de routage. Les `filter-list` filtrent les annonces sur base d'informations contenues dans l'AS-PATH. Les `distribute-list` et les `prefix-list` filtrent les annonces sur base d'informations contenues dans le NLRI. Les `route-map`, quant à eux, permettent non seulement de filtrer les routes sur base de l'AS-PATH, du NLRI et de l'attribut COMMUNITY, mais en plus, ils permettent de manipuler les attributs BGP. Or, malgré le potentiel de flexibilité important des `route-map`, des ISP importants utilisent plus facilement des outils classiques comme les `prefix-list` ou les `distribute-list` pour réaliser le filtrage de routes sur leurs routeurs de production.

Il nous a donc paru intéressant de comparer les performances de convergence de ces différents outils. Chaque outil présente de nombreuses possibilités de configuration. Afin de minimiser le nombre de tests à réaliser, nous avons défini trois groupes selon le critère de sélection utilisé pour le filtrage, à savoir l'AS-PATH (`filter-list` et `route-map` basé sur l'AS-PATH), le NLRI (`ditribute-list`, `prefix-list` et `route-map` basé sur le NLRI) et l'attribut COMMUNITY (`route-map`). Cette étude sera limitée au filtrage de route dans le cas où cinq critères sont examinés avant d'accepter une route.

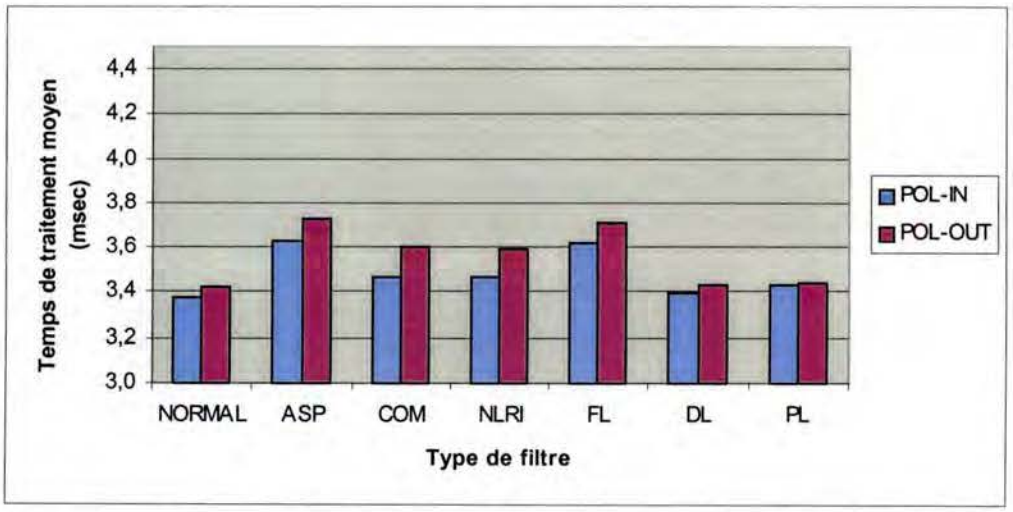

Figure 7-4 : Comparaison des temps de traitement moyens obtenus selon le type de filtre utilisé pour des politiques appliquées à l'entrée (POL-IN) ou à la sortie du routeur (POL-OUT)

La figure 7-4 nous montre que, la différence de comportement observée précédemment entre l'entrée et la sortie du routeur se retrouve pour certains filtres (tous les `route-map` et le `filter-list`). Malgré tout, les comportements sont assez semblables que le filtrage de route soit effectué à l'entrée ou à la sortie du routeur :

- les temps de traitement sont équivalents pour les `route-map` basés sur l'AS-PATH (ASP) et les `filter-list` (FL), aussi bien à l'entrée (3.62 et 3.60 msec) qu'à la sortie du routeur (3.71 et 3.72 msec) ;
- les temps de traitement sont plus élevés pour les `route-map` basés sur le NLRI (NLRI) que pour les `distribute-list` (DL) et `prefix-list` (PL) qui réalisent le filtrage de route en utilisant le même type de critère (le NLRI) : les valeurs obtenues sont respectivement de 3.47, 3.39 et 3.43 msec lorsque le filtrage de route est réalisé à l'entrée du routeur et 3.60, 3.44 et 3.44 msec lorsqu'il est réalisé à la sortie du routeur ;
- les temps de traitement sont plus élevés pour les `route-map` basés sur l'AS-PATH (ASP) que pour ceux basés sur le NLRI (NLRI) ou l'attribut COMMUNITY (COM) ;
- les temps de traitement sont plus élevés pour les `filter-list` (FL), qui effectuent le filtrage sur base d'une information contenue dans l'AS-PATH, que pour les `distribute-list` (DL) et `prefix-list` (PL), qui réalisent le filtrage de routes sur base d'une information contenue dans le NLRI. Les temps de traitement obtenus avec une `distribute-list` (DL) et une `prefix-list` (PL) sont du même ordre de grandeur que ceux des tests de référence.

On notera encore que lorsqu'on applique des politiques à la sortie du routeur, les temps de traitement semblent plus élevés que lorsqu'on applique les mêmes politiques à l'entrée du routeur, du moins en ce qui concerne les `route-map` et les `filter-list` (FL). Cette différence de comportement entre l'entrée et la sortie du routeur n'est pas observée pour les `distribute-list` (DL) et `prefix-list` (PL).

Le choix d'un outil pour appliquer des politiques de routage sera donc dicté par les objectifs que l'on veut réaliser. Les `route-map` sont les seuls qui permettent de modifier des attributs. Ils sont également les seuls à permettre le filtrage de route sur base de l'attribut COMMUNITY. De leur côté, les `prefix-list` et `distribute-list` sont plus efficaces que le `route-map` pour réaliser le filtrage de route sur base de l'information contenue dans le NLRI. Enfin, si l'objectif est de réaliser un filtrage sur base de la valeur de l'AS-PATH, on peut utiliser indifféremment un `filter-list` ou un `route-map`. Ces résultats pourraient expliquer pourquoi, malgré la souplesse d'utilisation offerte par les `route-map`, la configuration de routeurs du backbone d'ISP importants contiennent encore une large part de `distribute-list` et `prefix-list`.

Les résultats expérimentaux ne nous permettent pas de conclure que le composant qui permet d'appliquer les politiques à l'entrée du routeur est réellement différent du composant qui permet d'appliquer les politiques à la sortie du routeur ; mais le fait que l'on observe des temps de traitement plus élevés lorsque les politiques sont appliquées à la sortie du routeur, ainsi que la différence de comportement observée dans le cas de la manipulation d'attribut montrent qu'ils ne fonctionnent pas de la même manière dans les deux cas.

7.3. Effet de l'agrégation de routes

Selon le [RFC1771], l'agrégation de routes et la réduction d'informations peuvent se produire optionnellement pendant la 3^{e} phase du processus de décision, après que les routes aient été installées dans la base d'information du routeur. L'agrégation fait partie intégrante du processus de décision pour diminuer le nombre d'informations de routage qui seront incluses dans les Adj-RIBs-Out.

L'agrégation a pour effet de combiner les caractéristiques de plusieurs routes différentes, de sorte qu'une seule route sera annoncée. Les attributs MED et NEXT-HOP doivent être identiques pour que les préfixes puissent être agrégés. Le [RFC1771] gère également la manière dont les attributs ORIGIN et AS-PATH sont traités lorsqu'ils sont différents.

L'agrégation de routes peut s'accompagner ou non de l'annonce des routes spécifiques. Pour les besoins expérimentaux, nous mesurerons l'effet de l'utilisation de la commande `aggregate-address` sur l'envoi des annonces de préfixes spécifiques ; nous ne mesurerons pas le temps de formation de l'agrégat.

Nous avons testé quatre variantes de la commande `aggregate-address` :

- `aggregate-address` *`address mask`* (***AM***) réalise l'agrégation simple
- `aggregate-address` *`address mask`* `as-set` (***AMAS***) réalise l'agrégation et ajoute à l'agrégat l'information d'AS-PATH contenue dans les préfixes spécifiques afin d'éviter les boucles de routage
- `aggregate-address` *`address mask`* `as-set advertise-map` *`route-map-name`* (***ADVERTISE***) réalise l'agrégation en spécifiant quelle information de l'AS-PATH doit être retenue dans l'agrégat. Chaque agrégat constitue alors un sous-ensemble de préfixes partageant un AS
- `aggregate-address` *`address mask`* `attribute-map` *`route-map-name`* (***ATTR***) réalise l'agrégation et modifie un des attributs BGP dans l'agrégat

Nous avons étudié le comportement d'un routeur (DUT) dans deux situations d'agrégation différentes. Dans le premier cas, les voisins en aval du DUT lui annoncent des préfixes qu'il a le droit d'agréger. Cette situation peut se produire, par exemple, lorsque tous les clients d'un ISP annoncent exclusivement des préfixes qui constituent des subdivisions de son propre espace d'adressage. Dans le deuxième cas, les voisins du DUT annoncent exclusivement des préfixes qu'il n'a pas le droit d'agréger : cela peut se produire dans le cas d'un ISP de transit, dont les voisins sont eux-mêmes d'autres ISP, et qui agrège uniquement les informations de routage de sa propre infrastructure.

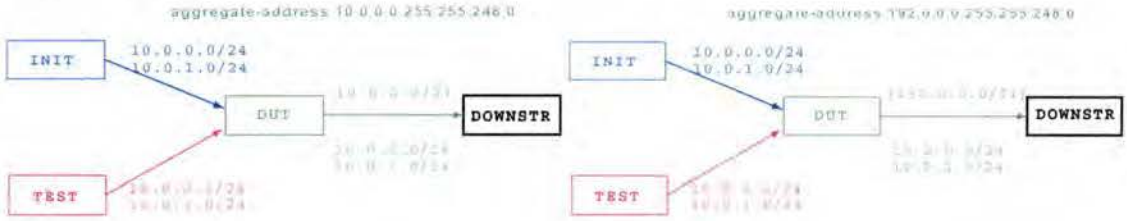

Figure 7-5 : Représentation des deux catégories d'agrégation testées

La figure 7-5 schématise les deux catégories d'agrégation testées. A gauche, le DUT agrège les préfixes qu'il a appris de ses voisins INIT et TEST : il annonce l'agrégat formé ainsi que les préfixes spécifiques. A droite, le DUT agrège des préfixes qui appartiennent à un autre espace d'adressage que les préfixes annoncés par INIT et TEST. Dans ce cas, aucun agrégat n'est annoncé, puisque la table BGP du DUT ne contient pas de préfixes spécifiques pour cet espace d'adressage.

Pour affiner notre compréhension du fonctionnement du processus d'agrégation, les tests de chaque catégorie ont été réalisés en formant des agrégats avec des longueurs de masque différentes : les agrégats avec un masque « long » (/21) résument les informations de 8 préfixes spécifiques (/24) et les agrégat avec un masque « court » (/16) résument les informations de 256 préfixes spécifiques.

7.3.1. Agrégation de son propre espace d'adressage avec un masque long (/21)

Lorsqu'on utilise la commande `aggregate-address` pour former des agrégats avec un masque de 21 bits, les préfixes spécifiques sont agrégés par groupe de 8. Pour couvrir l'ensemble des 1000 préfixes de notre jeu de test, le DUT aurait donc du former 125 agrégats différents. Le [RFC1878] nous donne la table de correspondance du nombre de sous-réseaux possibles pour différentes longueurs de masque.

Pour simplifier la configuration du DUT, nous avons limité à 32 le nombre d'agrégats à former ; nous avons utilisé un pas de 32 entre chaque agrégat pour les répartir au mieux sur l'ensemble des préfixes testés. Les agrégats annoncés sont donc 10.0.0.0/21

(préfixes 10.0.*0-7*.0/24)[10], 10.0.32.0/21 (préfixes 10.0.*32-39*.0/24), ... Les agrégats intermédiaires (10.0.8.0/21...) sont omis. Cela signifie que seul ¼ des préfixes spécifiques appris par le DUT sont inclus dans un des agrégats.

Il est peu probable qu'on rencontre dans la réalité une telle configuration, caractérisée par des agrégats discontinus : un ISP a intérêt à subdiviser son espace d'adressage et à le déléguer à ses clients de sorte que les préfixes annoncés à un même routeur soient aussi contigus que possible, ce qui rationalise les possibilités d'agrégation.

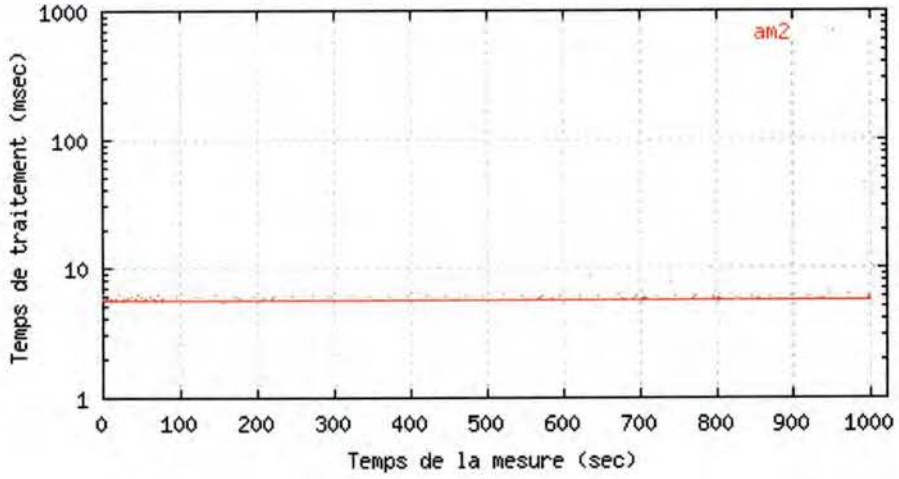

Figure 7-6 : Temps de traitement des UPDATE BGP obtenus avec la variante AM de la commande aggregate-address

La figure 7-6 représente la distribution des temps de traitement des messages BGP au cours du déroulement de l'expérience lorsque le DUT est configuré avec la variante ***AM*** de la commande `aggregate-address`. La majeure partie des temps de traitement des préfixes se situe aux alentours de 5.62 msec. De plus, malgré le fait que tous les préfixes appris par le DUT ne sont pas inclus dans un agrégat, les temps de traitement sont assez homogènes : on peut supposer que c'est du au fait que ces préfixes appartiennent tous au même espace d'adressage. Le profil obtenu avec la variante ***ATTR*** de la commande `aggregate-address` (non représenté) est très semblable.

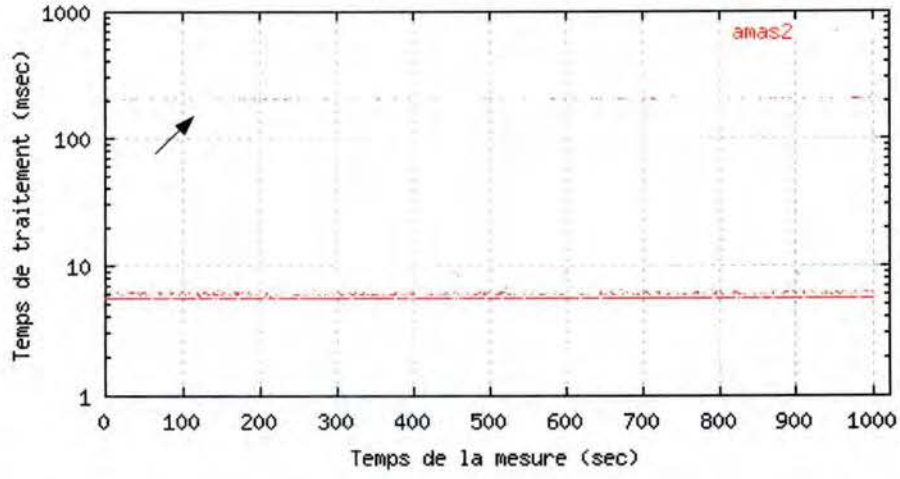

Figure 7-7 : Temps de traitement des UPDATE BGP obtenus avec la variante AMAS de la commande aggregate-address

[10] La notation N.N.*0-7*.0, indique les adresses de 8 sous-réseaux de classe C ou équivalent (/24) inclus dans le réseau N.N.0.0/21. Une notation similaire apparaît dans le [RFC1878]

La figure 7-7 représente la distribution des temps de traitement des messages BGP au cours du déroulement de l'expérience lorsque le DUT est configuré avec la variante ***AMAS*** de la commande `aggregate-address`. La majeure partie des temps de traitement des préfixes se situe également aux alentours de 5.64 msec, mais une part non négligeable (environ 10%) se situe aux alentours de 210 msec. La double distribution des temps de traitement observée dans ce cas-ci, et le faible pourcentage de valeurs élevées (10%) pourraient peut-être être attribués au fait que tous les préfixes entrant dans le DUT ne sont pas traités de la même manière, puisque seuls ¼ des préfixes sont utilisés pour former un agrégat. Nous n'avons pas eu l'occasion de vérifier cette hypothèse. Les mêmes conclusions sont également valables avec la variante ***ADVERTISE*** (profil non représenté).

Ce qui différencie les deux catégories est que de l'option `as-set` n'est pas utilisée dans le premier cas (***AM***, ***ATTR***), mais bien dans le second (***AMAS***, ***ADVERTISE***). L'option `as-set` permet d'inclure dans l'agrégat l'information d'AS-PATH contenue dans l'ensemble des préfixes spécifiques qui le composent.

Pour comprendre ce qui s'est passé, nous avons examiné le log[11] de la séquence d'événements d'un des tests réalisés avec la variante ***AMAS*** de la commande `aggregate-address`. Tout d'abord, nous avons constaté que le nombre d'annonces d'agrégats (138) est inférieur à ce qui est attendu[12]. Nous avons ensuite mis en évidence le fait que, les temps de traitement des préfixes qui ne doivent pas être inclus dans un agrégat (10.0.*8-31*.0/24, ...) est généralement de l'ordre de 5.62 msec, alors qu'il est de l'ordre de 6.3 msec pour ceux qui doivent être inclus dans l'agrégat (10.0.*0-7*.0/24, ...). Etant donné que nous n'avons pas pu établir la même corrélation dans les tests réalisés avec la variante ***AM***, nous attribuons la différence observée (0.68 msec) à l'utilisation de l'option as-set, et probablement à l'adaptation de l'AS-PATH de l'agrégat.

Nous avons ensuite examiné quels événements étaient déclenchés par la réception d'un préfixe spécifique qui doit être inclus dans un agrégat par le DUT. (Un extrait du log pour l'agrégat 10.0.32.0/21 se trouve en annexe J). Nous avons pu classer ces événements en trois catégories. La première est l'annonce par le DUT du préfixe spécifique après 6.3 msec. La deuxième est l'annonce par le DUT du préfixe spécifique après 6.3 msec et de l'agrégat après 210 msec. La troisième est l'annonce par le DUT de l'agrégat après 6.3 msec et du préfixe spécifique après 210 msec. Il est difficile de comprendre pourquoi l'agrégat n'est pas systématiquement annoncé. Il est également difficile de comprendre pourquoi ces deux événements sont dissociés, puisque manifestement, la modification et l'annonce de l'agrégat peut se faire dans le même laps de temps que l'annonce du préfixe spécifique, et pourquoi le délai qui sépare ces deux événements est aussi important.

7.3.2. Comparaison de différentes conditions d'agrégation

Nous avons voulu investiguer de façon plus approfondie l'influence de la commande `aggregate-address` sur les temps de traitement des préfixes. Nous avons donc répété les tests précédents dans les conditions suivantes :

- le DUT forme des agrégats qui appartiennent au même espace d'adressage que les préfixes testés avec un masque de 21 bits (*Other/21*). Les agrégats formés (192.0.0.0/21, 192.0.32.0/21,...) sont séparés par un pas de 32 préfixes : ils ne sont donc pas contigus.
- le DUT forme des agrégats qui appartiennent au même espace d'adressage que les préfixes testés avec un masque de 16 bits (10.0.0.0/16[13], 10.1.0.0/16...). (*Normal/16*)
- le DUT forme des agrégats qui appartiennent à un espace d'adressage différent de celui des préfixes testés avec un masque de 16 bits (192.0.0.0/16, 192.1.0.0/16...). (*Other/16*)

[11] Mesure des temps de passage en amont et en aval du routeur réalisée avec **`tethereal`**

[12] L'AS-PATH de tous les préfixes est différent. Chaque nouvelle annonce provoque donc une modification de l'AS-PATH de l'agrégat ; pour 32 agrégats concernant chacun 8 préfixes différents, on peut raisonnablement s'attendre à dénombrer 256 annonces d'agrégats.

[13] Cet agrégat résume les préfixes 10.0.*0-255*.0/24

La figure 7-8 représente la valeur médiane du temps de traitement pour différents scénarios d'agrégation et pour différentes variantes de la commande `aggregate-address`. Les temps de traitement obtenus lorsque le DUT ne réalise aucune agrégation (***REF***) sont représentés à des fins de comparaison. Ils sont de l'ordre de 3.38 msec.

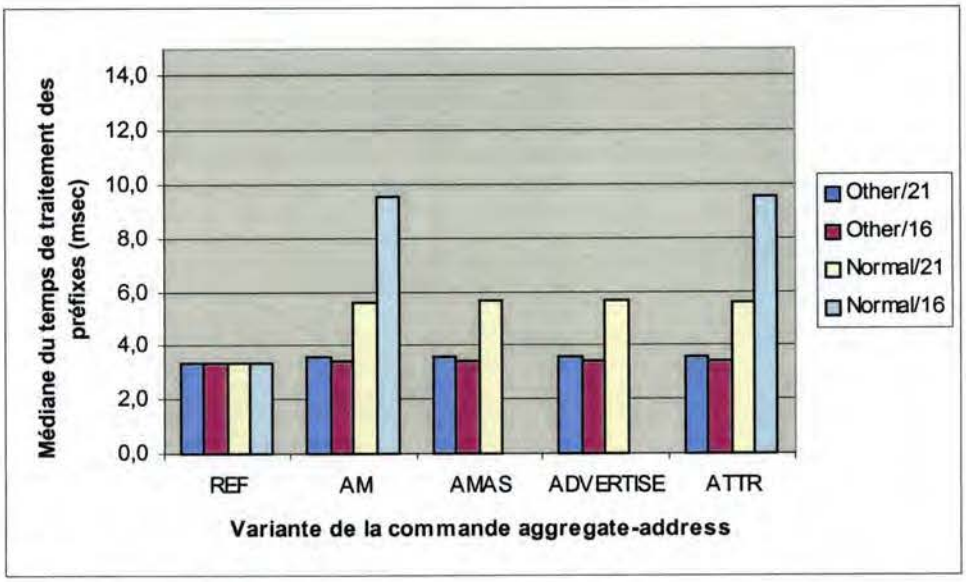

Figure 7-8 : Comparaison de différentes conditions d'agrégation sur la médiane du temps de traitement des messages BGP pour diverses variantes de la commande aggregate-address

Lorsque le DUT forme des agrégats avec un masque de 21 bits, dans un espace d'adressage différent de celui des préfixes testés (*Other/21*), les temps de redistribution des préfixes sont de l'ordre de 3.58 msec pour toutes les variantes de la commande `aggregate-address` (***AM, AMAS, ADVERTISE, ATTR***). Cela représente une augmentation d'environ 6% par rapport aux tests de référence. Lorsque le DUT forme des agrégats avec un masque de 16 bits dans un espace d'adressage différent de celui des préfixes testés (*Other/16*), les temps de traitement des UPDATE BGP sont de l'ordre de l'ordre de 3.42 msec, ce qui représente une augmentation d'environ 1.3% par rapport aux tests de référence. Dans ces conditions nous n'avons pas pu mettre en évidence des valeurs de convergence élevées : aucun agrégat n'est formé et par conséquent, le DUT ne doit pas recalculer un nouvel AS-PATH pour l'agrégat chaque fois qu'un nouveau préfixe apporte une information nouvelle.

Lorsque le DUT forme des agrégats avec un masque de 21 bits dans le même espace d'adressage que les préfixes testés (*Normal/21*) le temps de traitement des UPDATE BGP est de l'ordre de 5.62 msec, ce qui représente une augmentation de 66% par rapport aux tests de référence. Cette augmentation est importante, d'autant plus qu'il s'agit de la convergence des préfixes spécifiques, qui ont été appris par le DUT, et non du temps mis pour former l'agrégat. Cette augmentation (*Other/21*) ne tient pas compte des valeurs extrêmes observées dans le cas des variantes ***AMAS*** et ***ADVERTISE***. (voir section 7.3.1)

Lorsque le DUT forme des agrégats avec un masque de 16 bits dans le même espace d'adressage que les préfixes testés (*Normal/16*), le temps de traitement des UPDATE BGP est de 9.53 msec, ce qui représente une augmentation de 181% par rapport aux tests de référence. Il n'a pas été possible de calculer les temps de convergence pour les variantes ***AMAS*** et ***ADVERTISE*** dans cette situation : au cours du déroulement du test (phases d'initialisation et de test), un nombre important de messages NOTIFICATION[14] a été envoyé par le DUT au routeur DOWNSTREAM, suivis par une procédure d'ouverture de session entre ces deux pairs. Nous n'avons pas conservé le contenu du message NOTIFICATION pour déterminer la cause d'erreur.

[14] Un message NOTIFICATION est toujours suivi par une fermeture de session

Nous avons, par ailleurs, pu observer des AS-PATH avec un air rébarbatif pour quelques annonces de préfixes agrégés : ils étaient caractérisés par une longueur excessive, de nombreux AS-SET, et des AS qui n'appartiennent pas à la liste des AS possibles[15]. Cela peut s'expliquer par l'utilisation de l'option `as-set` par ces deux variantes de la commande `aggregate-address`. Cette option a pour objectif de former un ensemble qui permet de représenter les informations d'AS-PATH de tous les préfixes qui forment l'agrégat. Or, les préfixes que nous testons sont conçus pour pouvoir être testés de manière unique. Pour ce faire, ils diffèrent par le contenu de l'AS-PATH. Cela signifie que, dans ce contexte, l'attribut AS-PATH pourrait contenir jusqu'à 256 AS différents, ce qui ne semble pas bien géré par les routeurs Cisco. En pratique, les nouvelles recommandations en matière d'allocation de l'espace d'adressage donnent peu de chances à une telle situation, où chacun des préfixes qui forme un agrégat porte un attribut AS-PATH différent, de se produire dans la réalité. Il n'empêche que l'on peut se demander combien de préfixes avec un AS-PATH différent peuvent être gérés sans problèmes lorsque l'option as-set est utilisée.

L'utilisation de la commande `aggregate-address` influence fortement les temps de traitement des préfixes spécifiques, et cette influence est d'autant plus forte que la distance[16] qui sépare les préfixes spécifiques de l'agrégat est grande. Nous avons estimé la durée de la phase d'agrégation à 2.24 msec lorsque l'agrégat regroupe des blocs de 8 préfixes et à 5.95 msec lorsque l'agrégat regroupe des blocs de 256 préfixes. Sur base des mesures faites l'agrégation porte sur un autre espace d'adressage que les préfixes traités par le DUT, nous faisons l'hypothèse que le temps pris pour vérifier si un préfixe doit ou non être inclus dans un agrégat est de l'ordre de 0.20 msec. Ce nombre varie probablement en fonction du nombre d'agrégats à former, mais nos tests ne nous ont pas permis confirmer cette hypothèse. Nous estimons, enfin, à 0.68 msec l'impact de l'utilisation de l'option `as-set` : ce délai supplémentaire est probablement utilisé pour adapter l'AS-PATH dans l'agrégat. Nous avons constaté, lorsque l'option as-set est utilisée, que le préfixe spécifique et l'agrégat ne sont jamais annoncés en même temps : une des deux annonces est systématiquement retardée d'environ 204 msec, mais rien ne permet de prédire laquelle, ni ce qui se passe pendant ce laps de temps.

7.4. Influence du MinRouteAdverTimer (advertisement interval)

Le paramètre MinRouteAdvertisementInterval (MRAI) détermine la quantité minimale de temps qui doit séparer deux annonces de route pour une même destination faites par un seul annonceur BGP. [RFC1771] L'implémentation de ce timer n'est pas spécifiée. En pratique, toute technique qui permet d'assurer que l'intervalle entre deux messages UPDATE qui concernent un même ensemble de destinations envoyés par un même annonceur BGP sera au moins le MRAI et assure une limite supérieure constante à cet intervalle est acceptable.

Ce timer ne s'applique pas à l'intérieur des systèmes autonomes, étant donné qu'une convergence rapide y est nécessaire. De même, pour éviter les « trous noirs » de longue durée, il ne s'applique pas aux retraits explicites des routes devenues inaccessibles, c'est-à-dire aux destinations listées dans le champ WITHDRAWN des messages UPDATE. Ce timer ne limite pas la vitesse de la sélection de routes, mais bien la vitesse de redistribution des annonces sélectionnées. Si de nouvelles routes sont sélectionnées à de multiples reprises en attendant l'expiration du MRAI, seule la dernière route sélectionnée sera annoncée, ce qui permet de limiter les oscillations de routes.

Le test vise à comparer les temps de traitement obtenus pour différentes valeurs du MRAI :

- 30 sec est la valeur recommandée par le [RFC1771] pour les mises à jour envoyées entre systèmes autonomes différents ; c'est également la valeur utilisée par défaut par Cisco pour les sessions EBGP [Par01].
- 5 sec est la valeur utilisée par Cisco pour les sessions IBGP.
- 0 sec permet de minimiser le temps de redistribution.
- Nous avons testé quelques valeurs intermédiaires afin d'affiner les commentaires.

[15] Les valeurs d'AS utilisées dans nos tests sont toutes supérieures à 64999 (pool privé) ; or l'AS-PATH de certains agrégats contenait des valeurs du pool public

[16] La différence de longueurs de masques

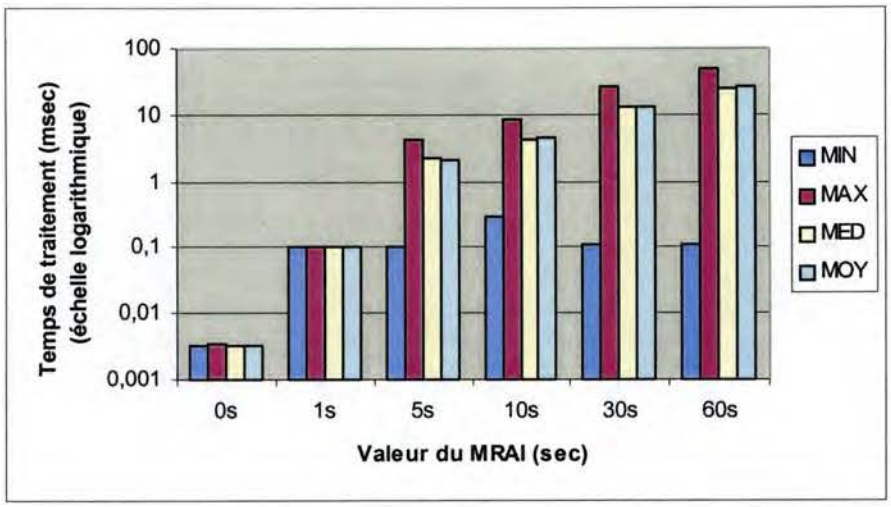

Figure 7-9 : Influence du MRAI sur les temps de traitement des UPDATE BGP

La figure 7-9 nous montre que :

- les valeurs moyennes des temps de traitement s'échelonnent entre 3.39 msec pour un MRAI de 0 sec et 25.59 sec pour un MRAI de 60 sec. On obtient une moyenne de 100 msec lorsque le MRAI vaut 1 sec : dans ces conditions, l'intervalle entre les UPDATE est égal à la valeur du MRAI.
- les valeurs minimales sont de 3.33 msec pour un MRAI de 0 sec, de 302.7 msec pour un MRAI de 10 sec et environ 100 msec pour les autres valeurs de MRAI
- les valeurs maximales sont respectivement de 3.45 msec, 103.41 msec, 4.3 sec, 8.3 sec, 26.1 sec et 51.1sec pour des MRAI de 0, 1, 5, 10, 30 et 60 sec.

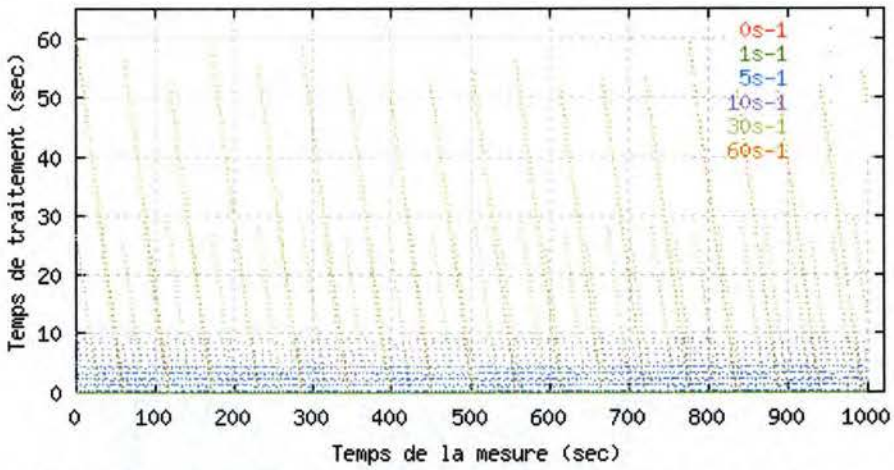

Figure 7-10 : Temps de traitement mesurés au cours du déroulement de l'expérience pour différentes valeurs de MRAI

La figure 7-10 nous montre que, pour des valeurs de MRAI comprises entre 5 et 60 sec, les temps de traitement montrent une décroissance linéaire au cours du temps. Ils ont pour limite supérieure la valeur du MRAI et pour limite inférieure environ 100 msec.

Notons encore que pour les valeurs de MRAI de 30 et 60 sec, certaines annonces ne sont pas redistribuées (résultats non représentés). Cette absence de redistribution coïncide avec la présence de trames mal-formées dans les log. Nous ne pouvons cependant pas dire s'il s'agit d'un artéfact de la méthode de mesure, ou s'il y a réellement des pertes de messages avec ces valeurs de MRAI.

Cette série de tests nous a donc montré que en pratique, avec un MRAI configuré à 0 sec, la redistribution des préfixes par routeur pouvait se faire en quelques millisecondes. Par

comparaison, lorsque le MRAI est configuré à 30 sec, conformément aux spécifications du [RFC1771] et à la valeur utilisée par défaut sur les routeurs Cisco, la redistribution des préfixes prend entre 100 msec et 30 secondes par préfixe.

Cette constatation permet d'expliquer pourquoi la convergence de BGP peut prendre quelques dizaines de minutes au niveau de l'Internet global [Ahu00] : une annonce de préfixe peut se voir ajouter un délai supplémentaire de 30 sec à chaque routeur qu'elle traverse. Si la suppression du MRAI fait partie des solutions proposées par [Ahu01] pour atteindre des convergences de l'ordre de la milliseconde, [PZW+02] ont montré que pour des valeurs faibles de MRAI la fréquence d'échange des UPDATE était élevée. Or, la raison d'être du MRAI est de limiter la quantité de données échangées, en permettant de grouper plusieurs préfixes dans une seule annonce, et de limiter les oscillations de routes, en ne conservant que la dernière route sélectionnée pour un préfixe pendant un intervalle de temps déterminé. Un bon compromis serait d'utiliser une valeur de MRAI qui minimise les échanges de messages sur le réseau en n'ayant pas un impact trop important sur la convergence du réseau. [PZW+02] ont montré, que pour certaines topologies de réseau, une valeur de MRAI de 7 sec permettait d'atteindre cet objectif.

7.5. Influence du Holdtimer

Le Holdtimer est un paramètre de configuration de BGP qui permet de fixer le délai d'attente maximum entre deux messages UPDATE et/ou KEEPALIVE reçus d'un voisin avant de considérer que ce voisin est inactif. La valeur du Holdtimer est négociée entre chaque couple de voisins lors de l'ouverture de la session BGP, et c'est la valeur la plus basse qui est choisie. Lorsque le Holdtimer vaut 0, la session est supposée être toujours active.

Généralement, la fréquence d'émission des KEEPALIVE est fixée à 1/3 de la valeur du Holdtimer. Nous avons voulu vérifier si l'émission ou la réception des messages KEEPALIVE pouvait influencer le temps de traitement des préfixes BGP, par exemple en augmentant ou en diminuant les temps de traitement moyens, ou en influençant la proportion des valeurs « anormales ». Le test porte, une fois de plus sur la redistribution de préfixes choisis sur base de la longueur de l'AS-PATH.

Le [RFC1771] recommande d'utiliser 90 sec comme valeur de Holdtime. Cisco, quant à lui, utilise 180 sec comme valeur par défaut [Par01]. Nous testerons en plus quelques valeurs qui dépassent la durée du test (1800 et 10800 sec), ainsi que la valeur de 0.

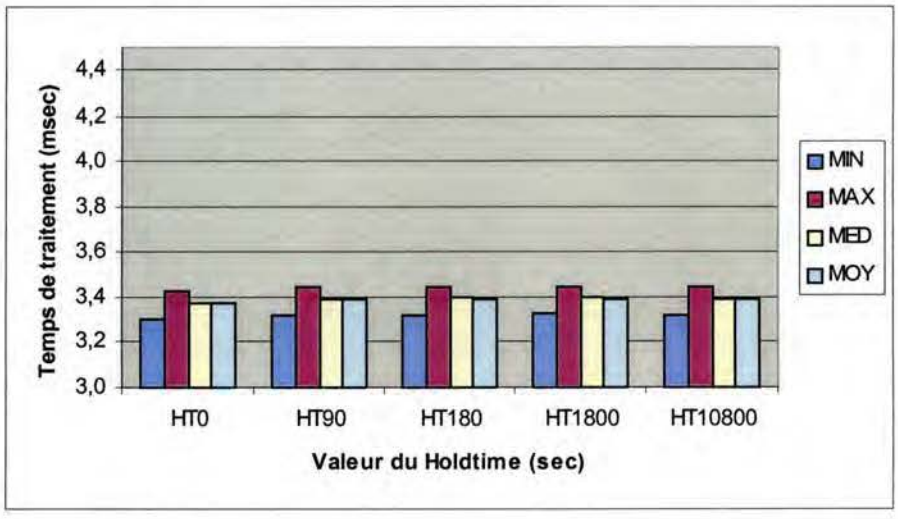

Figure 7-11 : Effet du Holdtimer sur les temps de traitement des UPDATE BGP

La figure 7-11 nous montre que la moyenne des temps de traitement mesurés tourne aux alentours de 3.39 msec et est à peu près identique quelle que soit la valeur de Holdtimer utilisée.

La figure 7-12 nous confirme que la répartition des temps de traitement pendant le déroulement de l'expérience est semblable quelle que soit la valeur utilisée pour le Holdtime. Lorsqu'on utilise un Holdtime de 180 sec, et seulement dans ce cas-là, on observe une série de valeurs de convergence de l'ordre de 4.3 msec, qui apparaissent avec un intervalle de 60 sec. Ces temps de traitement plus élevés pourraient être provoqués par la réception ou l'émission d'un message KEEPALIVE pendant le traitement du préfixe, mais cette hypothèse est difficile à confirmer vu qu'on n'observe pas un phénomène semblable dans le cas où le Holdtime est fixé à 90 sec. Pour les valeurs de Holdtime plus élevées, la durée des tests ne permet pas de mettre en évidence un effet semblable.

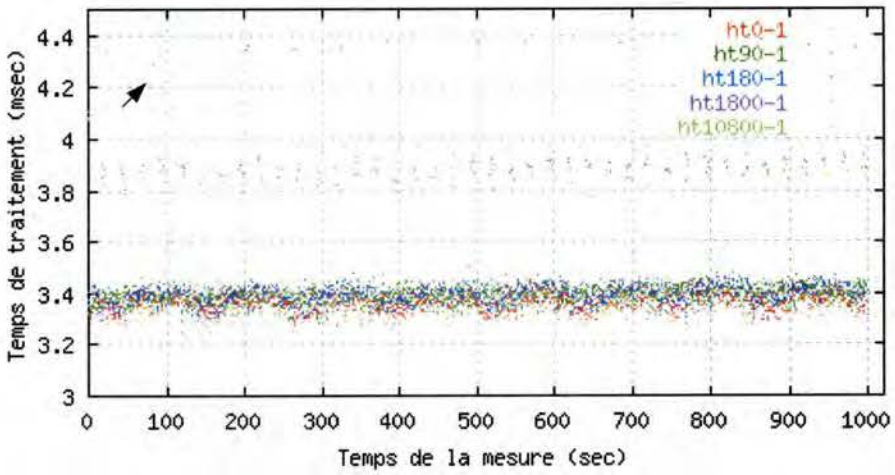

Figure 7-12 : Comparaison des temps de traitement mesurés au cours du déroulement de l'expérience pour différentes valeurs du Holdtimer

Quoi qu'il en soit, lorsque les messages sont envoyés à raison d'un par seconde, le fait d'avoir une valeur de convergence plus élevée toutes les 60 secondes n'influence pas fortement les calculs de moyenne. Nous avons choisi, pour la réalisation des tests une valeur de Holdtime de 10800, afin d'éviter l'émission de messages KEEPALIVE pendant la durée des tests. La valeur de 10800 est préféré à celle de 0, parce que, lors de la négociation de session, il est plus facile de diminuer cette valeur que de l'augmenter.

7.6. Influence de divers paramètres de configuration

L'objectif de cette série de tests est de mettre en évidence l'effet de divers paramètres de configuration sur le temps de convergence des préfixes. La plupart de ces paramètres sont utilisés dans la configuration de routeurs de production. Ils sont décrits dans [Par01, Cisco].

Nous avons testé les configurations suivantes :

`remove-private-as` : sert à enlever les numéros d'AS privés de l'AS-PATH. Lorsqu'un AS est connecté à un seul fournisseur, il est possible qu'il ne reçoive pas un numéro d'AS officiel, mais bien un numéro d'AS du pool privé (64512-65532). Dans ce cas, le fournisseur fait en sorte que les numéros d'AS privés ne soient pas annoncés dans l'Internet. Cette commande est utilisée uniquement vis-à-vis de pairs EBGP.

`maximum-prefix` : permet de limiter le nombre de préfixes appris d'un voisin. Lorsque le seuil est dépassé, le routeur envoie un message de NOTIFICATION à son voisin et ferme la session. L'option `threshold-value` permet de générer un message d'avertissement lorsqu'un certain pourcentage du seuil `maximum-prefix` est atteint : par défaut, ce seuil est fixé à 75%, et nous conserverons cette valeur. Afin de permettre à tous les messages testés d'être annoncés au pair en aval, nous utiliserons l'option `warning-only` pour nos tests (tous les messages sont acceptés et redistribués au pair en aval et un message d'avertissement est généré à la console).

`next-hop-self` : lorsqu'un routeur BGP apprend des routes d'un voisin EBGP, et qu'il les ré-annonce à un voisin IBGP, l'attribut NEXT-HOP reste inchangé. Dans certains cas il peut être intéressant que le routeur s'annonce lui-même comme NEXT-HOP, au moyen de la

commande `next-hop-self`. En effet, si le NEXT-HOP n'est pas directement connecté, un examen récursif de la table de routage est nécessaire pour trouver la route vers ce NEXT-HOP; cela implique que cette route doit également être injectée dans l'IGP. Sur les milieux à accès partagé (Ethernet ou ATM), on garde également comme NEXT-HOP l'adresse IP du premier routeur par lequel l'annonce est arrivée sur le réseau. Lorsque ce milieu n'est pas de type broadcast, la commande `next-hop-self` s'avère nécessaire pour permettre l'acheminement des paquet entre deux routeurs qui partagent un même sous-réseau, mais ne sont pas directement connectés.

`bgp bestpath med-missing-as-worst` : permet de ne pas donner la préférence aux préfixes pour lesquels le MED manque, en leur attribuant une valeur de MED élevée. Le comportement par défaut des routeurs BGP qui utilisent le logiciel Cisco IOS est de traiter les routes sans attribut MED comme ayant un MED de 0. Or, une décision récente de l'IETF prévoirait d'assigner une valeur infinie au MED manquant [Cisco]. Pour configurer les routeurs de façons à ce qu'ils se conforment au standard IETF, il est recommandé d'utiliser la commande `bgp bestpath med missing-as-worst`.

`send-community` : spécifie que les attributs COMMUNITY doivent être envoyés ou transmis à un voisin. Par défaut, l'attribut COMMUNITY n'est envoyé à aucun voisin.

`password` : permet l'authentification MD5 des pairs BGP sur une connexion TCP. Les messages sont envoyés en clair. Pour pouvoir utiliser la commande `password`, il est nécessaire de configurer les deux pairs de la session BGP.

`soft-reconfiguration inbound` : lorsqu'on modifie les politiques appliquées aux mises-à-jour de routage reçues d'un voisin, il est nécessaire de réinitialiser la session avec ce voisin pour que les nouvelles politiques soient prises en compte. La réinitialisation a un impact négatif sur le routage. L'option `soft-reconfiguration` permet de changer les politiques de routage sans devoir réinitialiser la session. Pour pouvoir réaliser la reconfiguration douce en entrée, le routeur BGP local doit stocker toutes les mises-à-jour de routes indépendamment des politiques utilisées ; les politiques de routage sont appliquées sur les mises-à-jour stockées. Cela implique une utilisation de mémoire plus importante.

`update-source` : la commande `update-source` est utilisée principalement pour permettre aux routeurs d'établir des sessions BGP indépendamment de l'état de leurs interfaces physiques. Cette commande est particulièrement utile dans le cas de sessions IBGP, puisqu'il existe souvent plusieurs chemins alternatifs pour joindre deux routeurs à l'intérieur d'un AS. Mais nous avons constaté qu'elle était aussi fréquemment utilisée par des ISP importants pour établir leurs sessions EBGP.

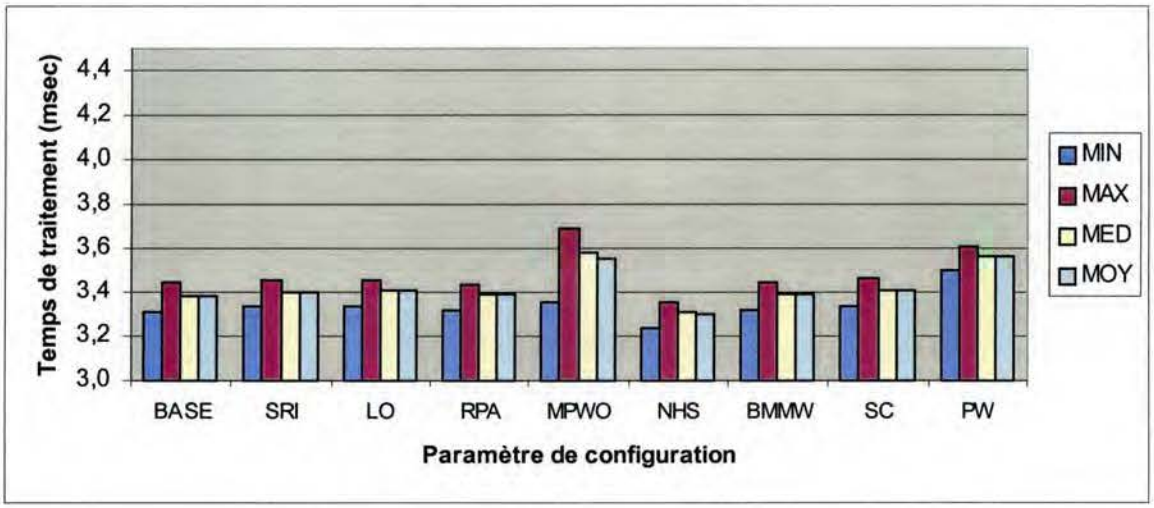

Figure 7-13 : Effet des différents paramètres de configuration sur les temps de traitement des UPDATE BGP

La figure 7-13 nous montre que la moyenne des temps de traitement obtenue pour les paramètres de configuration `soft-reconfiguration inbound` (SRI), `loopback` (LO,) `remove-private-as` (RPA), `bgp bestpath med-missing-as-worst` (BMMW) et `send-community` (SC) est à peu près égale à celle des tests de base (environ 3.39 msec). On peut considérer que ces configurations n'affectent pas la convergence des préfixes.

Par contre, la moyenne des temps de traitement obtenue pour la configuration `next-hop-self` (NHS) est inférieure à celle des tests de base. Elle se situe aux alentours de 3.30 msec, ce qui correspond à une diminution de 2.39%. En pratique, lorsqu'un routeur redistribue un préfixe à un voisin EBGP, il modifie la valeur du NEXT-HOP pour lui attribuer sa propre adresse, sauf dans le cas d'un milieu à accès multiple, lorsque le voisin en amont et en aval du DUT se trouvent sur le même réseau. La légère diminution (0.081 msec) des temps de traitement observée lorsque la commande `next-hop-self` est utilisée pourrait s'expliquer par le fait que le routeur n'a pas à déterminer à quel type de situation il a affaire avant de décider si oui ou non il change le valeur du NEXT-HOP de l'annonce. Nous attribuons donc cette valeur à la comparaison des réseaux vers le pair en amont et en aval du DUT.

La moyenne des temps de traitement mesurée avec la commande `password` (PW) se situe aux alentours de 3.56 msec. Nous faisons l'hypothèse que la différence par rapport aux tests de base peut être attribuée à l'opération de calcul de la valeur du hash qui permet d'authentifier les messages.

La moyenne des temps de traitement obtenue pour la configuration `maximum-prefix warning-only` (MPWO) est plus élevée que celle des tests de base. Elle se situe à 3.55 msec, ce qui correspond à une augmentation de plus de 5%. On constate que l'écart qui sépare la valeur minimale et la valeur maximale est plus important avec cette que dans les autres cas.

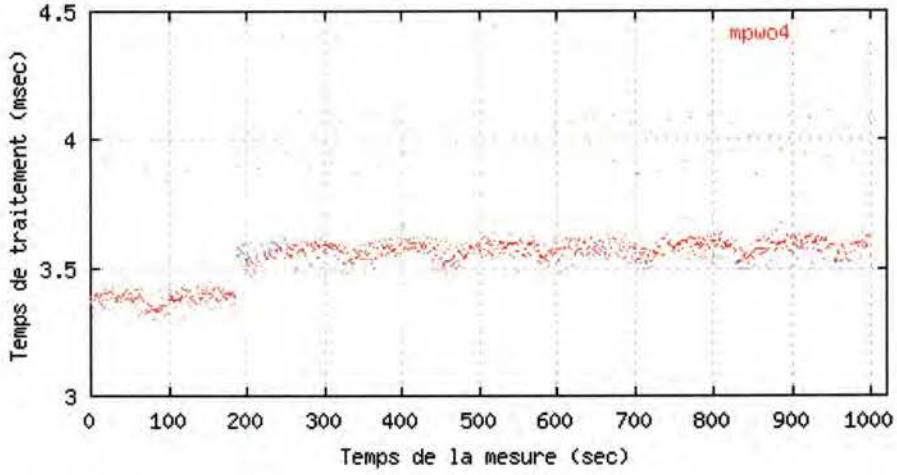

Figure 7-14 : Evolution des temps de traitement des UPDATE BGP obtenus avec la configuration maximum-prefix warning-only au cours du déroulement de l'expérience

La figure 7-14 nous montre que la distribution des temps de traitement obtenus avec la configuration `maximum-prefix warning-only` se répartit sur deux niveaux : ils sont de l'ordre de 3.39 msec pour les 187 premiers préfixes, soit la valeur des tests de référence, et de 3.58 msec pour tous les préfixes qui suivent. En se basant sur le fait que le nombre maximum de préfixes acceptés était fixé à 250, et que la commande utilise par défaut un seuil de 75% au-delà duquel un message d'avertissement est généré, on peut considérer que tant que le seuil n'est pas atteint, les temps de traitement ne sont pas influencés par la configuration. Par contre, ils sont augmentés de 0,197 msec dès que le seuil est dépassé. Nous attribuons cette différence à l'émission d'un message vers la console. L'objectif de cette commande est probablement de favoriser l'agrégation en limitant le nombre de préfixes annoncés par un voisin, et dans ce cas, la génération d'un message d'avertissement a probablement moins d'impact sur l'Internet global qu'une fermeture de session, qui se produit lorsque le seuil est dépassé et que l'option `warning-only` n'est pas utilisée.

7.7. Influence de la taille de la table de routage

La taille de la table de routage est un paramètre que l'on peut considérer comme un facteur externe au fonctionnement de BGP : elle varie en fonction du nombre de routes apprises dynamiquement par le routeur, et non pas sous l'effet du fonctionnement interne de BGP. Elle peut varier d'un nombre limité de préfixes dans les AS souches (stub), à plus de 120000 préfixes dans les routeurs du backbone [Potaroo]. Cette estimation ne tient compte que de la meilleure route pour un préfixe, choisie et annoncée aux autres pairs, et pas des multiples instances du préfixe qu'un routeur doit maintenir localement, parce qu'il a appris l'existence d'un même préfixe par plusieurs voisins différents.

D'après le [RFC1771] la fonction qui calcule le degré de préférence d'une route ne devrait pas être influencée par l'existence ou la non-existence d'autres routes. Nous avons donc voulu vérifier si dans l'implémentation de Cisco, la charge de la table de routage pouvait influencer la redistribution des préfixes. Pour ce faire, nous avons initialisé le DUT avec un nombre de préfixes croissant par puissances de 10 entre 0 et 10000[17].

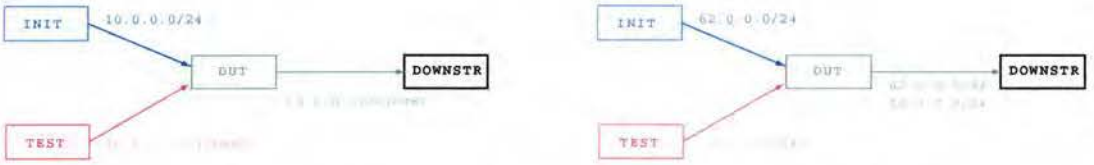

Figure 7-15 : Représentation des conditions d'initialisations testées

La figure 7-15 schématise les conditions d'initialisation utilisées pour mesurer l'influence de la taille de la table de routage sur les temps de traitement des UPDATE BGP. Dans le premier cas (g.), les préfixes qui servent à initialiser le DUT (INIT) appartiennent au même espace d'adressage que les préfixes TEST (n préfixes de longueur 24 constituant des sous-ensembles du préfixe 10/8). Lorsque le même préfixe a été annoncé par chacun des deux voisins, le DUT sélectionne celui qui a les meilleurs attributs de chemin. Dans le deuxième cas (d.), les préfixes qui servent à initialiser le DUT (INIT) appartiennent à un espace d'adressage différent de celui des préfixes TEST (n préfixes de longueur 24 constituant des sous-ensembles du préfixe 62/8). Le DUT sélectionne tous les préfixes reçus de chacun des deux voisins.

7.7.1. Initialisation du DUT avec un sous-ensemble des préfixes testés

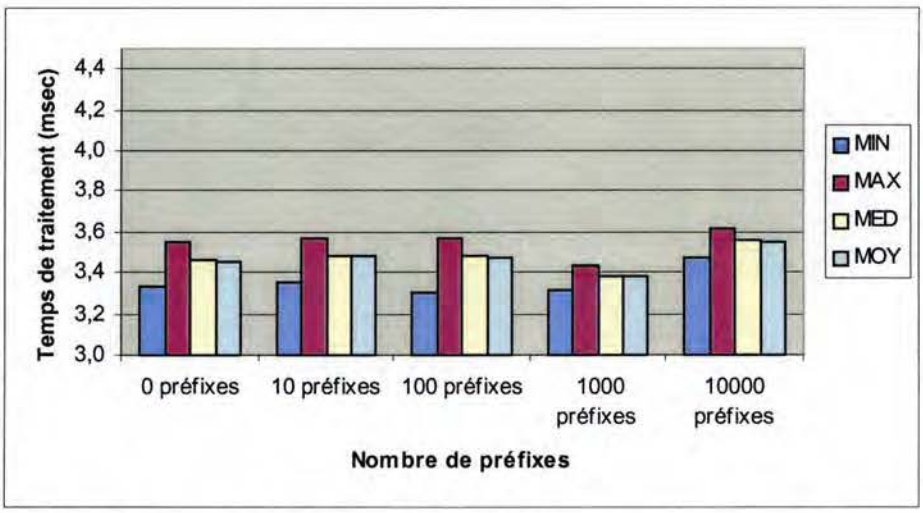

Figure 7-16 : Effet de la taille initiale de la table de routage du DUT sur les temps de traitement

[17] La RAM des routeurs Cisco mis à la disposition du laboratoire ne permettait pas de prendre en charge 100000 préfixes.

La figure 7-16 nous montre que le temps de traitement moyen le plus bas est obtenu lorsque la table de routage du DUT est initialisée avec 1000 préfixes (3.38 msec). Les valeurs sont légèrement plus élevées lorsque la table de routage du DUT est initialisée avec 0, 10 et 100 préfixes (environ 0.08 msec), mais dans ces cas, l'écart entre la valeur minimum et maximum est plus important. La valeur la plus élevée est observée lorsque la table de routage est initialisée avec 10000 préfixes (environ 3.55 msec), ce qui représente une augmentation d'environ 5% par rapport à une table de routage initialisée avec 1000 préfixes.

L'analyse de la distribution des temps de traitement au cours du déroulement de l'expérience nous permet de grouper les mesures en deux catégories.

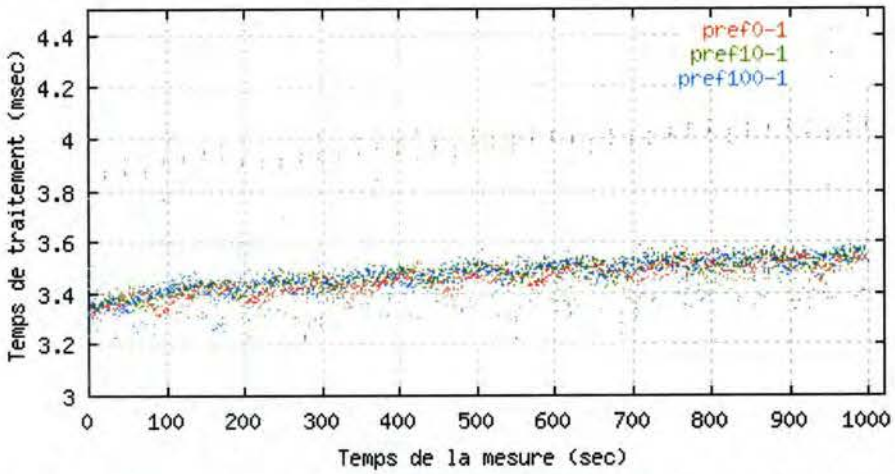

Figure 7-17 : Distribution des temps de traitement lorsque la table de routage du DUT est initialisée avec 0, 10 ou 100 préfixes

La figure 7-17 nous montre que lorsqu'on initialise la table de routage du DUT avec 0, 10 ou 100 préfixes qui appartiennent au même espace d'adressage que les préfixes testés les distribution des temps de traitement obtenues sont presque identiques. Quel que soit le nombre de préfixes utilisés pour initialiser la table de routage du DUT, les temps de traitement augmentent de manière semblable au cours du déroulement de l'expérience. On notera cependant que lorsque la table de routage du DUT est initialisée avec 100 préfixes, une partie des temps de traitement mesurés sont situés en dessous de la courbe principale.

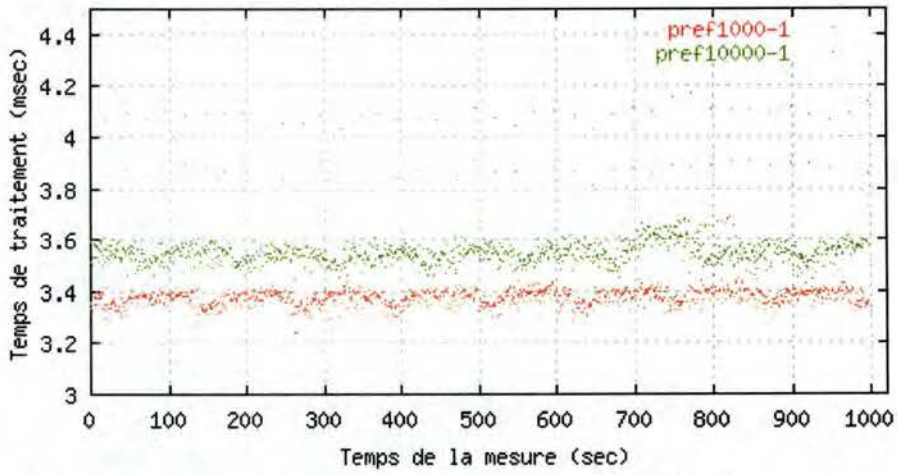

Figure 7-18 : Distribution des temps de traitement obtenus lorsque la table de routage du DUT est initialisée avec 1000 ou 10000 préfixes

Lorsqu'on initialise la table de routage du DUT avec 1000 ou 10000 préfixes, les temps de traitement restent stables au cours du déroulement de l'expérience (figure 7-18). Dans ces conditions, le niveau moyen des temps de traitement augmente en fonction du nombre de préfixes contenus dans la table de routage du DUT.

Ce qui différencie les deux groupes de résultats est le fait que lorsque la table de routage du DUT est initialisée avec moins de 1000 préfixes, une partie des préfixes testés sont choisis par le processus de décision parce qu'il n'y a pas d'autre route avec laquelle ils peuvent être comparés. Lorsque la table de routage du DUT est initialisée avec plus de 1000 préfixes, le DUT connaît une route pour chacun des préfixes avant de commencer l'expérience. Tous les préfixes reçus sont donc traités de la même manière.

7.7.2. Initialisation du DUT avec des préfixes différents des préfixes testés

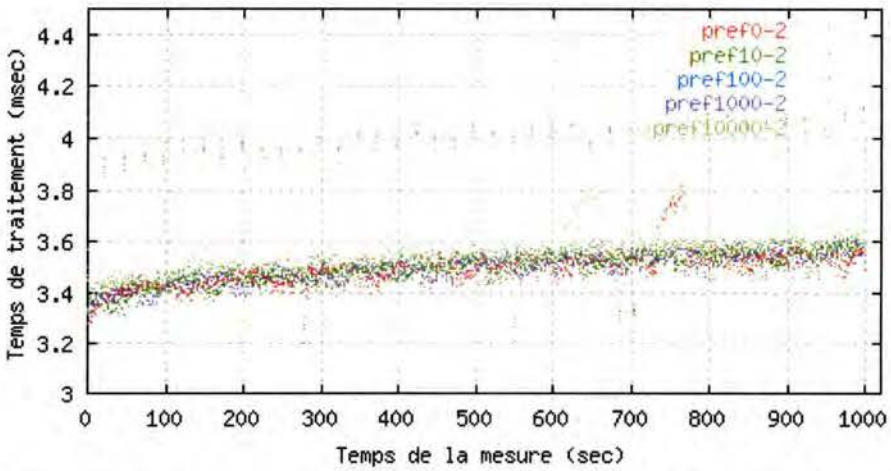

Figure 7-19 : Distribution des temps de traitement lorsque la table de routage du DUT est initialisée avec des préfixes différents des préfixes testés

La figure 7-19 nous montre que lorsque le DUT est initialisé avec des préfixes qui appartiennent à un autre espace d'adressage que les préfixes testés, les temps de traitement augmentent au cours du déroulement de l'expérience. La croissance des temps de traitement semble se ralentir au fur et à mesure du remplissage de la table de routage par les préfixes testés. La valeur du plateau semble se fixer aux alentours de 3.55 msec, ce qui correspond aux temps de traitement obtenus lorsque le DUT est initialisé avec 10000 préfixes qui appartiennent au même espace d'adressage que les préfixes testés (figure 7-18). La distribution des temps de traitement est la même quel que soit le nombre de préfixes utilisés pour initialiser la table de routage du DUT, et est comparable à celle obtenue lorsque le DUT est initialisé avec un très petit nombre de préfixes, si ces préfixes appartiennent au même espace d'adressage que les préfixes testés.

Nous déduisons de ces observations que pour pouvoir montrer un effet de la taille de la table de routage sur le temps de traitement des UPDATE BGP, il est nécessaire que la table de routage du DUT contienne déjà une instance des préfixes testés. Les conditions de déroulement du test peuvent donc influencer les conclusions.

Lorsque la table de routage du DUT contient déjà une instance du préfixe testé, le temps de traitement des UPDATE est stable au cours du déroulement de l'expérience et augmente en fonction de la taille initiale de la table de routage. Par contre, lorsque la table de routage ne contient pas encore d'instance des préfixes traités (ou une faible proportion), les temps de traitement des UPDATE semblent indépendants de la taille de la table de routage. De plus, dans ces conditions, la distribution des temps de traitement évolue au cours du temps mais atteint systématiquement un plateau après un certain temps. Il est possible que ce plateau corresponde à un seuil au-delà duquel les temps de traitement sont indépendants de la taille de la table de routage, quelles que soient les conditions de réalisation du test. Pour confirmer cette hypothèse, il suffirait de mesurer les temps de traitement lorsque la table de routage du DUT est initialisée avec 20000 préfixes, dans les deux situations.

Nos résultats expérimentaux ne nous permettent pas de tirer des conclusions définitives à propos de l'impact éventuel de la taille de la table de routage, surtout dans une situation réelle, où le contenu de la table de routage serait plus hétérogène, où la table de routage du routeur contiendrait un nombre variable d'instances de chaque préfixe testé (0, 1 ou plusieurs). Par contre, ils nous donnent la conviction que le dispositif de test que nous avons utilisé, dans lequel la table de routage du DUT contient préalablement une instance (moins bonne) de chaque préfixe testé, constitue une situation idéale pour mettre en évidence l'impact d'autres facteurs : dans ces conditions, les temps de traitement des préfixes sont stables au cours du temps.

7.8. Discussion

Au cours de ce chapitre, nous avons essayé de caractériser le fonctionnement interne d'un routeur BGP commercial au moyen de tests black-box, qui calculent le temps de traitement des UPDATE sur base de leur temps de passage à un point de contrôle situé avant l'entrée du routeur et après sa sortie. Une des difficultés de ce type de test consiste à trouver les événements ou configurations qui permettent de baliser le plus précisément possible les processus internes.

Les résultats de nos tests nous ont montré que les temps de traitement des UPDATE BGP sont suffisamment stables et reproductibles lorsque les tests sont réalisés dans de mêmes conditions pour pouvoir établir un lien de cause à effet entre un changement de configuration et une variation du temps de traitement.

L'étude du processus de décision de routes nous a montré que, globalement, les temps de traitement étaient semblables lorsque la sélection de routes était basée sur le WEIGHT, l'AS-PATH, l'ORIGIN et le RID. Par contre, le temps de traitement mesuré pour une sélection basée sur le MED est plus élevé. Cela peut paraître surprenant, dans l'hypothèse d'un examen séquentiel des différents critères du processus de décision. Mais cette différence peut s'expliquer par le fait qu'en cas d'inégalité du MED, il faut également que les UPDATE ont été émis par le même AS.

L'étude du fonctionnement du processus de décision réalisée dans d'autres conditions de redistribution (vers un voisin IBGP, retrait de préfixes, ...) confirme nos observations. Nous constatons en plus que le temps de traitement est plus faible lorsque le DUT redistribue des annonces à un voisin IBGP. Cette différence pourrait s'expliquer par le fait que l'AS-PATH ne doit pas être modifié en cas d'annonce à un voisin IBGP. Finalement, nous constatons que les temps de traitement sont plus bas dans les cas de retraits de routes. Et nous avons attribué cette différence au fait qu'il n'est pas nécessaire de calculer un degré de préférence dans le cas de retrait de routes.

La durée des processus de filtrage de routes et de manipulation d'attributs varie en fonction du nombre de critères à examiner, aussi bien à l'entrée qu'à la sortie du routeur. A l'entrée du routeur, le temps de traitement obtenu pour la modification d'attribut est supérieur à celui obtenu pour le filtrage de route. Nous avons attribué cette différence à l'application de la clause `set`. A la sortie du routeur, le temps de traitement obtenu pour la manipulation d'attribut est équivalent à celui obtenu pour le filtrage de route. Nous en avons déduit que l'application de la clause `set` était transparente à la sortie du routeur. Le filtrage de route prend plus de temps lorsqu'il est appliqué à la sortie du routeur qu'à l'entrée : nous en avons déduit que les machines d'application de politiques de routage sont différentes à l'entrée et à la sortie du routeur. Enfin, nous avons estimé la durée élémentaire de chaque examen d'une clause `match`, dans le cas d'un `route-map` appliqué à l'entrée ou à la sortie du routeur.

Nous avons ensuite comparé différentes techniques de filtrage de routes. D'une manière générale, les `route-map` sont moins efficaces que les filtres classiques (`filter-list`, `distribute-list` et `prefix-list`) ; de plus le filtrage basé sur l'AS-PATH est moins efficace que le filtrage basé sur le NLRI. Nos observations nous permettent de confirmer le bien-fondé de la sagesse populaire, qui recommande d'utiliser les `filter-list` pour un filtrage basé sur l'AS-PATH, les `prefix-list` pour un filtrage basé sur le NLRI et réserver les `route-map` pour le filtrage basé sur l'attribut COMMUNITY ou pour la manipulation d'attributs, malgré l'extrême flexibilité de cet outil.

L'agrégation de routes a pour objectif de diminuer les informations de routage, tout en maintenant un même degré de connectivité. Elle se produit après la sélection du meilleur chemin par le processus de décision. Généralement, seuls les préfixes agrégés sont accessibles, mais dans certaines circonstances, il est souhaitable d'annoncer également les préfixes spécifiques, par exemple pour mieux répartir le trafic à l'intérieur de l'AS qui a procédé à l'agrégation. La commande `aggregate-address` a un impact important sur les temps de traitement des UPDATE portant des préfixes spécifiques. Cet impact est d'autant plus important que la distance qui sépare le préfixe spécifique de l'agrégat est grande. Nous avons également mis en évidence l'impact de l'utilisation de l'option `as-set` sur le temps de traitement, et le fait que la redistribution de l'agrégat et du préfixe spécifique n'était jamais simultanée, et que le délai entre ces événements était très élevé.

Après avoir étudié le fonctionnement interne du processus de décision, nous avons étudié l'impact potentiel de facteurs externes au processus de décision sur le comportement du routeur. Le MRAI (MinRouteAdvertisementInterval) est utilisé pour temporiser l'émission de nouveaux UPDATE, afin de permettre le regroupement de préfixes qui partagent les mêmes attributs et d'éviter d'annoncer à de multiples reprises un préfixe qui aurait été sélectionné plusieurs fois dans un court laps de temps. En principe, le MRAI permet à un routeur d'avoir eu l'occasion d'apprendre la sélection de routes de tous ses pairs avant d'annoncer sa propre sélection de préfixes. Le MRAI a un impact important sur la convergence globale du protocole BGP, mais il est nécessaire pour limiter le taux d'échange des UPDATE. Nos résultats expérimentaux nous ont permis de confirmer que le MRAI imposait bien une limite supérieure au temps de traitement des UPDATE, et que pour les routeurs Cisco, il était implémenté par pair, et non pas sur une base « par pair et par préfixe ».

Le Holdtimer permet de fixer le délai d'attente maximum entre deux messages reçus d'un voisin avant de considérer que ce voisin est inactif. Nous n'avons pas pu montrer que ce timer avait un effet important sur les temps de traitement moyen des UPDATE BGP. Nous ne pouvons donc pas expliquer pourquoi la configuration par défaut des routeurs Cisco utilise une valeur de Holdtime supérieure à celle qui est recommandée par le [RFC1771], alors qu'une valeur plus basse est sensée accélérer la détection des défaillances du réseau.

Parmi les paramètres de configuration utilisés pour influencer le comportement de base de BGP, peu ont un impact sur les temps de traitement, et lorsqu'un effet est mis en évidence, il est de faible amplitude. La commande `next-hop-self` permet de diminuer légèrement le temps de traitement, probablement parce qu'elle permet de modifier l'attribut NEXT-HOP de manière inconditionnelle. La commande `maximum-prefix` n'affecte pas les temps de traitement tant qu'on n'a pas dépassé un certain seuil, qui dépend de la limite au-delà de laquelle un message d'avertissement est généré à l'intention de l'administrateur de réseau. Au-delà de cette limite, les temps de traitement augmentent légèrement, probablement à cause de la génération du message. Enfin, la commande `password` augmente également les temps de traitement. Cette commande est néanmoins utile afin de permettre l'authentification de la source des messages.

Finalement, nous avons montré que l'impact de la taille de la table de routage sur le temps de traitement des UPDATE dépendait des conditions de réalisation du test. Néanmoins, le dispositif de test que nous avons utilisé, dans lequel la table de routage du DUT contient préalablement une instance (moins bonne) de chaque préfixe testé, constitue une situation idéale pour mettre en évidence l'impact d'autres facteurs.

En conclusion, nos tests nous ont permis de mettre en évidence l'importance relative des différents processus internes du routeur BGP tels que la sélection de routes, l'application de politiques à l'entrée ou à la sortie du routeur, l'agrégation de préfixes dans les temps de traitement des UPDATE BGP. Pour ce faire, nous avons établi une relation entre la variation du temps de traitement des UPDATE induite par certaines situations de test et le processus interne potentiel que cette situation était censée influencer. Sur base de ces constats, nous avons établi une liste de différents processus internes et nous leur avons attribué une durée (voir annexe K). Cette liste est incomplète : elle ne concerne que les processus pour lesquels nous avons pu trouver un événement discriminant et pour lesquels l'effet était supérieur à l'écart-type. En conclusion, nos tests nous ont permis de brosser un portrait assez général du fonctionnement interne du routeur BGP.

8. Tests avec plusieurs pairs en aval du DUT

La commande `peer-group` est généralement utilisée dans des situations où un routeur dispose de plusieurs voisins et qu'il leur applique des politiques de mise-à-jour identiques. Elle permet au routeur de calculer la meilleure route vers une destination une seule fois puis de l'envoyer à tous les membres du groupe, au lieu d'effectuer le calcul pour chacun des pairs séparément. Pour pouvoir évaluer l'effet de cette commande, il est nécessaire que le DUT dispose de plusieurs voisins.

La commande `peer-group` peut s'appliquer à des pairs internes ou externes. C'est ce dernier cas que nous étudierons. La configuration d'un groupe de pairs externes (external peer-group) doit répondre à un certain nombre d'exigences. D'une part, tous les voisins appartenant au groupe doivent avoir des adresses IP qui appartiennent au même sous-réseau. D'autre part, ils doivent appartenir à des systèmes autonomes différents. Enfin, il faut éviter que le routeur sur lequel est configuré le peer-group serve de relais pour les mises-à-jour de routage entre les membres du groupe : si une annonce est reçue d'un des membres du groupe, elle ne doit pas être retransmise aux autres. Nous tiendrons compte de ces considérations au moment de configurer les voisins du DUT, que la commande `peer-group` soit utilisée ou non.

Nous avons décidé de tester un peer-group qui ne s'applique qu'aux mises-à-jour qui sortent du DUT, et donc de travailler avec plusieurs pairs en aval du DUT : ce scénario permet de mettre en évidence des différences de comportement entre les membres du groupe, puisque avec un seul signal déclencheur (le message qui entre dans le DUT), on peut avoir une ou plusieurs réponses, selon le nombre de pairs en aval du DUT.

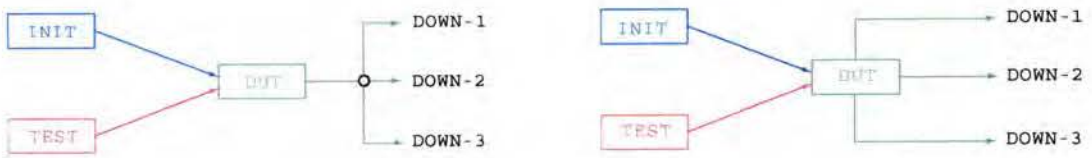

Figure 8-1 : Représentation de l'utilisation d'une (g.) ou plusieurs (d.) interfaces de sortie pour l'envoi des messages aux voisins en aval du DUT

La figure 8-1 représente la manière dont le DUT peut établir des sessions BGP avec ses pairs en aval : soit il utilise une même interface de sortie pour toutes les sessions (figure 8-1, g.), soit il utilise une interface de sortie différente pour chaque session (figure 8-1, d.). Dans la pratique, il est peu probable que le DUT utilise des interfaces différentes (et des liens physiques différents) pour établir une session vers des pairs qui partagent le même sous-réseau (comme l'impose l'utilisation de la commande `peer-group`) : cela nécessiterait quelques artifices supplémentaires pour garantir la connectivité des différents voisins entre eux. Les tests seront donc réalisés prioritairement en utilisant la première configuration.

Notre dispositif de test doit nous permettre de répondre aux questions suivantes :

- le nombre de voisins influence-t-il les temps de traitement des préfixes ?
- est-il intéressant d'utiliser la commande `peer-group` si aucune politique particulière n'est appliquée aux voisins ?
- la commande `peer-group` influence-t-elle les temps de traitement en cas d'agrégation de route ?
- la commande `peer-group` influence-t-elle les temps de traitement lorsque des politiques de filtrage de route ou de manipulation d'attributs sont utilisées ?

8.1. Influence du nombre de pairs

Dans les situations où un routeur dispose de plusieurs voisins, il est intéressant de savoir si les temps de traitement des préfixes dépendent du nombre de voisins d'une part, et s'ils sont identiques pour toutes les sessions.

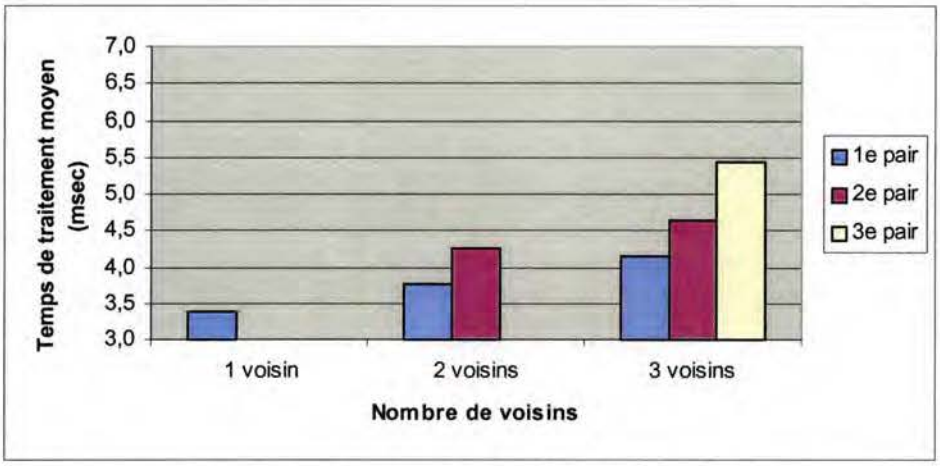

Figure 8-2 : Influence du nombre de pairs en aval du DUT sur les temps de traitement moyens de chacun des pairs

La figure 8-2 nous montre que le nombre de voisins en aval du DUT influence fortement les temps de traitement des préfixes. On remarque que plus le nombre de pairs augmente, plus les temps de traitement moyens des messages envoyés au premier pair[18] en aval du DUT augmente. Ainsi, les temps de traitement calculés pour le premier pair en aval du DUT sont respectivement de 3.39, 3.76, et 4.15 msec. La différence mesurée entre la situation avec trois pairs en aval du DUT et celle avec un seul correspond à une augmentation d'environ 22% des temps de traitement. De plus, si on tient compte d'une éventuelle variabilité expérimentale, on peut déduire que le temps de traitement augmente de 0.38 msec à chaque fois qu'on ajoute un nouveau voisin à la configuration. Un test complémentaire avec 4 pairs en aval du DUT nous aurait permis de vérifier cette hypothèse.

On constate également que les temps de traitement des préfixes ne sont pas identiques pour tous les voisins en aval du DUT. Par exemple, dans la situation où on a trois pairs, les temps de traitement moyens des préfixes vers chacun des pairs sont respectivement de 4.15, 4.65 et 5.43 msec. Si on examine l'écart entre le deuxième et le premier pair, on constate qu'il est de 0.492 msec. C'est également le cas pour la configuration avec deux pairs en aval du DUT. L'écart entre le troisième et le premier pair est quant à lui de 1.272 msec : il a tendance à augmenter de manière exponentielle.

Sur base de nos mesures, on peut donc supposer qu'il y a moyen de prédire le temps de traitement du $k^{ième}$ pair dans une configuration avec n routeurs (k<=n) au moyen d'une formule mathématique, mais nos résultats ne nous permettent pas généraliser au-delà de 3 routeurs. En l'état actuel, la formule pourrait s'établir comme suit :

$$Temps_{(k,\,n)} = Temps_{base} + (0.38*(n-1)) + coeff_k$$

Avec
k : routeur vers lequel on veut estimer le temps de traitement
n : nombre de routeurs en aval dans la configuration (le nombre de routeurs en amont est supposé constant)
$coeff_1 = 0$; $coeff_2 = 0.492$; $coeff_3 = 1,272$.

[18] Les pairs sont triés par ordre croissant d'adresse IP. Les temps de traitement vers le pair dont l'adresse IP est la plus basse sont toujours les plus faibles.

Des résultats semblables ont été obtenus lorsque le DUT utilise des interfaces de sortie différentes pour chacun des voisins (les temps de traitement sont néanmoins légèrement plus élevés dans cette situation).

Cette situation est susceptible d'amener des informations incorrectes aux routeurs situés en aval du DUT. Imaginons que les annonces qui passent par le meilleur chemin entre deux routeurs soient systématiquement annoncées vers le troisième pair en aval à chacune des étapes intermédiaires. Parallèlement, imaginons que toutes les annonces qui passent par un chemin moins bon entre ces deux mêmes pairs soient systématiquement annoncées sur le premier pair à chacune des étapes intermédiaires. Avec de telles différences de temps de traitement, les annonces qui passent par la moins bonne route pourraient arriver avant celles qui passent par la meilleure, avec comme conséquence la nécessité de procéder au retrait des premières annonces, et donc des oscillations de routes jusqu'à ce que tous les routeurs aient pris connaissance d'une route stable. En pratique, l'utilisation du MRAI est censé limiter les oscillations de ce type.

8.2. Effet de la commande `peer-group`

8.2.1. Sur les tests de base

Cette série de tests va nous permettre de vérifier s'il est intéressant d'utiliser la commande `peer-group` dans le cas où aucune politique particulière n'est appliquée à la sortie du routeur.

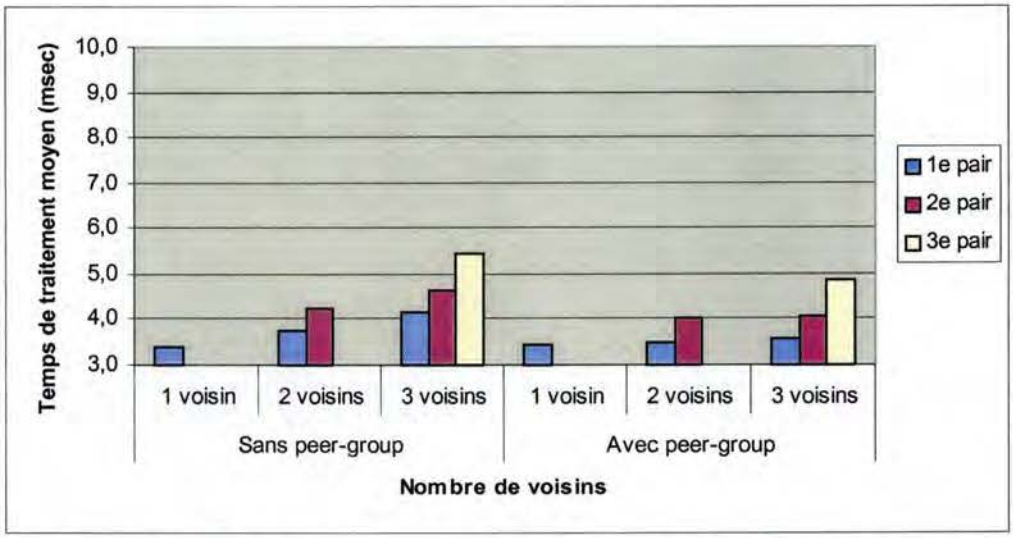

Figure 8-3 : Influence de la commande `peer-group` en fonction du nombre de pairs

La figure 8-3 nous montre que dans des situations avec plusieurs pairs en aval du DUT, la commande `peer-group` permet de diminuer le temps de traitement des préfixes pour chacun des pairs. La diminution des temps de traitement semble être la même pour tous les voisins du DUT. Par exemple, lorsque le DUT dispose de deux voisins en aval, les temps de convergence mesurés pour le premier et le deuxième voisin sont respectivement de 3.76 et 4.26 msec lorsque la commande `peer-group` n'est pas utilisée, et de 3.50 et 4.00 msec lorsque la commande `peer-group` est utilisée, ce qui correspond à une diminution de l'ordre de 0.26 msec. De plus, la diminution semble être plus importante lorsque le nombre de pairs en aval du DUT augmente. Par exemple, la diminution des temps de traitement provoquée par l'utilisation de la commande `peer-group` est de l'ordre de 0.59 msec avec trois pairs en aval. Par contre, avec un seul pair en aval du DUT, on constate que le temps de traitement semble augmenter légèrement (0.032 msec), mais cette augmentation est du même ordre de grandeur que l'écart-type habituel : nous la considérons donc comme négligeable.

La conséquence de l'utilisation de la commande `peer-group` est que, pour une même position du pair en aval, les temps de traitement sont ramenés à une valeur à peu près équivalente, quel que soit le nombre de voisins. Par exemple, les temps de traitement

vers le premier pair en aval du DUT sont de 3.43, 3.50 et 3.57 msec suivant que le DUT a 1, 2 ou 3 voisin. La commande `peer-group` n'affecte cependant pas la différence de temps de traitement due à la position du voisin dans la configuration.

8.2.2. Lors de l'application de politiques de routage

L'effet de la commande `peer-group` en présence de différentes politiques de routage a été étudié dans le cas où le DUT possède trois voisins en aval. Les tests ont été réalisés dans les cas où aucune politique particulière n'est appliquée à la sortie du routeur (REF), mais également dans les cas où un `route-map` est utilisé à la sortie du routeur modifier un attribut (RM-OUT[19]), et enfin lorsque la commande `aggregate-address` est utilisée, avec deux longueurs de masque (21 et 16 bits respectivement).

En ce qui concerne la commande `aggregate-address`, les tests ont été réalisés avec les différentes options de la commande (***AM, AMAS, ATTR, ADVER***), mais les résultats étant généralement similaires, nous n'avons représenté que ceux obtenus avec l'option ***AM*** pour simplifier les commentaires.

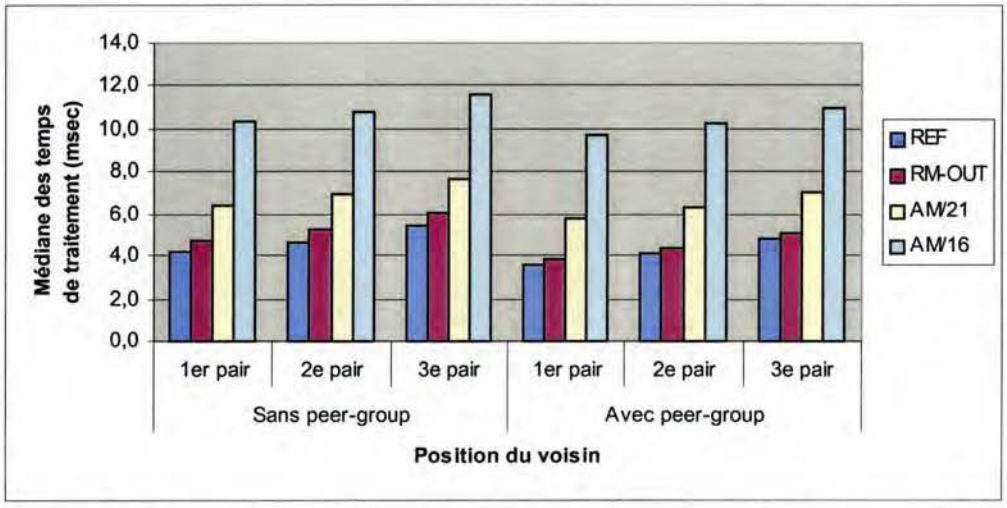

Figure 8-4 : Effet de la commande `peer-group` sur la médiane des temps de traitement des UPDATE BGP envoyés à chacun des voisins du DUT sur les tests de base, en présence d'un `route-map` ou dans le cas de l'agrégation de préfixes.

La figure 8-4 (g.) nous montre que, lorsque le DUT a établi une session avec plusieurs pairs en aval, les temps de traitement des préfixes varient en fonction de la position du pair, que l'on applique une politique à la sortie du routeur ou non. Nous avons comparé les valeurs expérimentales avec les valeurs obtenues lorsqu'on additionne le temps de traitement d'un test de référence (un seul pair, sélection basée sur l'AS-PATH), avec la durée de traitement spécifique à la commande (déterminée aux sections 7.2 pour les `route-map` et 7.3 pour la commande `aggregate-address`), et le supplément imputable à la configuration calculé sur base de la formule 8.1. Généralement, les résultats sont comparables. Sauf dans le cas du `route-map` : la valeur calculée est inférieure d'environ 0.39 msec à la valeur expérimentale, quel que soit le pair. Une explication possible à cette différence est le fait que dans le section 7.2 nous avons modifié l'attribut COMMUNITY, alors qu'ici, nous modifions l'attribut MED. Une autre serait due au fait que contrairement à la commande `aggregate-address`, qui s'applique globalement à tous les pairs, le `route-map` s'applique pour chacun des pairs. Si cette hypothèse est correcte, il faudrait encore pouvoir vérifier si le facteur correctif dépend du nombre de pairs dans la configuration, et s'il est le même pour toutes les politiques spécifiques que l'on peut appliquer à la sortie du routeur (`filter-list`, `distribute-list` ...).

La figure 8-4 (d.) nous montre que, lorsque le DUT a établi une session BGP avec plusieurs pairs en aval, la commande `peer-group` permet de diminuer le temps de

[19] Une valeur de MED est assignée à toutes les routes dont l'AS-PATH répond à un certain critère

traitement des préfixes redistribués à chacun des pairs, qu'une politique (route-map, agrégation de préfixes) soit appliquée ou non à la sortie du routeur. Lorsqu'une même politique est appliquée à la sortie du routeur, la diminution des temps de traitement provoquée par l'utilisation de la commande peer-group est la même pour tous les pairs. La diminution est respectivement de 0.62 msec lorsque aucune politique particulière n'est appliquée (REF), 0.93 msec en présence d'un route-map (RM-OUT), 0.62 msec lorsque la commande aggregate-address est utilisée pour former des agrégats avec un masque de 21 bits (AM/21) et 0.59 msec lorsque la commande aggregate-address est utilisée pour former des agrégats avec un masque de 16 bits (AM/16). L'effet le plus important est observé en présence de route-map, et assez curieusement, cela compense presque le surplus expérimental que nous avions mentionné en l'absence de peer-group.

Nous proposons donc, mais en émettant une réserve, l'hypothèse que, lorsqu'une politique globale est appliquée (aggregate-address) ou en l'absence de politique de routage à la sortie du routeur, la commande peer-group permet de diminuer les temps de traitement de 0.26 msec pour un routeur avec deux pairs, et 0.59 msec avec trois pairs en aval, conformément à ce que nous avions relevé à la section 8.2.1. Par contre, lorsqu'une politique spécifique est appliquée (route-map), la commande peer-group permet de diminuer les temps de traitement de 0.93 msec dans une configuration avec trois pairs en aval.

Finalement, nous avons répété la même expérience lorsque le routeur utilise une interface différente pour chacune des session avec ses pairs en aval (figure 8-1, d.). Globalement, les résultats sont semblables à ce qui vient d'être décrit. Les temps de convergence sont toutefois légèrement plus élevés (d'environ 0.05 msec) lorsqu'on utilise une interface différente pour chaque session.

8.3. Discussion

Nos résultats expérimentaux nous montrent que lorsque le nombre de pairs dans une configuration augmente, le temps de traitement des préfixes envoyés au premier pair augmente, et que cette augmentation semble proportionnelle au nombre de pairs. De plus, lorsqu'un routeur a plusieurs voisins, le temps de traitement des préfixes vers chacun des voisins est différent. Nous avons constaté que pour chaque voisin dans une configuration, l'écart de temps de traitement par rapport au premier pair dépend de la position du voisin et non pas du nombre de pairs dans la configuration. Sur base de ces observations, nous avons établi une formule qui permet de calculer les temps de traitement vers chaque pair pour des configurations de maximum trois pairs.

L'augmentation des temps de traitement en fonction du nombre de pairs dans la configuration et de la position du pair est également observée lorsque des politiques telles que l'agrégation de routes ou la manipulation d'attributs sont utilisées. Les effets sont additifs dans le cas de l'agrégation de routes, et également dans le cas du route-map, si on tient compte d'un facteur correctif.

La commande peer-group permet de diminuer les temps de traitement des préfixes qu'une politique soit utilisée à la sortie du routeur ou non. L'effet de la commande peer-group est le même pour tous les voisins du routeur. Il dépend du nombre de voisins (il augmente lorsque le nombre de voisins augmente) et de la commande utilisée (il est plus marqué dans le cas du route-map que dans le cas de la commande aggregate-address ou en l'absence de politique de routage).

Conclusion

La première partie de notre mémoire a consisté à mettre au point un dispositif qui permet de calculer la re-convergence incrémentale de BGP au moyen de tests black-box, c'est-à-dire sur base du temps de passage des UPDATE BGP à un point de contrôle situé avant l'entrée dans le routeur étudié et après sa sortie.

Ce dispositif comporte quatre volets. Le premier consiste à envoyer des messages dont on contrôle les caractéristiques, afin de distinguer les messages, d'influencer le processus de décision et d'étudier des fonctionnalités particulières du routeur BGP. Le second consiste à utiliser une configuration appropriée du DUT pour permettre aux messages d'être acceptés, sélectionnés et ré-annoncés, et pour étudier différentes propriétés de BGP. Le troisième consiste à établir les conditions nécessaires pour calculer les temps de traitement des UPDATE, et à déployer le dispositif de mesure. Le dernier consiste à établir les relations entre les différents intervenants, et à planifier les différentes étapes de chaque série de mesure.

Si ce dispositif nous a permis de mettre en évidence certaines propriétés du fonctionnement de BGP, nous déplorons sa complexité. Tout d'abord parce qu'il n'est pas facile de le décrire clairement. Mais aussi parce que, en introduisant des intermédiaires non indispensables entre `sbgp` et le DUT il devient plus fastidieux de contrôler tous les paramètres du test et tous les attributs BGP.

Nous avons également défini une méthode qui permet de comparer rapidement les résultats des différents tests. Cette méthode nous donne une première estimation de l'effet de différentes configurations sur les temps de traitement des UPDATE BGP. Elle permet de mettre clairement en évidence des effets de plus de 0.1 msec.

Nous avons utilisé cette méthodologie pour étudier le fonctionnement interne du processus BGP, et pour mettre en évidence les paramètres BGP et les facteurs externes qui influencent le plus le temps de traitement des UPDATE. Nous avons utilisé comme matériel un routeur commercial dont l'implémentation ne nous est pas connue, à la fois pour vérifier si l'implémentation commerciale semble bien fonctionner conformément aux spécifications et descriptions générales du protocole, mais aussi pour vérifier la validité de certaines idées préconçues à propos de l'impact potentiel de certains facteurs sur la convergence de BGP.

Le premier enseignement que nous pouvons retirer de notre étude est que, lorsque les tests sont réalisés dans les mêmes conditions, les résultats sont globalement identiques. Ce constat nous a permis de faire l'hypothèse que si on modifie un paramètre et que des temps de traitement différents sont observés, c'est probablement dû à l'effet du paramètre.

Au moyen de notre dispositif de test, nous avons pu brosser un portrait assez général du fonctionnement interne du routeur BGP. Il nous a permis de mettre en évidence l'importance relative des grands composants internes du routeur BGP dans les temps de traitement des UPDATE BGP, à savoir le processus de décision, l'application de politiques à l'entrée ou à la sortie du routeur et l'agrégation de routes. Il nous a en plus permis de déterminer la durée de certaines procédures particulières, comme l'examen d'une clause match dans le filtrage de routes.

Nous avons montré que pour les critères étudiés, la durée du processus de décision était sensiblement la même, sauf dans le cas du MED. Nous avons attribué la différence observée à la nécessité d'effectuer une vérification supplémentaire, externe au processus de décision proprement dit, lorsque ce critère est utilisé.

Nous avons également montré que la durée de la phase de filtrage de route dépendait du type de filtre utilisé, mais aussi du nombre de critères à vérifier avant d'accepter une route ainsi que du fait que la politique est appliquée à l'entrée ou à la sortie du routeur. Nous avons montré que lorsqu'on combinait le filtrage de routes et la manipulation d'attributs, l'effet de la manipulation d'attribut est perceptible à l'entrée du routeur, mais pas à la sortie. Ces constats nous permettent de faire l'hypothèse non seulement qu'il s'agit effectivement de deux composants différents, mais en plus que leur implémentation semble différente.

Nous avons également mis en évidence l'impact potentiel que pouvait avoir l'agrégation sur le temps de traitement des préfixes spécifiques, et le fait que l'intensité de l'effet dépendait de la distance entre le masque du préfixe et celui de l'agrégat. Si l'effet est spectaculaire lorsque les préfixes spécifiques appartiennent au même espace d'adresses que l'agrégat, on peut néanmoins le relativiser parce qu'une telle situation se rencontre probablement très peu fréquemment. Néanmoins, on observe déjà un léger effet lorsque le préfixe spécifique appartient à un espace d'adressage différent de l'agrégat.

Parmi les paramètres de configuration propres à BGP, très peu influencent le temps de traitement des UPDATE BGP : la commande `next-hop-self` permet de le diminuer, alors que les commandes `maximum-prefix` et `password` l'augmentent légèrement. Nous avons également constaté que le Holdtimer ne semblait pas avoir d'impact : ce constat nous amène à nous demander pourquoi la valeur par défaut utilisée par Cisco est supérieure à la valeur recommandée par le RFC, alors qu'une valeur plus basse est susceptible d'accélérer la détection de défaillances éventuelles d'un routeur. Par contre le MRAI influence très fortement les temps de redistribution des préfixes. Ce paramètre est utilisé pour permettre à des préfixes partageant des attributs communs d'être regroupés dans un seul UPDATE, mais aussi empêcher les oscillations de routes au cas où un préfixe serait sélectionné à de multiples reprises avant l'expiration du timer.

Enfin, parmi les autres facteurs externes susceptibles d'influencer le temps de traitement des UPDATE, nous avons constaté que la taille de la table de routage ne semblait influencer la durée du processus de décision que dans le cas où une autre instance de la route évalué y est déjà représentée. Par contre, le nombre de voisins en aval du routeur, ainsi que la position de ce voisin dans la configuration semblent influencer fortement les temps de traitement. L'augmentation des temps de traitement s'ajoute à l'effet observé dans le cas de l'application de politiques de routage. L'utilisation de la commande `peer-group` permet de diminuer les temps de traitement, qu'une politique soit appliquée ou non, de sorte que pour une même position du pair, ils semblent indépendants du nombre de voisins utilisés dans la configuration. Par contre, cette commande n'a aucun effet sur la différence de temps de traitement observée entre les différents voisins d'un même routeur.

En conclusion, on peut se demander s'il est vraiment utile de continuer à étudier les propriétés de convergence de BGP, étant donné qu'en fin de compte, l'utilisation du MRAI va ralentir la redistribution des UPDATE, et que dans ces conditions, l'impact de chaque phase du processus de décision semble marginal. Dans l'état actuel des choses, c'est difficile à imaginer. Il faut dire que nous nous sommes limités à l'étude d'un modèle simplifié, où un UPDATE contenant un seul préfixe est envoyé une fois par seconde. Nous avons pu déterminer la durée de certaines phases du processus de sélection de routes, mais nous ne savons pas si un verrou n'est pas imposé sur certaines de ces phases. Que se passe-t-il dans des cas plus complexes, lorsque les UPDATE contiennent plusieurs préfixes, ou lorsqu'un routeur doit traiter les annonces de plusieurs voisins simultanément ? Que se passe-t-il lorsque la densité ou la fréquence du trafic de contrôle augmente ? La durée de chaque phase n'est-elle pas à même de perturber le trafic de données ?

Bibliographie

[Ahu00] **Abha AHUJA.** Experimental study of delayed internet routing convergence. RIPE37-routing WG. 14/11/00. http://www.ripe.net/ripe/meetings/archive/ripe-37/presentations/RIPE-37-convergence/

[Ahu01] **Abha AHUJA.** The impact of policy and topology on Internet routing convergence. RIPE EOF 38, 22/01/01. http://www.ripe.net/ripe/meetings/archive/ripe-38/presentations/eof-38/

[BBG+01] **Steve BELLOVIN, Randy BUSH, Timothy G. GRIFFIN and Jennifer REXFORD.** Slowing routing table growth by filtering based on address allocation policies. June 2001. http://www.research.att.com/~jrex/papers/filter.pdf (Accès : 12/2002)

[BHR+02] **H. BERKOWITZ, S. HARES, A. RETANA, P. KRISHNASWAMY, M. LEPP, E. DAVIES.** Benchmarking Methodology for Basic BGP Device Convergence. draft-ietf-bmwg-bgpbas-01.txt, February 2002, *work in progress*.

[Cisco-BGP] **Cisco**. BGP commands. http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/12cgcr/np1_c/1cprt1/1cbgp.htm (Accès: 2002)

[Cisco-DP] **Cisco**. Processus de décision. http://www.cisco.com/warp/public/459/25.shtml

[Cisco-MED] **Cisco.** How BGP Routers Use the Multi-Exit Discriminator for Best Path Selection. http://www.cisco.com/warp/public/459/37.html

[Cpan] **Cpan.** Comprehensive Perl Archive Network. http://www.cpan.org/ (Accès : 05/2003).

[Eth] **Ethereal.** The Ethereal Network Analyzer. http://www.ethereal.com/ (Accès : 05/2003).

[GW99] **Thimothy G. GRIFFIN and Gordon WILFONG** An Analysis of BGP Convergence Properties. Sigcom99

[Hal97] **Bassam HALABI**. Internet Routing Architectures. Cisco Press, 1997

[Hus01] **Geoff HUSTON**. Analyzing the Internet's BGP Routing Table. January 2001. http://www.potaroo.net/papers.html

[LAB+00] **Craig LABOVITZ, Abha AHUJA, Abhijit BOSE, Farnam JAHANIAN**. Delayed Routing Convergence. . In Proc. ACM SIGCOMM '00 (Stockholm, Sweden, 2000), pp. 175--187. http://citeseer.nj.nec.com/labovitz00delayed.html

[LAB-TR-00] **Craig LABOVITZ, Abha AHUJA, Abhijit BOSE, Farnam JAHANIAN**. An Experimental Study of Internet Routing Convergence. *Technical Report MSR-TR-2000-08, Microsoft Research, 2000*.

[LAW+01] **Craig LABOVITZ, Ahba AHUJA, Roger WATTENHOFFER, Srinivasan VENKATACHARY.** The Impact of Internet Policy and Topology on Delayed Routing Convergence. . *INFOCOMM, pp 537-546, 2001*. http://citeseer.nj.nec.com/article/labovitz01impact.html

[Man02] **Olaf MAENNEL**. RTG, a routing table generator. http://www.olafm.de/2002/networking/rtg.html

[Merit] **MRT**. Multi-threated routing toolkit. http://www.mrtd.net/ ; http://www.merit.edu/mrt/

[NC02] **Johan NYKVIST and Lenka CARR-MOTYCKOVA.** Simulating Convergence Properties of BGP. In *Proceedings of the 11th International Conference on Computer Communications and Networks (IC3N 2002)*. pages 124-129. 2002.

[NSS+01] **Evi NEMETH, Garth SNYDER, Scott SEEBASS, Trent R. HEIN et al**. Unix system administration handbook. Prentice Hall PTR, 2001

[Par01] **William R. PARKHURST**. Cisco BGP-4 Command and configuration Handbook. Cisco Press, 2001

[Potaroo]. BGP table statistics. http://www.potaroo.net (Accès : 04/2003)

[PZW+02] **Dan PEI, Xiaoliang ZHAO, Lan WANG, Dan MASSEY, Allison MANKIN, S. Felix WU, Lixia ZHANG.** Improving BGP Convergence Through Consistency Assertions. IEEE Infocom 2002, New York, June 2002.

[QB02] **Bruno QUOITIN and Olivier BONAVENTURE**. A survey of the utilization of BGP community attribute. Internet-draft. February 2002

[RFC1771] **Y. REKHTER, T. LI**. RFC 1771 A Border Gateway Protocol 4 (BGP-4). March 1995.(Obsoletes RFC1654)

[RFC1772] **Y. REKTHER, P. GROSS**. RFC 1772 Application of the Border Gateway Protocol in the Internet. March 1995. (Obsoletes RFC1655)

[RFC1878] **T. PUMMILL & B. MANNING**. Variable Length Subnet Table For IPv4. December 1995.

[RFC3221] **G. HUSTON**. RFC3221. Commentary on Inter-domain routing in the Internet. December 2001.

[SG01] **Aman SHAIKH, Albert GREENBERG**. Experience in Black-box OSPF Measurement. In *Proc. ACM SIGCOMM Internet Measurement Workshop (IMW)*, November 2001.

[SP02] **Randal L. SCHWARTZ & Tom PHOENIX**. Introduction à Perl. O'Reilly, 2002

[Ste01] **John W. STEWART III**. BGP4 – Inter-Domain Routing in the Internet. Addison-Wesley, 2001

[Tan99] **Andrew TANNENBAUM**. Réseaux. Dunod/Prentice Hall, 1999

[Teth] **Tethereal.** Dump and analyze network traffic. http://www.ethereal.com/tethereal.1.html (Accès : 05/2003).

[VGE99] **Kannan VARADHAN, Ramesh GOVINDAN, Deborah ESTRIN. Persistent Route Oscillations in Inter-Domain Routing (1999)**. In Computer Networks, vol. 32 nr 1, pp 1—16, 2000, http://citeseer.nj.nec.com/varadhan99persistent.html

[WZP+02] **Lan WANG, Xiaoliang ZHAO, Dan PEI, Randy BUSH, Daniel MASSEY, Allison MANKIN, S. Felix WU, Lixia ZHANG**. Observation and Analysis of BGP Behavior under Stress. *Proceedings of the second ACM SIGCOMM Internet Measurement Workshop*, November 2002.

[HKL+02] **Sue HARES, Padma KRISHNASWAMY, Marianne LEPP, Alvaro RETANA, Howard BERKOWITZ, Elwyn DAVIES**. BGP Convergence Measurement Issues. http://www.ietf.org/proceedings/01dec/slides/bmwg-1/, 15/01/2002.

Annexes

A. Description du laboratoire

Le laboratoire d'apprentissage des réseaux de Namur (LEARN) est équipé de quatre PC Linux et de quatre routeurs Cisco 3640 supportant une version IOS 12.7 interconnectés entre eux de manière à pouvoir construire des topologies complexes.

Chaque PC dispose de quatre cartes Ethernet et chaque routeurs de six ports Ethernet. Chaque interface Ethernet possède son adresse IP propre. L'une d'entres elle appartient au pool des adresses IP des FUNDP, ce qui permet à la machine d'être accessible depuis le réseau des Facultés. Les autres appartiennent au pool des adresses privées de classe A (10/8) et sont configurées de manière à appartenir à des sous-réseaux 10/24 différents. Les adresses du pool privé peuvent être modifiées selon les besoins expérimentaux.

La plupart des connexions entre routeurs ou entre un PC et un routeur passent par un hub, ce qui permet à plusieurs machines de communiquer sur un même sous-réseau.

B. Commandes réseau sous linux

Les descriptions ci-dessous nous donnent un aperçu de l'intérêt de diverses commandes réseau pour ce travail. Une description détaillée se trouve dans [NSS+01] ou dans les « man pages » de ces commandes.

ifconfig : les PC du laboratoire disposent de plusieurs interfaces Ethernet nommées eth0, eth1, eth2, eth3. Une adresse Ethernet est configurée dans la mémoire ROM de chaque carte, tandis que l'adresse IP est configurée par le gestionnaire de la machine. La commande `ifconfig` permet de visualiser l'état d'une interface ; les renseignements les plus intéressants sont l'adresse IP de l'interface, le masque de sous-réseau, l'état d'activité de l'interface et la taille maximale des paquets IP sur cette interface. La commande `ifconfig` permet également et de la configurer manuellement une interface. Cette commande est particulièrement intéressante dans le cadre de ce travail pour configurer des interfaces virtuelles. Ceci nous permet de simuler une ou plusieurs connexions virtuelles entre un PC et un routeur sur un seul lien physique, sans changer les adresses des interfaces physiques. La syntaxe de cette commande est la suivante :

```
ifconfig eth3:1 157.164.170.4 netmask 255.255.255.0\
broadcast 157.164.170.255
```

route : la commande `route` a deux utilisations principales. Elle sert à afficher le contenu de la table de routage du PC. Elle sert également à modifier la table de routage en ajoutant ou en supprimant des routes. La syntaxe générale de cette commande est :

route {add|del} -net *destination/netmask* gw [*gateway*]

où *destination* représente l'adresse IP à atteindre, *netmask* le nombre de bits correspondant au masque de sous-réseau et *gateway* l'adresse IP du prochain routeur sur la route vers la destination.

netstat : `netstat` est un outil qui permet de diagnostiquer des problèmes du réseau. Il permet d'afficher de façon conviviale les différentes tables maintenues par le kernel Linux. Il informe, par exemple, sur le contenu de la table de routage, donne des statistiques sur l'état des interfaces et le nombre de paquets envoyés et reçus. Il fournit également des renseignements sur l'état des serveurs, fournit la liste des différentes connexions TCP établies ainsi que les numéros de ports pour lesquels un serveur est en attente d'un établissement de connexion.

tethereal :`tethereal` est un analyseur de trafic qui fonctionne en mode terminal. Cet outil permet de capturer les paquets qui transitent sur le réseau. La capture peut se faire en format binaire ou en format texte. Le format binaire est exploitable par des outils graphiques, tels que `ethereal`, disponible pour des plate-formes UNIX et Windows.

Pour les captures en mode texte, les trames sont parfaitement identifiées, ainsi que les différentes couches de protocoles à l'intérieur d'une trame.

L'exemple ci-dessous nous montre à quoi ressemble la première section d'une trame correspondant à un message BGP KEEPALIVE. Elle nous donne des informations générales telles que : numéro de trame, longueur de la trame, moment de la capture et délai par rapport à la trame précédente.

```
Frame 40 (73 on wire, 73 captured)
    Arrival Time: Nov  1, 2002 10:16:38.7637
    Time delta from previous packet: 0.001724 seconds
    Time relative to first packet: 51.344472 seconds
    Frame Number: 40
    Packet Length: 73 bytes
    Capture Length: 73 bytes
```

Les autres sections nous donnent dans l'ordre, des informations relatives à la couche Ethernet, au protocole IP, au protocole TCP, et au protocole BGP. Elles ne seront pas détaillées ici.

tftp : `tftp` est un protocole de transfert de fichier qui peut être utilisé par les machines lorsqu'elles démarrent. Il permet de sauvegarder les fichiers de configuration de routeurs sur un serveur et de les charger. Les chargements et sauvegardes se font au moyen de la commande `copy`, qui prend comme paramètres les URL des fichiers.

cisconf : `cisconf` permet de gérer les configurations de routeurs Cisco en utilisant le protocole `tftp`. L'intérêt de cet outil réside dans le fait qu'il permet de charger les fichiers de configurations ou de les sauvegarder sur le routeur à partir du serveur `tftp` local.

C. MRT

MRT (multi-threated routing software), est un ensemble de d'outils qui peuvent être utilisés comme démons de routage, pour la génération de trafic de test mais également pour la collecte et l'analyse du trafic Internet. Ces outils ont été conçus selon une approche modulaire, ce qui fait qu'ils sont fréquemment utilisés par les universités pour le prototypage de protocoles de routage expérimentaux et d'algorithmes de politiques inter-domaines. MRT fournit principalement des outils de routage et de mesure de performance réseaux. Sa capacité à capturer une session BGP, à l'enregistrer en temps réel et à la rejouer à particulièrement retenu notre attention.

C.1 MRTd

c'est un démon de routage qui lit des fichiers de configuration semblables à ceux des routeurs Cisco et qui supporte une interface en ligne de commande très semblable à celle des routeurs Cisco. Il est capable de sauvegarder sa table de routage BGP complète dans un fichier, et de réinjecter le contenu de ce fichier dans une session BGP avec ses voisins.

Le démon est appelé par la commande :

mrtd [-f <config-file>] [-l <routing-table-file>] [-v]

MRTd lit le fichier de configuration (par défaut, /etc/mrtd.conf), qui contient les configurations des protocoles de routage, des voisins et des politiques. Une fois le démon lancé, il est accessible par la commande :

telnet localhost <port>

où port est le numéro de port configuré pour mrtd dans le fichier /etc/services. Cela permet d'entrer dans l'interface ligne de commandes. A quelques détails près, les instructions de configuration pour BGP sont semblables à celles que l'on trouve sur les routeurs Cisco.

C.2 sbgp

sbgp est un annonceur BGP4 simple. Il n'est pas capable d'appliquer des politiques aux routes, ni de maintenir les informations de routage qu'il a apprises dans une table. Par contre, il a la capacité d'enregistrer les informations de routage (updates) qu'il reçoit de ses pairs, et d'injecter des informations de routage dans une session BGP.

Le démon sbgp est lancé par la commande :

sbgp [-v] [-i <input-file>] [-o <output-file>] my_AS [peer_IP peer_AS]$^{+}$

sbgp injecte les routes du fichier binaire input-file dans toutes les sessions BGP qu'il a établies. Alternativement, sbgp peut sauvegarder les routes qu'il apprend via toutes ses sessions BGP dans le fichier binaire output-file.

C.3 route_btoa

route_btoa produit une description ASCII formatée des paquets de mises à jour BGP4 à partir des messages MRT binaires, par exemple les fichiers capturés par sbgp. La commande possède les options suivantes :

route_btoa [-i input-file] [-m]

L'entrée est un fichier binaire MRT. L'application de la commande route_btoa sur un fichier binaire MRT dans le cas d'une annonce (gauche) ou d'un retrait (droite) de route

```
TIME: 09/29/02 12:05:04
TYPE: BGP/UPDATE
FROM: 10.0.0.4 AS400
TO: 10.0.0.14 AS1400
ORIGIN: IGP
ASPATH: 400
NEXT_HOP: 10.0.0.4
MULTI_EXIT_DISC: 50
ATOMIC_AGGREGATE
AGGREGATOR: AS400 138.48.115.155
ANNOUNCE
 196.0.0.0/8
```

```
TIME: 09/29/02 12:06:54
TYPE: BGP/UPDATE
FROM: 10.0.0.4 AS400
TO: 10.0.0.14 AS1400
WITHDRAW
 138.48.0.0/16
 193.0.0.0/24
 194.0.0.0/24
```

L'option -m fournit un format qui permet d'être traité plus facilement par des scripts Perl pour effectuer des calculs. Elle ne préserve pas la limite des paquets : une ligne est générée pour chaque information de NLRI (paquet) d'un paquet BGP.

```
BGP|1033293904|A|10.0.0.4|400|194.0.0.0/24|400 100 4 5 6|IGP|10.0.0.1|0|50|1234|NAG||
BGP|1033294014|W|10.0.0.4|400|193.0.0.0/24
```

On trouve les champs suivants :

[1] Protocol : BGP
[2] Time : représente le nombre de secondes écoulées depuis un moment de référence (1/01/1970 00:00:00, ou Epoch) au moment de la capture.
[3] Type : vaut A pour une annonce et W pour un retrait de route
[4] PeerIP : adresse IP du voisin
[5] PeerAS : numéro d'AS du voisin
[6] Prefix : préfixe concerné par la mise à jour
[7] AS-PATH
[8] ORIGIN
[9] NEXT-HOP
[10] LOCAL-PREF
[11] MED
[12] COMMUNITY
[13] ATOMIC-AGGREGATE
[14] AGGREGATOR

Les champs 7 à 14 sont spécifiques aux annonces de routes (Announce), et correspondent aux attributs de chemin de BGP. Ils ne sont pas présents pour les retraits (Withdrawal).

C.4 route_atob

route_atob convertit des descriptions ASCII de messages au format binaire MRT.
La commande possède l'option suivante :

route_atob –i {ascii-input-file|stdin}

L'entrée doit être conforme à ce qui est produit par la commande route_btoa.

D. configuration d'un routeur BGP

Il est possible de modifier les configurations des routeurs en y accédant par `telnet`. Les routeurs fonctionnent selon trois mode. Le mode **exec** permet d'exécuter quelques commandes de base sans modifier la configuration du routeur. C'est le mode par défaut dans lequel on se trouve quand on se connecte au routeur par `telnet`. Le mode **exec privilégié** permet de modifier certains paramètres du routeur et d'accéder à des commandes complémentaires. Ce mode permet notamment de visualiser les diverses tables et configurations du routeur. On accède au mode **exec privilégié** à partir du mode **exec** en tapant la commande `enable`. Le mode **global config** permet quant à lui de modifier complètement la configuration du routeur. Il permet entre autres d'attribuer une adresse IP aux différentes interfaces, de configurer des routes statiques, de spécifier des access-lists qui permettront d'appliquer des politiques de routage et de lancer divers processus de routage. On accède à ce mode à partir du mode **exec privilégié** en tapant la commande `configure terminal`.

Nous utiliserons principalement les modes **global config** pour configurer les routeurs et **exec privilégié** pour ce qui est du diagnostic et du déboguage. On peut trouver dans [Par01], ainsi que sur le site de Cisco [Cisco-BGP], un descriptif détaillé des diverses commandes pour configurer BGP-4 sur des routeurs Cisco. Les commandes qui ont le plus d'intérêt pour ce travail sont décrites ci dessous.

Lancer un processus BGP

router bgp <as-number>

Cette commande est exécutée à partir du mode **global config**. Elle permet de lancer un processus BGP, et d'entrer dans le mode de configuration du routeur pour BGP. Le paramètre *as-number* permet de définir le système autonome auquel le routeur appartient. Les routeurs Cisco permettent d'exécuter un seul processus BGP à la fois : ils appartiennent à un et un seul système autonome.

Configuration des voisins BGP

neighbor {ip address | peer-group-name**}** **remote-as** <as-number>

Cette commande permet de configurer la session BGP avec un voisin. Elle spécifie l'adresse IP et le numéro d'AS du voisin. Lors de la réception du message d'ouverture, le numéro d'AS annoncé par le voisin est comparé avec celui qui est configuré. S'il n'y a pas de correspondance, la session ne s'établit pas et un message de notification est envoyé à l'émetteur. Si le numéro d'AS du voisin est le même que celui qui est configuré dans le routeur, la session qui s'établit est de type IBGP (interne) ; sinon, il s'agit d'une session EBGP (externe).

Annonce de routes

network <ip-address>
network <ip-address> **mask** <network-mask>

L'objectif de ces commandes est de déterminer quels réseaux seront annoncés aux voisins BGP. Pour qu'un réseau puisse être annoncé, il doit être connu du routeur qui fait l'annonce : il s'agit d'un réseau directement connecté, statique, ou appris par un protocole de routage dynamique. Le nombre de réseaux qui peuvent être listés par la commande network est limité.

Redistribution de routes

redistribute <protocol>

Cette commande permet de redistribuer dans BGP des routes qui ont été apprises par un autre protocole que BGP. Les routes peuvent être redistribuées entre autres à partir de ospf, static et connected.

configuration d'options

neighbor {ip address | peer-group-name} **update-source** <interface>

permet aux sessions BGP d'utiliser une autre interface pour les connexions TCP. L'intérêt de cette option réside dans la possibilité d'utiliser une interface virtuelle plutôt qu'une interface physique pour l'établissement de la connexion.

neighbor {ip address | peer-group-name**}** **description** <text>

Cette option associe une description à un voisin BGP.

neighbor {ip address | peer-group-name**}** **password** <password>

Demande une authentification MD5.

neighbor {ip address | peer-group-name**}** **remove-private-as**

Cette commande est utilisée pour enlever les numéros d'AS privés des annonces vers les voisins. Les numéros d'AS privés appartiennent à la classe 64512-65535 et ne doivent jamais être annoncés dans l'Internet. Elle ne s'applique que vis-à-vis de pairs EBGP et uniquement si l'AS-PATH ne contient que des AS privés.

neighbor {ip address | peer-group-name**}** **weight** <weight>
neighbor {ip address | peer-group-name} **send-community**
neighbor {ip address | peer-group-name} **soft-reconfiguration-inbound**

Configuration du filtrage de route et de la manipulation d'attributs

neighbor {ip address | peer-group-name} **distribute-list** {ip-access-list-number-or-name | prefix-list-name} **in | out**
neighbor {ip address | peer-group-name} **filter-list** <as-path-list-number> **in | out**
neighbor {ip address | peer-group-name} **prefix-list** <prefix-list-name> **in | out**
neighbor {ip address | peer-group-name} **route-map** <route-map-name> **in | out**

Les trois premières commandes permettent d'effectuer le filtrage des routes, à l'entrée ou à la sortie du routeur. Elles filtrent les mises-à-jour qui entrent dans le routeur ou en sortent sur base des critères spécifiés dans une ip access-list ou une ip prefix-list. Les filtres des prefix-list et distribute-list s'appliquent aux adresses réseaux, tandis que les filtres des filter-list s'appliquent aux numéros d'AS.

Les route-map, quant à eux, permettent non seulement d'accepter ou de refuser des routes en entrée ou en sortie, mais également, si une route est acceptée, de modifier ses attributs. Le filtrage peut s'effectuer sur base d'une correspondance de système autonome, de communauté ou de réseau.

Configuration des groupes de pairs

neighbor <peer-group-name> **peer-group**
neighbor <ip-address> **peer-group** <peer-group-name>

La première commande permet de créer le groupe de pair et de lui donner un nom. La deuxième commande permet d'affecter un voisin à un groupe de pairs existant. Le groupe de pair permet de faciliter les configurations des voisins auxquels on applique les mêmes politiques. De plus, il permet de calculer une seule fois une mise à jour de routage et de l'envoyer à tous les voisins. Conditions : le routeur qui utilise les groupes de pairs ne doit pas servir de hub pour les messages qu'il reçoit d'un des membres du groupe. Tous les membres du peer-group doivent se trouver sur le même sous-réseau.

Timers BGP

timers bgp <keepalive> <holdtime>

Cette commande permet de fixer globalement les valeurs de holdtime et keepalive pour tous les voisins. Le keepalive spécifie la fréquence à laquelle un routeur envoie ses messages KEEPALIVE à ses voisins. Le holdtime est le temps maximum qui peut s'écouler entre la réception de deux

messages consécutifs (KEEPALIVE ou UPDATE) avant de considérer qu'un voisin est mort et que la session soit terminée.

Commandes spécifiques à BGP

bgp deterministic-med
bgp log-neighbor-changes

Agrégation de routes

aggregate-address <address> <mask>
aggregate-address <address> <mask> **as-set**
aggregate-address <address> <mask> **as-set advertise-map** <route-map-name>
aggregate-address <address> <mask> **as-set route-map** <route-map-name>

L'objectif de ces commandes est de créer un agrégat dans la table BGP. L'agrégat est créé uniquement si une entrée plus spécifique existe dans la table BGP. Cette forme de la commande permet l'annonce de l'agrégat et des routes spécifiques qui font partie de l'agrégat. L'option as-set permet de garder l'information d'AS-PATH des routes spécifiques qui forment l'agrégat. La variante advertise-map est utilisée pour spécifier quelle partie de l'information d'AS-PATH est conservée dans l'agrégat. La variante route-map permet de modifier les attributs BGP de l'agrégat.

Outils de diagnostic

show ip bgp

permet d'afficher le contenu de la table de routage BGP.

show ip bgp prefix

permet d'afficher les renseignements concernant un (ou plusieurs) préfixes. Cette commande permet notamment de vérifier les valeurs de l'attribut Community associées à une route.

show ip bgp neighbor

permet de visualiser l'état d'activité des différentes sessions BGP (Active, OpenSent, Established ...).

show ip route

permet d'afficher le contenu de la table de forwarding

clear ip bgp *

permet de réinitialiser une session BGP

E. gen_pref

L'outil doit nous permettre de modifier les valeurs de certains paramètres à partir d'une instruction en ligne de commande. Le script **gen_pref** assure les fonctionnalités suivantes :

- Découpe du fichier en trames
- Détermination du moment de la capture (par rapport au premier paquet du fichier)
- Détermination de l'adresse IP source et destination
- Détermination du type de message (OPEN, KEEPALIVE, NOTIFICATION, UPDATE)
- Détermination du type d'annonce (Annonce ou retrait de route)
- Détermination de quelques attributs de chemin (ORIGIN, AS-PATH, NEXT-HOP, COMMUNITIES)
- Détermination du ou des préfixes contenu dans le message

```
#!/usr/bin/perl -w

## gen_pref by Christine Vandesteene
##
## Objectif : génération d'updates BGP au format route_btoa -m
##
## Remarque : Divers paramètres peuvent être fixés
##            au moyen d'options disponibles à partir
##            de la ligne de commande (valeurs des attributs,
##            nombre de pairs, adresse IP de l'annonceur, nombre de
##            préfixes...)
##
##       Donne la possibilité de mélanger les préfixes
##
##
## Usage :   perl gen-pref [-h] [-o <output>] [-p <peer-ip>] [-P <nr-of-pref>]\
##           [-A <aspath-lgth>] [-T <seconds>] [-W] [-M] [-m <MED>]\
##           [-a <ASN>] [-c <community>] [-l <local-pref>] [-O <origin>]
##
## Input :   aucun
##
## Output :  fichier texte représentant chaque message BGP sous forme
##           d'une ligne dont les champs sont séparés par |
##
##         Chaque ligne comprend les champs suivants:
##                    - BGP (protocole)
##                    - timestamp (permet de fixer l'intervalle
##                              entre les annonces)
##                    - A ou W (annonce ou retrait)
##                    - adresse IP du voisin
##                    - AS du voisin
##                    - prefix
##                    - AS-PATH
##                    - ORIGIN
##                    - NEXT-HOP
##                    - LOCAL-PREF
##                    - MED
##                    - COMMUNITY
##                    - NAG (AGGREGATE_ADDRESS)
##                    - (AGGREGATOR)
##
##       Le champ AGGREGATE_ADDRESS garde la valeur NAG. Le champ
##           AGGREGATOR est laissé libre. Nous ne formons pas d'agrégat
##           mais le format est conforme à ce qui est produit
##           par la commande route_btoa -m sur un fichier binaire
##           généré par sbgp.
##
##
## Version
##
## 0.03 [2002-06-13]
## 0.04 [2002-11-16] : ajout de commentaires
```

```
## Module permettant de fixer des options à partir de la ligne de commande
## Configuration des valeurs par défaut

use Getopt::Std;

my %options=(
      "o"=>"-",                         ## Output. Par défaut : stdout
      "p"=> "193.190.23.14",            ## Adresse IP de l'annonceur
      "A"=> 1,                          ## Longueur de l'AS-PATH
      "P"=>10,                          ## Nombre de préfixes
      "T"=>0,                           ## Intervalle entre les messages
      "m"=>0,                                 ## MED
      "c"=>"",                    ## COMMUNITY
      "l"=>0,                           ## LOCAL_PREF
      "O"=>"IGP",                       ## ORIGIN
);

getopts("ho:p:P:A:T:WMm:a:c:l:O:", \%options);

if ($options{h}) {
      die "\nusage: perl gen-pref [-h] [-o <output>] [-p <peer-ip>] [-P <nr-of-pref>]\\
\t\t [-A <aspath-lgth>] [-T <seconds>] [-W] [-M] [-m <MED>]\\
\t\t [-a <ASN>] [-c <community>] [-l <local-pref>] [-O <origin>]\n
\t-h : help
\t-o : output file (default: stdout)
\t-p : peer ip-address
\t-P : number of prefixes
\t-A : AS-path length
\t-T : interval between updates
\t-W : withdraws prefixes
\t-M : mix prefixes
\t-m : MED value for all prefixes
\t-a : marker AS (value that will bi inserted in 2nd position)
\t-c : community value (should be between quotes if more than 1 value)
\t-l : LOCAL_PREF value
\t-O : ORIGIN value
";}


## OPTIONS

$output = $options{o};                          ## Nom du fichier de sortie
$number_of_prefixes = $options{P};    ## Nombre de préfixes dans le fichier
$as_path_length = $options{A};                  ## longueur de l'AS-PATH
$interval_between_updates= $options{T}; ## Intervalle séparant les messages
$peer_ip= $options{p};                          ## Adresse IP de l'annonceur
$community= $options{c};              ## Attribut COMMUNITY
$med=$options{m};                     ## Attribut MED
$plus_as=$options{a};                           ## AS utilisé comme marqueur
$local_pref=$options{l};              ## Attribut LOCAL_PREF
$origin=$options{O};                            ## Attribut ORIGIN


## GLOBAL PARAMETERS

$timestamp=1021000000;                ## Temps correspondant au premier message
$AS=65000;                      ## Numéro d'AS de l'Annonceur
                                ## Valeur du premier AS dans l'AS-PATH
                                ## Choisi dans le pool des AS privés > 65000



## Fixation des AS dans l'AS-PATH du premier message
## Tous les AS de l'AS-PATH sont différents
##
## Pour que les préfixes soient traités individuellement
## il faut qu'ils diffèrent par au moins 1 attribut
## L'AS-PATH a été sélectionné, parce que c'est un attribut obligatoire
## et qu'il présente suffisamment de possibilités de combinaisons
## s'assurer que chacun des préfixes annoncés s'est vu attribuer
## un AS-PATH différent
```

```
for ($j=0;$j<$as_path_length;$j++){ $as_path[$j]=$AS+$j;}


## Ouverture d'un descripteur de fichier pour l'output

open (F, ">$output") or die "Problème : $!";


## Génération des préfixes, tous différents
## Les préfixes sont représentés sous la forme A.B.C.D/x
##  - le premier octet est fixé à 10 (tous préfixes appartiennent
##          au pool des adresses privées de classe A)
##  - le dernier octet est fixé à 0 (on forme des sous réseaux
##          avec un masque de 24
##  - le deuxième et troisième octet sont variables. Cela permet
##          de générer plus de 65000 préfixes différents


$i=0;
for ($b=0;$b <256; $b++){
for ($c=0;$c < 256;$c++){
push @pref, "10.$b.$c.0/24";
last if $i == $number_of_prefixes -1;
$i++;
}

last if $i == $number_of_prefixes -1;
}


## Fixation d'un AS-PATH unique pour chaque préfixe
##  (possible uniquement si la longueur de l'AS-PATH
##   est au moins égal au nombre d'octets nécessaires
##   pour représenter tous les préfixes générés +1)
##
## Mélange des préfixes, si l'option a été choisie
##
## Ajout de l'AS marqueur en 2e position de l'AS-PATH
##
## Impression dans le fichier de sortie du message
##  - annonce du préfixe
##  - retrait du préfixe, si l'option a été choisie
##          (suit immédiatement l'annonce)


## Choix du préfixe à traiter
##  - le premier de la liste, si les préfixes doivent être ordonnés
##  - choisi au hasard, si les préfixes doivent être mélangé

$max = $#pref+1;
for ($k=0;$k<$max;$k++) {
    if ($options{M}){
            $my_pref = splice (@pref, int  (rand($#pref+1)),1)
            } else {$my_pref= shift @pref};


## Attribution de l'AS-PATH à ce préfixe
##  - le premier AS reste identique (celui de l'annonceur BGP)
##  - si la longueur de l'AS-PATH est >= 2, on ajoute la valeur du
##          3e octet du préfixe à la valeur de base du 2e AS de l'AS-PATH
##          (le 3e octet du préfixe est le premier à varier
##           dans notre méthode de génération des préfixes)
##  - si la longueur de l'AS-PATH est >= 3, on ajoute la valeur du
##          2e octet du préfixe à la valeur de base du 3e AS de l'AS-PATH
## Cette méthode permet d'obtenir des AS-PATH uniques dans l'intervalle d'ASN
##  du pool privé si la longueur de l'AS-PATH est suffisante

    @my_as_path=@as_path;
```

```
        if (defined $my_as_path[1]){
                my ($a, $b, $c, $d)= split /\./, $my_pref;
                $my_as_path[1]+=$c;
                if (defined $my_as_path[2]){ $my_as_path[2]+=$b;}
                }


## Ajout d'un AS marqueur en 2e position de l'AS-PATH
##  (critère de sélection, marqueur)


        if ($options{a}){
                splice (@my_as_path, 1,0,$plus_as);
                }

## Incrémentation du timestamp entre chaque message
##   - l'utilisateur peut choisir la valeur à ajouter


        $timestamp+=$interval_between_updates;



## Impression des annonces dans le fichier de sortie


        print F "BGP|$timestamp|A|$peer_ip|$AS|$my_pref|@my_as_path|";
        print F "$origin|$peer_ip|$local_pref|$med|$community|NAG||\n";



## Si des retraits de route doivent être effectués,
## impression du message de retrait dans le fichier de sortie


        if ($options{W}) {
                $timestamp+=$interval_between_updates;
                print F "BGP|$timestamp|W|$peer_ip|$AS|$my_pref\n";}

} ## end for

close F;
```

F. init_table

```
#!/usr/bin/perl -w

## init_table by Christine Vandesteene
##
## Objectif : transformation de messages BGP d'un format texte
##            "machine-parseable" au format texte "human-readable"
##
## Usage :    perl init_table [-h] [-i <input>] [-o <output>]
##
## Input :    messages BGP au format "route_btoa -m"
##            chaque message est exprimé sur une ligne comprenant les champs
##                    - BGP (protocol)
##                    - timestamp (moment de la capture)
##                    - type de message (Announce, Withdraw)
##                    - adresse IP de l'annonceur
##                    - AS de l'annonceur
##                    - préfixe
##                    - AS_PATH
##                    - ORIGIN
##                    - NEXT_HOP
##                    - LOCAL_PREF
##                    - MED
##                    - COMMUNITY
##                    - ATOMIC_AGGREGATE
##                    - AGGREGATOR
##
## Output :   description ASCII du même message (format route_btoa).
##            Le contenu des champs est décrit explicitement
##
## Remarque : diverses sources possibles pour l'input :
##            transformation par route_btoa -m d'updates (sbgp), de
##                    table_dumps (mrtd, zebra)
##            génération par gen_pref
##
## Version
##
## 0.03 [2002-05-26]
## 0.04 [2002-11-16] : ajout de commentaires



## Module permettant de choisir des options à partir de la ligne de commande
## Fixation des paramètres par défaut

use Getopt::Std;

my %options=(
      "i" => "-",                       ## default : stdin
      "o" => "-",                       ## default : stdout
);

getopts("hi:o:", \%options);

if ($options{h}) { die "\nusage : perl init_table [-h] [-i <input>] [-o <output>]\n\t-h : help\n\t-i : input
file (array)\n\t-o : output file (text)\n";}



## OPTIONS

$input = $options{i};
$output = $options{o};



## PARAMETRES GLOBAUX
## valeurs pour la ligne
## TO: <PEER-IP-ADDRESS>  AS<PEER-AS_NUMBER>
```

```
## absentes dans le format route_btoa -m

$to_IP = "172.0.0.14";
$to_AS = "17214";


## Ouverture des descripteurs de fichiers
##  pour l'input et l'output

open (DF, "< $input") or die "Problème avec le fichier $input : $!";
open (F1, "> $output") or die "Problème avec l'output : $!";



## Traitement de l'entrée
## Séparation de chaque ligne en plusieurs champs sur base du séparateur |

while ($ligne=<DF>){

my ($proto,$time, $a_w, $peer_ip, $peer_AS, $prefix, $AS_path, $origin, $next_hop, $local_pref, $MED,
$community, $atomic_aggregate, $aggregator) = split /\|/, $ligne;


## transformation du format de la date
## des corrections sont apportées pour le mois et l'année

my ($sec, $min, $heure, $day, $mon,$year, $wday, $yday, $isdst)=localtime ($time);

$mon++;             #correction des nombres attribués aux mois,
                    #en attribuant une valeur de 1 à janvier, au lieu de 0
$year-=100;         #$year = nombre d'années depuis 1900
                    #année en cours= $year+1900
                    #représentation des années sur 2 chiffres=$year-2000



## Impression dans le fichier de sortie du contenu
##  des différents champs, accompagnés de leur description textuelle
## Formatage de la date
## Les messages BGP sont séparés par une ligne vide
## Chaque message BGP contient un seul préfixe. Il n'est pas prévu de
##  regrouper les préfixes, même s'ils partagent des attributs communs
##
## L'attribut AGGREGATOR doit être reconstitué sur base
##  - du contenu du champ AGGREGATOR (adresse IP du routeur qui a formé
##          l'agrégat)
##  - de l'information de l'AS-PATH. L'AS qui a formé l'agrégat est
##          - AS précédant un AS-SET (marqué par [])
##          - le dernier AS de l'AS-PATH (origine de la route agrégée)
printf F1 ("TIME: %02d/%02d/%02d %02d:%02d:%02d\n", $mon, $day, $year, $heure, $min,$sec);
print F1 "TYPE: BGP\/UPDATE\n";
print F1 "FROM: $peer_ip AS$peer_AS\n";
print F1 "TO: $to_IP AS$to_AS\n";

if ($a_w eq "W") {print F1 "WITHDRAW\n\t$prefix\n";}
    else {
        print F1 "ORIGIN: $origin\n";
        print F1 "ASPATH: $AS_path\n";
        print F1 "NEXT_HOP: $next_hop\n";
        if ($MED) { print F1 "MULTI_EXIT_DISC: $MED\n"};
        if ($local_pref) { print F1 "LOCAL_PREF: $local_pref\n"};
        if ($community) {print F1 "COMMUNITY: $community\n"};

        if ($aggregator) {
            @AS = split /\s+/, $AS_path;
            $cand = shift @AS;
            $_ = shift @AS;
            while (($_) && ( /^[\d]+/)){$cand=$_; $_ = shift @AS; }

            if ($atomic_aggregate eq "AG") {
                print F1 "ATOMIC_AGGREGATE\nAGGREGATOR:";
                print F1 " AS$cand $aggregator\n"
```

```
                } else {
                        print F1 "AGGREGATOR: AS$cand $aggregator\n"
                }
        };#end aggregator

        print F1 "ANNOUNCE\n\t$prefix\n\n";
    }
};

close F1;
close DF;
```

G. grep_capture

```
#!/usr/bin/perl -w

## grep_capture by Christine Vandesteene
##
## Objectif : extraire les informations qui permettent de caractériser
##            un message BGP
##
## Remarque : étant donné que chaque préfixe peut être traité
##            individuellement par le routeur,
##            les informations seront consignées pour chaque préfixe
##
## Usage :    perl grep_capture <file>
##
## Input :    fichier de trames capturées par l'analyseur de trafic tethereal
##
## Output :   toutes les informations concernant un préfixe individuel sont
##            consignées sur une ligne. Les informations relatives aux
##            messages OPEN, KEEPALIVE et NOTIFICATION sont également
##            conservées.
##
##          chaque ligne comprend plusieurs champs séparés par |
##                      - Type de message
##                      - Numéro de la trame
##                      - Comptabilisation du nombre d'UPDATES par trame
##                      - Adresse IP source
##                      - Adresse IP destination
##                      - Moment de la capture (relatif au premier paquet)
##                      - Annonce ou retrait
##                      - Préfixe
##                      - ORIGIN
##                      - AS-PATH
##                      - NEXT-HOP
##                      - COMMUNITY
##                      - Indique si la trame est mal-formée
##
## Version
##
## 0.02 [2002-06-02] : ajout des commentaires


my $name = $ARGV[0];

## Ouverture du fichier qui recevra les informations pertinentes
## extraites d'un fichier généré par l'analyseur de trafic tethereal

open (F,"> $name.t") or die "Problème : $!";


## Lecture ligne par ligne de l'entrée (fichier généré par tethereal)

while ($ligne=<>){


## Reconnaissance d'une nouvelle trame
## Initialisation du tableau correspondant à une ligne d'information
## à imprimer en sortie
## $i permet de calculer le nombre de messages BGP UPDATE
## contenus dans chaque trame
## Remarque : on ne calcule pas le nombre de préfixes annoncés ou retirés
##            dans un même message BGP

if ($ligne=~ /^Frame ([0-9]+)\s\(.*\)/){
      $i=0;
      @message=("","","","","","","","","","","","","");
      $message[1]=$1;
}
```

```
## Extraction du moment de la capture de la trame analysée

if ($ligne=~ /Time relative.*\s(\d+\.\d+) seconds/ ){
      $message[5]=$1;
}


## Extraction des informations relatives au protocole IP
##   - Adresse IP de la source
##   - Adresse IP de la destination
## La reconnaissance du format décimal pointé s'impose
## pour éviter la confusion avec les informations de source
## et de destination incluses dans la couche Ethernet

if ($ligne=~ /Source\: ([0-9]+\.[0-9]+\.[0-9]+\.[0-9]+).*/){
      $message[3]=$1;
}

if ($ligne=~ /Destination\: ([0-9]+\.[0-9]+\.[0-9]+\.[0-9]+).*/ ){
      $message[4]=$1;
}



## Détermination du type de message BGP
## Prise en compte de l'information relative
## aux trames mal-formées
## Une remise à 0 des champs propres aux UPDATE (annonces)
## est nécessaire pour les messages KEEPALIVE et UPDATE
## dans les cas où plusieurs messages BGP
## sont inclus dans une même trame Ethernet


if ($ligne=~ /^\s*(KEEPALIVE) Message/ ) {
      $message[0]=$1;
      @message[6..11]=( "","","","","","");
      print F join "|", @message ,"\n";
}

if ($ligne=~ /^\s*(OPEN) Message/ ) {
      $message[0]=$1;
      print F join "|", @message ,"\n";
}

if ($ligne=~ /(Malformed Frame).*/){
      $message[11]=$1;
      print F join "|", @message ,"\n";
}

if ($ligne=~ /^\s*(NOTIFICATION).*/ ) {
      $message[0]=$1;
      print F join "|", @message ,"\n";
}

if ($ligne=~ /^\s*(UPDATE).*/ ){
      $i++;
      @message[7..10]=("","","","");
      $message[2]=$i;
      $message[0]=$1;
      }


## Information de retrait de route

if ($ligne=~ /^\s*Unfeasible routes length: ([\d]+) bytes/ ) {
      if ($1!=0){$message[6]="W";}
}


## Attributs de chemin retenus pour caractériser une annonce
## Lorsque plusieurs préfixes sont annoncés dans un même message BGP
```

```
## ils partagent les mêmes attributs
##  - ORIGIN
##          - AS-PATH
##          - NEXT-HOP
##          - COMMUNITY

if ($ligne=~ /^\s*ORIGIN: (.*) \(\d+ bytes\)/ ) {
    $message[8]=$1;
}

if ($ligne=~ /^\s*AS_PATH: (.*) \(\d+ bytes\)/) {
    $message[9]=$1;
}

if ($ligne=~ /^\s*NEXT_HOP: (.*) \(\d+ bytes\)/) {
    $message[10]=$1;
}

if ($ligne=~ /^\s*COMMUNITIES: (.*) \(\d+ bytes\)/) {
    $message[11]=$1;
}


## Reconnaissance d'une annonce de préfixe

if ($ligne=~ /^\s*Network layer reach/ ) {
    $message[6]="A";
}


## Reconnaissance de chacun des préfixes contenus dans le message BGP
## Une ligne est imprimée dans le fichier de sortie pour
## chacun des préfixes rencontrés

if ($ligne=~ /([0-9]+\.[0-9]+\.[0-9]+\.[0-9]+\/[0-9]+)/ ) {
    $message[7]=$1;
    print F join "|", @message ,"\n";
}

} ## end while

close F;
```

H. calcule

```
#!/usr/bin/perl

## Calcule by Christine Vandesteene
##
## Objectif : calculer l'intervalle de temps entre les annonces
##            entrantes et sortantes d'un même préfixe
##
## Remarque : les calculs sont effectués sur base de l'adresse IP
##            de destination du message, aussi bien en amont
##            qu'en aval du DUT.
##            Si toutes les sessions BGP se font sur des réseaux
##            distincts (physiques ou virtuels), il est possible
##            de contrôler pour quelle session en amont on souhaite
##            effectuer les calculs de convergence sur base du destinataire.
##            Dans le cas contraire, la convergence sera calculée
##            pour la dernière annonce entrée.
##            Dans ce cas, des interférences peuvent être évitées
##            si les différents pairs annoncent leurs routes
##            à des moments différents
##
## Usage :    perl calcule -h -u <upstream> -d <downstream> <input> > <output>
##                    -h : help
##                    -u : destination ip address of upstream bgp messages
##                    -d : destination ip address of downstream bgp messages
##
## Input :    fichier contenant les informations les plus pertinentes
##            à propos des mises à jour de routage. Ce fichier
##            est obtenu par l'application du script "grep_capture"
##            sur un fichier de paquets BGP capturés avec l'analyseur
##            de trafic tethereal
##
##            Les informations qui nous intéressent sont le préfixe,
##            la destination et le moment de la capture
##
## Output :   fichier contenant
##                    - préfixe
##                    - moment de capture à l'entrée (upstream)
##                    - moment de capture à la sortie (downstream)
##                    - temps de convergence (calculé)
##
## Version
##
## 0.01 [2002-06-01]
## 0.01 [2002-06-29]
## 0.01 [2002-06-30]
## 0.02 [2002-11-16] : ajout de commentaires



## Module et instructions permettant de prendre des paramètres en option
## Fixation des paramètres par défaut (sur base du dispositif de test)

use Getopt::Std;

my %options=(
      "u"=> "157.164.170.2", #ip destination for upstream test updates
      "d"=> "10.0.0.1", #ip destination for downstream updates
);

getopts("hu:d:", \%options);

if ($options{h}) {
      die "usage : perl calcule -h -u <upstream> -d <downstream> \n
\t-h : help
\t-u : destination ip address of upstream bgp messages
\t-d : destination ip address of downstream bgp messages
";}
```

```
my $upstream = $options{u}; ## destination des messages entrants
my $downstream=$options{d}; ## destination des messages sortants


## Lecture ligne par ligne du contenu du fichier en entrée
## Division de chaque lignes en plusieurs champs sur base du séparateur |

while ($ligne=<>){

my ($type, $frame_nr, $update_nr, $ip_src, $ip_dst, $time, $a_w, $prefix, $origin, $as_path, $next_hop,
$community)= split /\|/, $ligne;


## Crée des tables de hashage avec le préfixe comme clé (unique)
##   et le temps comme valeur.
##   Si la clé n'est pas unique (plusieurs représentations du
##      même préfixe), la dernière valeur de temps rencontrée
##   sera retenue.
##
## Trois tables de hashage sont créées
##   - Annonces en amont du DUT (%upstream_announce)
##   - Retraits en amont du DUT (%upstream_withdraw)
##   - Annonces et retraits en aval du DUT (%downstream)
##
## Les retraits de routes n'ont pas d'effet si le préfixe n'a
##   pas été annoncé auparavant. C'est pourquoi il est
##   nécessaire de distinguer les annonces des retraits
##   en amont du DUT.
##   Etant donné que notre modèle suppose que la table de routage
##   du DUT a été initialisée, tout retrait de route en amont du DUT
##   sera suivi de l'annonce d'une route moins bonne en aval
##

if ($ip_dst eq $upstream && $type eq "UPDATE"&& $a_w eq "A") {$upstream_announce{$prefix}=$time;}

if ($ip_dst eq $upstream && $type eq "UPDATE"&& $a_w eq "W") {$upstream_withdraw {$prefix}=$time;}

if ($ip_dst eq $downstream && $type eq "UPDATE") {$downstream{$prefix}=$time;}
}


## Calcul de la convergence et affichage des résultats

if (!keys %upstream_withdraw){
print "ANNOUNCES\n";
print "PREFIX\t\tUPSTREAM\tDOWSTREAM\tCONV_TIME\n";
foreach $prefix (sort keys %upstream_announce){
$diff=$downstream{$prefix}-$upstream_announce{$prefix};

printf "%s\t%10.6f  %10.6f  %10.6f\n",$prefix,$upstream_announce{$prefix},$downstream{$prefix},$diff;
}


}else{

print "WITHDRAWS\n";
print "PREFIX\t\tUPSTREAM\tDOWNSTREAM\tCONV_TIME\n";
foreach $prefix (sort keys %upstream_withdraw){
                $diff=$downstream{$prefix}-$upstream_withdraw{$prefix};
                printf "%s\t%10.6f  %10.6f
%10.6f\n",$prefix,$upstream_withdraw{$prefix},$downstream{$prefix},$diff;
}
}
```

I. Configuration de base des routeurs

Cisco1 :

```
interface Ethernet1/2
 description Connexion vers le hub central H8
 ip address 10.0.0.1 255.255.255.0

router bgp 100
 bgp log-neighbor-changes
 timers bgp 3600 10800
 neighbor 10.0.0.2 remote-as 200
```

Cisco2 :

```
interface Loopback0
 ip address 212.20.151.2 255.255.255.0
interface Loopback1
 ip address 157.164.170.2 255.255.255.0
interface Ethernet1/2
 description Connexion vers le hub central H8
 ip address 10.0.0.2 255.255.255.0

router bgp 200
 bgp log-neighbor-changes
 timers bgp 3600 10800
 neighbor 10.0.0.1 remote-as 100
 neighbor 10.0.0.1 advertisement-interval 0
 neighbor 157.164.170.3 remote-as 65000
 neighbor 157.164.170.3 update-source
Loopback1
 neighbor 157.164.170.3 route-map Filter out
 neighbor 212.20.151.4 remote-as 65000
 neighbor 212.20.151.4 update-source
Loopback0
 neighbor 212.20.151.4 route-map Filter out
no ip classless
ip route 157.164.170.3 255.255.255.255
10.0.0.3
ip route 212.20.151.4 255.255.255.255
10.0.0.4
```

Cisco3 :

```
interface Loopback0
 ip address 157.164.170.3 255.255.255.0
interface Ethernet1/2
ip address 10.0.0.3 255.255.255.0

router bgp 65000
 no synchronization
 bgp log-neighbor-changes
 timers bgp 3600 10800
 neighbor 157.164.170.2 remote-as 200
 neighbor 157.164.170.2 update-source
Loopback0
 neighbor 157.164.170.2 send-community
 neighbor 157.164.170.2 advertisement-
interval 0
 neighbor 157.164.170.2 route-map
COMMUNITY out
 neighbor 157.164.170.4 remote-as 65000
 neighbor 157.164.170.4 update-source
Loopback0
ip route 157.164.170.2 255.255.255.255
10.0.0.2
ip route 157.164.170.4 255.255.255.255
10.0.0.14
ip as-path access-list 1 permit 10.0.0.0
0.255.255.255
ip pim bidir-enable
route-map COMMUNITY permit 10
 match ip address 1
 set community 1415
```

Cisco4 :

```
interface Loopback0
 ip address 212.20.151.4 255.255.255.0
interface Ethernet1/2
 description Connexion vers le hub central H8
 ip address 10.0.0.4 255.255.255.0

router bgp 65000
 no synchronization
 bgp log-neighbor-changes
 timers bgp 3600 10800
 neighbor 212.20.151.2 remote-as 200
 neighbor 212.20.151.2 update-source
Loopback0
 neighbor 212.20.151.234 remote-as 65000
 neighbor 212.20.151.234 update-source
Loopback0
ip route 212.20.151.2 255.255.255.255
10.0.0.2
ip route 212.20.151.234 255.255.255.255
10.0.0.14
```

Les configurations des routeurs sont sauvegardées sur PC4 au moyen de cisconf.

J. Séquence d'événements avec "aggregate-address as-set"

Examen de la sequence d'événement : annonce du préfixe spécifique par le pair en amont du DUT (IP-source=157.164.170.3), annonce du préfixe spécifique par le DUT (IP-source=10.0.0.2) avec ou sans annonce de l'agrégat. Dans certains cas, l'agrégat est annoncé avant le préfixe spécifique, dans d'autres après.

TEST : 2002-08-08.aggr-ebgp1-a.amas.log
Agrégat : 10.3.32.0/21

```
UPDATE|872|1|157.164.170.3|157.164.170.2|426.118034|A|10.3.38.0/24|IGP|65000 65000 64999 65039
65005 |157.164.170.4|0:1415|
UPDATE|873|1|10.0.0.2|10.0.0.1|426.124303|A|10.3.38.0/24|IGP|200 65000 65000 64999 65039 65005
|10.0.0.2||
UPDATE|877|1|10.0.0.2|10.0.0.1|426.323042|A|10.3.32.0/21|IGP||10.0.0.2||

UPDATE|1207|1|157.164.170.3|157.164.170.2|474.118933|A|10.3.35.0/24|IGP|65000    65000    64999
65036 65005 |157.164.170.4|0:1415|
UPDATE|1208|1|10.0.0.2|10.0.0.1|474.125198|A|10.3.35.0/24|IGP|200   65000   65000   64999   65036
65005 |10.0.0.2||
UPDATE|1210|1|10.0.0.2|10.0.0.1|474.127462|A|10.3.32.0/21|IGP||10.0.0.2||

UPDATE|1869|1|157.164.170.3|157.164.170.2|570.120690|A|10.3.34.0/24|IGP|65000    65000    64999
65035 65005 |157.164.170.4|0:1415|
UPDATE|1870|1|10.0.0.2|10.0.0.1|570.126950|A|10.3.32.0/21|IGP||10.0.0.2||
UPDATE|1874|1|10.0.0.2|10.0.0.1|570.325835|A|10.3.34.0/24|IGP|200   65000   65000   64999   65035
65005 |10.0.0.2||

UPDATE|1965|1|157.164.170.3|157.164.170.2|584.121002|A|10.3.39.0/24|IGP|65000    65000    64999
65040 65005 |157.164.170.4|0:1415|
UPDATE|1966|1|10.0.0.2|10.0.0.1|584.127051|A|10.3.39.0/24|IGP|200   65000   65000   64999   65040
65005 |10.0.0.2||

UPDATE|2940|1|157.164.170.3|157.164.170.2|729.123760|A|10.3.36.0/24|IGP|65000    65000    64999
65037 65005 |157.164.170.4|0:1415|
UPDATE|2941

UPDATE|3533|1|157.164.170.3|157.164.170.2|819.125346|A|10.3.32.0/24|IGP|65000    65000    64999
65033 65005 |157.164.170.4|0:1415|
UPDATE|3534|1|10.0.0.2|10.0.0.1|819.131601|A|10.3.32.0/24|IGP|200   65000   65000   64999   65033
65005 |10.0.0.2||
UPDATE|3538|1|10.0.0.2|10.0.0.1|819.333665|A|10.3.32.0/21|IGP||10.0.0.2||

UPDATE|5138|1|157.164.170.3|157.164.170.2|1060.129805|A|10.3.33.0/24|IGP|65000    65000    64999
65034 65005 |157.164.170.4|0:1415|
UPDATE|5139|1|10.0.0.2|10.0.0.1|1060.136108|A|10.3.32.0/21|IGP||10.0.0.2||
UPDATE|5143|1|10.0.0.2|10.0.0.1|1060.337033|A|10.3.33.0/24|IGP|200   65000   65000   64999   65034
65005 |10.0.0.2||

UPDATE|6453|1|157.164.170.3|157.164.170.2|1256.133375|A|10.3.37.0/24|IGP|65000    65000    64999
65038 65005 |157.164.170.4|0:1415|
UPDATE|6454|1|10.0.0.2|10.0.0.1|1256.139623|A|10.3.37.0/24|IGP|200   65000   65000   64999   65038
65005 |10.0.0.2||
UPDATE|6458|1|10.0.0.2|10.0.0.1|1256.338291|A|10.3.32.0/21|IGP||10.0.0.2||
```

K. Quantification des processus internes

	Annonces, voisin EBGP					Annonces, voisin IBGP					Retraits, voisin EBGP					Retraits, voisin IBGP			
	WEIGHT	AS-PATH	ORIGIN	MED	RID	WEIGHT	AS-PATH	ORIGIN	MED	RID	WEIGHT	AS-PATH	ORIGIN	MED	RID	WEIGHT	AS-PATH	ORIGIN	MED
Phase 0																			
0.1. Non déterminé	2,48	2,48	2,48	2,48	2,48	2,48	2,48	2,48	2,48	2,48	2,48	2,48	2,48	2,48	2,48	2,48	2,48	2,48	2,48
0.2. Variation expérimentale																			
Phase 1																			
1.1.Calcul degré de préférence	0,69	0,69	0,69	0,69	0,69	0,69	0,69	0,69	0,69	0,69	0,00	0,00	0,00	0,00	0,00				
1.2. Politique d'entrée																			
1.2.0. Non attribué																			
1.2.1 Filtrage																			
1.2.1.1. Examen clause match																			
1.2.1.2. Nombre de clauses match [1,5,25]																			
Sous-total Filtrage																			
1.2.2. Manipulation d'attribut																			
1.2.2.1. Application clause set																			
Phase 2																			
2.1. Application règles d'arbitrage																			
2.1.1 Comparaison AS émetteur	0	0	0	0	0	0	0	0	0,32	0	0	0	0	0	0	0	0	0	0
Phase 3																			
3.1Modification AS-PATH avant redistribution	0,10	0,10	0,10	0,10	0,10	0	0	0	0	0	0,10	0,10	0,10	0,10	0,10				
3.2. Politique de sortie																			
3.2.0. Non attribué																			
3.2.1 Filtrage																			
3.2.1.1. Examen clause match																			
3.2.1.2. Nombre de clauses match [1,5,25]																			
Sous-total filtrage																			
3.2.2. Manipulation d'attribut																			
3.2.2.1. Application clause set																			
3.3 Agrégation																			
3.3.0. Non déterminé																			
3.3.1. Vérifier si préfixe doit être agrégé																			
3.3.2. Adaptation as-path agrégat																			
3.3.3. Distr.optionnelle 2e message (préf ou agrégat)																			
3.4. Comparaison réseaux pairs Upstr et Dow nstr	0,08	0,08	0,08	0,08	0,08	*0,08*	*0,08*	*0,08*	*0,08*	*0,08*	0,08	0,08	0,08	0,08	0,08	*0,08*	*0,08*	*0,08*	*0,08*
3.5. Calcul du hash (MD5)																			
3.6. Emission d'un message à la console (au-delà du seuil)																			
Total	**3,35**	**3,35**	**3,35**	**3,65**	**3,35**	**3,25**	**3,25**	**3,25**	**3,57**	**3,25**	**2,66**	**2,66**	**2,66**	**2,96**	**2,66**	**2,56**	**2,56**	**2,56**	**2,86**
Valeurs expérimentales	***3,34***	***3,34***	***3,36***	***3,67***	***3,38***	***3,26***	***3,26***	***3,28***	***3,56***	***3,28***	***2,69***	***2,70***	***2,68***	***2,97***	***2,67***	***2,58***	***2,59***	***2,58***	***2,86***

	Tests de base					*Policy (AS-PATH)*				*Agrégation*				*Paramètres BGP*		
	WEIGHT	AS-PATH	ORIGIN	MED	RID	ACC-IN	MOD-IN	ACC-OUT	MOD-OUT	AG-21	AG-21, as-set	AG-16	AG-21, other	NHS	PW	MPWO
Phase 0																
0.1. Non déterminé	2,48	2,48	2,48	2,48	2,48	2,48	2,48	2,48	2,48	2,48	2,48	2,48	2,48	2,48	2,48	2,48
0.2. Variation expérimentale						0,03	0,03	0,03	0,03	0,03	0,03	0,03	0,03	0,03	0,03	0,03
Phase 1																
1.1.Calcul degré de préférence	0,69	0,69	0,69	0,69	0,69	0,69	0,69	0,69	0,69	0,69	0,69	0,69	0,69	0,69	0,69	0,69
1.2. Politique d'entrée																
1.2.0. Non attribué						0,09	0,09									
1.2.1 Filtrage																
1.2.1.1. Examen clause match						0,03	0,03									
1.2.1.2. Nombre de clauses match [1,5,25]						5	5									
Sous-total Filtrage						***0,16***	***0,16***									
1.2.2. Manipulation d'attribut																
1.2.2.1. Application clause set						0	0,08									
Phase 2																
2.1. Application règles d'arbitrage																
2.1.1 Comparaison AS émetteur	0	0	0	0	0											
Phase 3																
3.1Modification AS-PATH avant redistribution	0,10	0,10	0,10	0,10	0,10	0,10	0,10	0,10	0,10	0,10	0,10	0,10	0,10	0,10	0,10	0,10
3.2. Politique de sortie																
3.2.0. Non attribué								0,20	0,20							
3.2.1 Filtrage																
3.2.1.1. Examen clause match								0,03	0,03							
3.2.1.2. Nombre de clauses match [1,5,25]								5	5							
Sous-total filtrage								***0,15***	***0,15***							
3.2.2. Manipulation d'attribut																
3.2.2.1. Application clause set								0	0							
3.3 Agrégation												6,15				
3.3.0. Non déterminé										2,04	2,04					
3.3.1. Vérifier si préfixe doit être agrégé										0,20	0,20		0,20			
3.3.2. Adaptation as-path agrégat											0,68					
3.3.3. *Distr.optionnelle 2e message (préf ou agrégat)*											*204,00*					
3.4. Comparaison réseaux pairs Upstr et Downstr	0,08	0,08	0,08	0,08	0,08	0,08	0,08	0,08	0,08	0,08	0,08	0,08	0,08	0	0,08	0,08
3.5. Calcul du hash (MD5)															0,18	
3.6. Emission d'un message à la console (au-delà du seuil)																0,20
Total	**3,35**	**3,35**	**3,35**	**3,65**	**3,35**	**3,63**	**3,71**	**3,72**	**3,72**	**5,62**	**6,30**	**9,53**	**3,58**	**3,30**	**3,56**	**3,58**
Valeurs expérimentales	***3,34***	***3,34***	***3,36***	***3,67***	***3,38***	***3,63***	***3,72***	***3,74***	***3,74***	***5,62***		***9,53***	***3,58***	***3,30***	***3,56***	***3,55***

	3 pairs - Sans PEER-GROUP											
	First				**Second**				**Third**			
	Ref	**RM-OUT**	**AG-21**	**AG-16**	**Ref**	**RM-OUT**	**AG-21**	**AG-16**	**Ref**	**RM-OUT**	**AG-21**	**AG-16**
Phase 0												
0.1. Non réparti (référence)	3,39	3,39	3,39	3,39	3,39	3,39	3,39	3,39	3,39	3,39	3,39	3,39
Phase 1												
1.1.Calcul degré de préférence												
1.2. Politique d'entrée												
1.2.1 Filtrage												
1.2.2. Manipulation d'attribut												
Phase 2												
2.1. Application règles d'arbitrage												
Phase 3												
3.1Modification AS-PATH avant redistribution												
3.2. Politique de sortie		0,23				0,23				0,23		
3.2.1 Filtrage												
3.2.2. Manipulation d'attribut												
3.3 Agrégation			2,24	6,15			2,24	6,15			2,24	6,15
3.4. Comparaison réseaux pairs Upstr et Dow nstr												
3.5. Calcul du hash (MD5)												
3.6. Emission d'un message à la console (au-delà du seuil)												
3.7. Effet de l'augmentation du nombre de pairs												
3.7.1. Facteur "configuration"												
3.7.1.1. Coefficient par pair	0,38	0,38	0,38	0,38	0,38	0,38	0,38	0,38	0,38	0,38	0,38	0,38
3.7.1.2. Nombre de pairs	3	3	3	3	3	3	3	3	3	3	3	3
Sous-total	***0,76***	***0,76***	***0,76***	***0,76***	***0,76***	***0,76***	***0,76***	***0,76***	***0,76***	***0,76***	***0,76***	***0,76***
3.7.2. Facteur "pair"												
3.7.2.1. Coeff 1e pair (0)	0,00	0,00	0,00	0,00								
3.7.2.1. Coeff 2e pair (0.492)					0,49	0,49	0,49	0,49				
3.7.2.1. Coeff 1e pair (1.272)									1,27	1,27	1,27	1,27
3.7.3. Facteur "politique spécifique" (route-map)												
3.7.3.1. Pour trois pairs		0,39				0,39				0,39		
3.8. Effet de la commande peer-group												
3.8.1. Diminution "politique globale" avec deux pairs (-0.26)												
3.8.2. Diminution "politique globale" avec trois pairs (-0.59)												
3.8.2. Diminution "politique spécifique" avec trois pairs (-0.93)												
Total	**4,15**	**4,77**	**6,39**	**10,30**	**4,65**	**5,26**	**6,89**	**10,80**	**5,43**	**6,04**	**7,67**	**11,58**
Valeurs expérimentales	***4,19***	***4,76***	***6,43***	***10,33***	***4,67***	***5,26***	***6,92***	***10,81***	***5,45***	***6,05***	***7,69***	***11,59***

	3 pairs - Avec PEER-GROUP											
	First				**Second**				**Third**			
	Ref	**RM-OUT**	**AG-21**	**AG-16**	**Ref**	**RM-OUT**	**AG-21**	**AG-16**	**Ref**	**RM-OUT**	**AG-21**	**AG-16**
Phase 0												
0.1. Non réparti (référence)	3,39	3,39	3,39	3,39	3,39	3,39	3,39	3,39	3,39	3,39	3,39	3,39
Phase 1												
1.1.Calcul degré de préférence												
1.2. Politique d'entrée												
1.2.1 Filtrage												
1.2.2. Manipulation d'attribut												
Phase 2												
2.1. Application règles d'arbitrage												
Phase 3												
3.1Modification AS-PATH avant redistribution												
3.2. Politique de sortie		0,23				0,23				0,23		
3.2.1 Filtrage												
3.2.2. Manipulation d'attribut												
3.3 Agrégation			2,24	6,15			2,24	6,15			2,24	6,15
3.4. Comparaison réseaux pairs Upstr et Dow nstr												
3.5. Calcul du hash (MD5)												
3.6. Emission d'un message à la console (au-delà du seuil)												
3.7. Effet de l'augmentation du nombre de pairs												
3.7.1. Facteur "configuration"												
3.7.1.1. Coefficient par pair	0,38	0,38	0,38	0,38	0,38	0,38	0,38	0,38	0,38	0,38	0,38	0,38
3.7.1.2. Nombre de pairs	3	3	3	3	3	3	3	3	3	3	3	3
Sous-total	***0,76***	***0,76***	***0,76***	***0,76***	***0,76***	***0,76***	***0,76***	***0,76***	***0,76***	***0,76***	***0,76***	***0,76***
3.7.2. Facteur "pair"												
3.7.2.1. Coeff 1e pair (0)	0,00	0,00	0,00	0,00								
3.7.2.1. Coeff 2e pair (0.492)					0,49	0,49	0,49	0,49				
3.7.2.1. Coeff 1e pair (1.272)									1,27	1,27	1,27	1,27
3.7.3. Facteur "politique spécifique" (route-map)												
3.7.3.1. Pour trois pairs		0,39				0,39				0,39		
3.8. Effet de la commande peer-group												
3.8.1. Diminution "politique globale" avec deux pairs (-0.26)												
3.8.2. Diminution "politique globale" avec trois pairs (-0.59)	-0,59		-0,59	-0,59	-0,59		-0,59	-0,59	-0,59		-0,59	-0,59
3.8.2. Diminution "politique spécifique" avec trois pairs (-0.93)		-0,93				-0,93				-0,93		
Total	**3,56**	**3,84**	**5,80**	**9,71**	**4,06**	**4,33**	**6,30**	**10,21**	**4,84**	**5,11**	**7,08**	**10,99**
Valeurs expérimentales	***3,57***	***3,83***	***5,80***	***9,75***	***4,06***	***4,33***	***6,29***	***10,23***	***4,85***	***5,12***	***7,06***	***11,01***