

ESSI 3 – Systèmes et Applications Réparties  
DEA Réseaux et Systèmes Distribués

# Conception et Réalisation d'un Protocole d'Interconnexion de CDN

Fabien Germain  
<*Fabien.Germain@ActiVia.net*>

4 septembre 2001

Effectué au sein de la société *ActiVia Networks*

Ecole Supérieure en Sciences Informatiques – Université de Nice Sophia Antipolis









## Table des matières

<b>1</b>	<b>Coordonnées</b>	<b>7</b>
<b>2</b>	<b>Sujet de stage</b>	<b>8</b>
<b>3</b>	<b>Introduction</b>	<b>9</b>
3.1	Présentation du problème . . . . .	9
3.2	La société ActiVia Networks . . . . .	10
3.2.1	Ses origines . . . . .	10
3.2.2	L'environnement . . . . .	11
<b>4</b>	<b>Environnement du stage</b>	<b>13</b>
4.1	Qu'est ce qu'un CDN ? . . . . .	13
4.2	L'intérêt du CDN . . . . .	14
4.3	Interconnexion des CDN ( <i>content peering</i> ) . . . . .	14
4.4	La solution proposée par ActiVia . . . . .	15
<b>5</b>	<b>Objectifs scientifiques</b>	<b>17</b>
<b>6</b>	<b>Plan de travail</b>	<b>18</b>
<b>7</b>	<b>Outils utilisés</b>	<b>19</b>
<b>8</b>	<b>Travail réalisé</b>	<b>20</b>
8.1	Interconnexion de CDN . . . . .	20
8.2	Test d'interconnexion avec AT&T . . . . .	21
8.3	Protocole de Content Peering . . . . .	23
8.3.1	“Request Routing Peering Protocol” . . . . .	23
8.3.2	Implémentation du protocole . . . . .	24
<b>9</b>	<b>Conclusion</b>	<b>27</b>
<b>A</b>	<b>Request Routing Peering Protocol</b>	<b>29</b>
<b>B</b>	<b>Exemple de script Perl</b>	<b>52</b>
	<b>Références</b>	<b>57</b>



# 1 Coordonnées

Ce travail a été réalisé au sein de la société **ActiVia Networks** à Sophia Antipolis, dans l'équipe de recherche et développement.

Les responsables de ce projet sont **Delphine Kaplan** (recherche) et **Martin May** (production) :

ActiVia Networks  
Space Antipolis 5  
2323, chemin Saint Bernard  
06225 Vallauris Cedex

Delphine Kaplan : *kaplan@activia.net*  
+33 (0)4 97 23 46 64

Martin May : *mmay@activia.net*  
+33 (0)4 97 23 46 52

Mes coordonnées personnelles sont :

Fabien Germain  
26, rue Louis Pasteur  
Ecole maternelle Louis Pasteur  
76120 Le Grand-Quevilly

*Fabien.Germain@ActiVia.net*  
+33 (0)6 22 091 291

## 2 Sujet de stage

### Conception et Réalisation d'un Protocole d'Interconnexion de CDN

Face à la croissance explosive en taille de l'Internet, les caches fleurissent depuis plusieurs années déjà en bordure du réseau. Les caches sont faciles à déployer et satisfont à la fois les opérateurs réseaux qui économisent leur bande passante, et les utilisateurs qui réduisent leurs temps d'attente.

Plus récemment les **réseaux d'acheminement de contenus** (CDN) ont vu le jour ainsi qu'un certain nombre d'outils qui permettent de les construire et de les gérer. Un CDN est une architecture d'éléments réseau basée sur la technologie des caches, chargée d'acheminer de manière efficace du contenu numérique (pages Web statiques ou dynamiques, flux continus audio-visuels en direct ou à la demande).

Plus précisément un CDN contient un ensemble de serveurs "représentants" qui possèdent une copie du contenu à livrer, et met en œuvre trois systèmes :

1. le système de redirection qui redirige une requête utilisateur vers le serveur représentant optimum,
2. le système de distribution qui déplace le contenu du serveur origine vers ses représentants,
3. le système de facturation chargé, entre autre, d'évaluer les coûts et d'établir une tarification.

Devant le nombre croissant de CDN, se pose aujourd'hui le problème de leur interconnexion pour un meilleur service global. Il est en effet crucial qu'un fournisseur de contenu, qui sous-traite la distribution/livraison de son contenu à un certain CDN A, ait la possibilité d'augmenter son audience en étant distribué par d'autres CDN "pairs" du CDN A. L'interopérabilité des CDN passera bien sûr par l'interopérabilité des trois composantes ci-dessus.

Il s'agit durant le stage de participer à la spécification d'un protocole d'interconnexion entre les systèmes de redirection de différents CDN et d'implémenter un prototype de ce protocole en C.

Ce travail sera réalisé au sein de l'équipe de recherche et de développement d'ActiVia Networks. ActiVia Networks est une société qui a été fondée en avril 2000 avec le soutien de l'INRIA. Elle est membre de l'initiative Content Alliance ([www.content-alliance.org](http://www.content-alliance.org)) et participe pleinement à l'effort de standardisation sur l'interconnexion des CDN, à travers l'écriture de drafts Internet dans le groupe IETF CDI - Content Distribution Internetworking.

## 3 Introduction

### 3.1 Présentation du problème

Technologie critique pour la large diffusion des services d’acheminement de contenus, le *content peering*<sup>1</sup> permet aux CDN de différents fournisseurs de services de travailler ensemble en s’échangeant leurs contenus respectifs.

Trois types d’acteurs interviennent dans le système du CDN : Le fournisseur de contenu (l’utilisateur du CDN), l’opérateur réseaux (backbone, FAI, hébergeur) et l’utilisateur final (le client à satisfaire, celui qui profite de la qualité de service fournie par le CDN).

Les réseaux d’acheminement de contenus (*Content Distribution Networks*, CDN) accélèrent l’accès aux sites web et aux flux audios et vidéos en redirigeant les requêtes vers des “caches intelligents” situés à proximité de l’utilisateur. Ces caches sont en général situés chez les opérateurs réseaux.

Toutefois peu d’opérateurs réseaux peuvent se permettre d’avoir un CDN qui couvrirait toutes les régions du globe. De même, un CDN isolé ne peut pas fournir qualité de service et sécurité à grande échelle.

Etant donné le nombre de réseaux indépendants qui forment l’Internet actuel, l’interopérabilité des CDN permet de hautes performances pour la distribution des contenus web, stockés dans des caches proches des utilisateurs, et une couverture optimale.

Grâce à l’interconnexion des CDN, un site web peut travailler avec son hébergeur favori, tout en profitant des performances des réseaux interconnectés.

Afin de permettre un fonctionnement complet et cohérent, l’interconnexion des CDN nécessite le partage d’informations dans trois domaines différents :

1. *Distribution du contenu* : Copier les fichiers depuis la “source” sur les caches. La source réelle pouvant appartenir à un autre CDN, le terme “source” désigne ici la machine qui introduit le contenu à l’intérieur d’un CDN (il pourra donc venir de l’interconnexion avec un CDN voisin).
2. *Routage des requêtes* : Rediriger la requête d’un utilisateur vers le cache approprié “le plus proche”, c’est-à-dire le cache le plus apte à répondre dans les meilleures conditions (vitesse, bande passante, etc... selon le type de contenu demandé).

---

<sup>1</sup>Il n’y a pas de traduction réelle en français, “échange de contenu”. Ce terme désigne l’interconnexion des CDN

3. *Facturation* : Comptabiliser les accès, facturer la consultation de certains contenus, même si ils sont consultés par l'intermédiaire d'un CDN qui n'est pas autoritatif sur ceux-ci.

## 3.2 La société ActiVia Networks

ActiVia Networks est une société qui conçoit et fournit des solutions globales d'acheminement de contenus sur l'Internet. ActiVia Networks permet aux opérateurs backbone internationaux, aux opérateurs locaux, aux fournisseurs d'accès Internet, ainsi qu'aux entreprises de construire et de gérer leur propre réseau d'acheminement de contenus (CDN). Depuis sa création, en avril 2000, ActiVia Networks a constitué son équipe de direction, a lancé la première version de Constellation, sa solution CDN basée sur son produit phare, le routeur de contenu A\*Star, et a déployé des pilotes en Europe et aux Etats-Unis.

Comme exposé précédemment, ActiVia Networks s'appuie sur une technologie de pointe développée depuis 1997 à l'INRIA, Institut National de Recherche en Informatique et Automatique. Soutenu par un collège de scientifiques et de conseillers renommés, et contribuant à de nombreuses initiatives de standardisation, telles que Content Alliance et iCAP, ActiVia Networks conduit la prochaine génération de solutions d'acheminement de contenu Internet (CDN). ActiVia Networks compte parmi ses partenaires des sociétés telles que IBM, Network Appliance et CacheFlow.

ActiVia Networks dispose d'un siège social basé dans la "Telecom valley" européenne, Sophia Antipolis, France. C'est une société privée financée par Sofinnova Partners, Cross Atlantic Ventures et I Source, fonds d'amorçage de l'INRIA.

### 3.2.1 Ses origines

L'histoire d'ActiVia débute en 1997, mais la compagnie est créée en avril 2000 par cinq personnes. Trois d'entre elles, Frank Lyonnet, Laurent Gautier et Martin May, étaient membres du projet de recherche RODEO mis en place par l'INRIA. Le groupe a travaillé en étroite collaboration avec le World Wide Web Consortium (W3C) sur des développements web comme le "caching" et le "mirroring" et il a gagné une reconnaissance internationale pour ses travaux sur les technologies Internet.

ActiVia Networks est réellement créée lorsque Frank Lyonnet, Laurent Gautier et Martin May s'allient à Jean-François Abramatic (premier président du W3C) et Gérard Schreder (Directeur commercial et marketing Telenet Communication Corp, fondateur de Cap Sesa Conseil). Leur but commun

était de développer au-delà le travail de fond qu'ils avaient effectué à l'INRIA et de commercialiser ces technologies.

En fondant ActiVia Networks, Frank Lyonnet, Laurent Gautier et Martin May ont reçu l'appui d'institutions scientifiques importantes. L'INRIA les a soutenu dans la création de l'entreprise et continue aujourd'hui à lui apporter tout son soutien. Une partie du logiciel d'ActiVia est sous licence de l'INRIA.

De plus ActiVia Networks a récolté pour son premier tour de financement l'appui de capital risqueurs évoluant dans le domaine du High Tech : Sofinnova Partners, Cross Atlantic Ventures, I Source. A ce jour la société prépare sa seconde levée de fonds (prévue pour la mi-octobre) en sollicitant des investisseurs Nord Américains et Européens. Le montant de cette seconde levée de fonds devrait atteindre les dix voir quinze millions d'Euros.

Les fondateurs d'ActiVia Networks ont également remporté deux prix nationaux d'entrepreneuriat décernés par le Ministère de l'Education, de la Recherche et de l'Industrie.

### **3.2.2 L'environnement**

L'Internet souffre aujourd'hui d'une congestion importante résultant d'une croissance exponentielle du nombre d'utilisateurs et des nouvelles applications multimédia gourmandes en ressources réseaux et systèmes. Les fournisseurs d'accès à l'Internet parviennent avec peine à absorber la demande soutenue en terme de capacité de transport. La saturation de la bande passante, qui entraîne des pertes de données et donc de forts délais de transmission, affecte sensiblement les conditions de restitution des informations à l'utilisateur. Cette bande passante a un coût. Celui-ci est supporté par les utilisateurs (de manière directe ou indirecte) mais surtout par les fournisseurs de contenu. Ces derniers sont donc amenés à considérer de nouveaux services de transmission leur permettant d'optimiser la qualité de service rendue à leurs clients ainsi que leurs coûts.

On constate également un besoin grandissant de distribuer de nouvelles formes de contenus comme l'audio, et la vidéo (aussi bien en direct qu'à la demande) et ce besoin se confirme avec des applications comme la TV sur l'Internet, les applications distribuées interactives, etc...

Les opérateurs réseaux sont passés d'une simple vente de connexions Internet à la volonté d'offrir des solutions et services à valeur ajoutée pour augmenter l'efficacité de leur réseau tout en essayant de réduire les coûts. La capacité de fournir cette qualité de service à leurs clients est l'une de leurs plus grandes priorités.

Pour augmenter la qualité, des processeurs plus rapides et des serveurs ont été développés, la bande passante destinée au réseau a été augmentée et les

solutions de cache ainsi que la technologie des routeurs ont été améliorées. Toutes ces innovations ont aidé à améliorer les problèmes liés à la bande passante pour les opérateurs backbone, mais subsiste le problème du passage à l'échelle. Néanmoins ces options n'ont réussi qu'à résoudre seulement une partie du problème, et ce pour un coût élevé. De plus, ces solutions n'ont pas été suffisantes pour répondre à la créativité des fournisseurs de contenus.

Ainsi l'infrastructure d'Internet a vu apparaître une nouvelle ère technologique. Elle devra offrir une extension des capacités de l'ordinateur d'un point de vue technique et plus important encore elle devra fournir une meilleure qualité de service à l'utilisateur final. Des solutions sous la forme d'acheminement de contenus (CDNs) ont donc été mises en œuvre pour distribuer de nouveaux types de contenus via Internet, tout en assurant une qualité de service.

Les CDN placent des serveurs représentants près de l'utilisateur pour livrer les milliers de contenus venant de la source, sans surcharger la bande passante des opérateurs et dépasser la capacité des serveurs. En ajoutant une plus haute performance d'acheminement à l'infrastructure Internet existante, les opérateurs vont pouvoir redéfinir et optimiser le routage de leur contenu via Internet.

Vu que seulement une ou deux entreprises dominaient le marché des services d'acheminement de contenus grâce à leur propre infrastructure "clé en main", le choix d'un fournisseur de contenus pour les opérateurs réseaux restait naturellement très restreint. Ces opérateurs réseaux ont réagi en construisant leur propre réseau CDN pour s'adresser au marché des fournisseurs de contenus.

Ils participent également au développement d'un nouveau modèle basé sur l'interconnexion entre les opérateurs réseaux Internet : C'est le Content Peering (échange de contenu). Grâce à cette ouverture, ActiVia Networks s'est concentré sur le Content Peering et croit fermement que ce modèle viendra à s'appliquer dans l'industrie du CDN.

Cette intuition a dors et déjà été confirmée par deux leaders sur le marché du CDN : Inktomi, avec le Content Bridge Consortium, et CISCO en développant la Content Alliance qui impose le concept du Content Peering comme un standard. ActiVia Networks a rejoint la Content Alliance (IETF) en septembre 2000.

## 4 Environnement du stage

### 4.1 Qu'est ce qu'un CDN ?

Un CDN est un ensemble d'éléments réseaux composant un réseau virtuel, "au-dessus" de l'Internet, et qui est conçu pour acheminer de la manière la plus efficace possible divers types de contenus. Ce nouveau réseau a pour but d'améliorer les performances des applications multimédia du web, comme les flux audios ou vidéos qui peuvent également être interactifs.

Le défi posé par les CDN aux opérateurs réseaux, ainsi qu'aux entreprises fournissant des contenus nécessitant une grosse bande passante, est de fournir ces nouveaux services tout en maintenant la disponibilité, les performances et la sécurité des applications traditionnelles. Ceci nécessite une gestion efficace :

1. en permettant l'interconnexion de CDN afin d'augmenter l'audience.
2. en répartissant intelligemment le trafic entre les différents serveurs représentants (*load balancing*).
3. en mettant en place des politiques de distribution pour l'importation, la pré-visualisation et l'exportation des flux audios/vidéos et des contenus statiques.
4. en implémentant le *reverse proxy caching*<sup>2</sup> (également appelé *surrogate*, "serveur représentant" en français), l'hébergement web distribué, la réplication de contenus. Cela permet en effet de réduire les temps d'accès et de réponse dus à la distance, puisque le contenu est fourni au plus près du demandeur.
5. en personnalisant et en mettant en place des priorités aux contenus pour certains utilisateurs.

D'après le draft IETF "Model for Content Internetworking" [DCTR01], un CDN doit combiner l'approche de gestion de cache du *reverse caching proxy* à la réplication de contenu du *proxy cache*. Un CDN contient plusieurs copies de chaque contenu hébergé. Une requête depuis un navigateur vers l'un de ces contenus est dirigée vers la "bonne" copie, de telle sorte que le contenu soit délivré plus rapidement au client, comparé au temps nécessaire pour rapatrier le contenu depuis le serveur d'origine. Des informations statiques sur les emplacements géographiques et la connexion réseau ne sont en général pas suffisantes pour sélectionner un cache. Un CDN utilisera plutôt des informations dynamiques pour vérifier l'encombrement du réseau et la charge des caches, permettant ainsi de faire de l'équilibrage de charge sur tous les caches et d'assurer la meilleure qualité de service possible.

---

<sup>2</sup>proxy-cache inverse, <http://www.newi.ac.uk/pullina/proxy/sld007.htm>

## 4.2 L'intérêt du CDN

Différents acteurs entrent en jeu dans les CDN :

1. **Les éditeurs**, créateurs de contenus. Ceci comprend aussi bien les pages web statiques que les pages générées dynamiquement à partir de bases de données ou bien à partir de requêtes utilisateurs, ou encore les flux vidéos et audios, et même les données non-web délivrées directement à une application cliente.
2. **Les serveurs “représentants”** (ou proxy-caches inverses) sont en général localisés sur les points de présence (PoP) d'un CDN. Ils jouent le rôle du serveur d'origine et en sont les représentants locaux en fournissant le contenu à l'utilisateur. Les données sont collectées auprès de l'éditeur via le système de distribution du CDN.
3. **Les redirecteurs** sont au cœur du CDN. Les requêtes des utilisateurs concernant certains contenus web sont orientés non pas vers le serveur web, mais plutôt vers un système complexe de recherche qui prend en compte un certain nombre de facteurs lui permettant de sélectionner le *serveur représentant* le plus apte à fournir le contenu demandé par le client. Ces facteurs peuvent être : Rechercher le cache ayant la charge minimale, minimiser le temps de réponse en fonction de la localisation du client, personnaliser le contenu de la page web avec des informations locales afin de fournir un contenu ayant la plus forte valeur ajoutée possible au client.
4. **Les CDN** offrent une analyse des flux de données permettant de gérer les paiements au travers d'un CDN. Par exemple : Un éditeur paie un CDN pour ses services, les clients paient pour la qualité de service fournie ou bien pour obtenir un contenu spécifique, les CDN se paient entre eux pour faire transiter ou bien distribuer leurs contenus.

## 4.3 Interconnexion des CDN (*content peering*)

Le système d'interconnexion des CDN n'est pas encore défini, seuls les besoins sont exprimés au travers de drafts IETF [CSMB01] [ATS01] [GNS00]. Les requêtes sur des contenus peuvent être routées via une série de CDN interconnectés. De manière similaire, deux ou plusieurs systèmes de distribution pourraient être connectés afin de fournir une distribution élargie du contenu. Enfin, la facturation et la comptabilité des informations devraient également pouvoir être échangées entre les CDN.

Afin de lier plusieurs CDN, une passerelle d'interconnexion<sup>3</sup> sera utilisée,

---

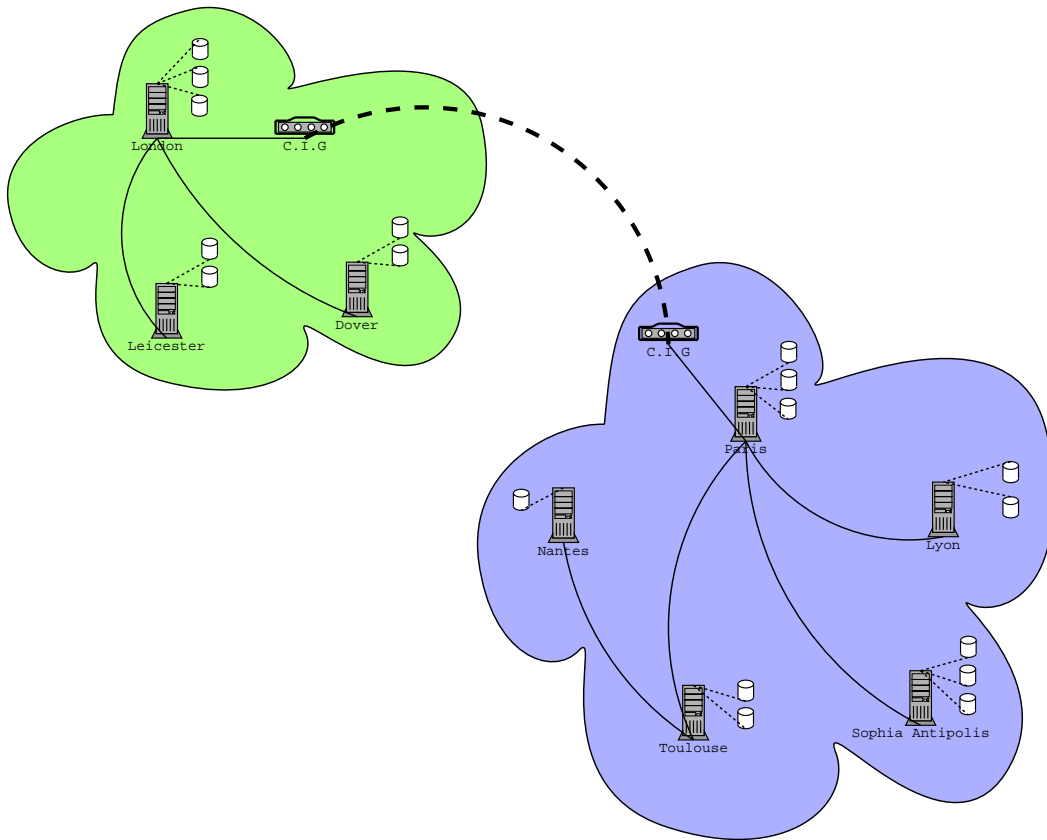
<sup>3</sup> *Content Internetworking Gateway* (CIG)

c'est-à-dire un routeur de contenu en charge de la communication et des échanges entre deux CDN. Les CIG s'échangent des annonces, et il en existe trois sortes différentes correspondant à la facturation, au routage et à la distribution. Une annonce de facturation décrit une collection de contenus pour lesquelles un système de facturation désire recevoir des informations quand le contenu est distribué. Une annonce de routage décrit une collection de contenus qui peuvent être fournis au travers du CDN émetteur. Enfin une annonce de distribution décrit les niveaux de services disponibles à partir des *serveurs représentants* à destination d'une audience.

#### 4.4 La solution proposée par ActiVia

Les CDN proposés par *ActiVia Networks* sont principalement basés sur le DNS : Lorsqu'une requête arrive sur le serveur DNS autoritatif du domaine utilisant un CDN, l'IP du cache le plus "proche" est renvoyée. Par *proche*, nous entendons une proximité en terme de topologie réseau, mais également en tenant compte de l'encombrement du réseau ainsi que de la charge des caches, le but étant de fournir la meilleure qualité de service possible à l'utilisateur final. Il est possible de fournir des flux audios et vidéos de type *Real* ou encore *Windows Media*, ainsi que des pages web statiques et dynamiques.

## Exemple d'interconnexion de deux CDN



Voici un exemple d'interconnexion de deux CDN. Le premier CDN est localisé en France et se compose de 5 points de présence, et le second en Angleterre compte 3 points de présence. Les points de présence sont les portes d'accès aux utilisateurs finaux.

Chaque point de présence se compose d'une A\*Star, ainsi que d'un ou plusieurs caches (Network Appliance, Cache Flow, Inktomi). Un cache contient des documents web, audios et/ou vidéos. Chaque cache obtient son contenu de sa A\*Star mère. Les A\*Star de Paris et de Londres étant les A\*Star maître des deux CDN.

Les deux CDN sont interconnectés par deux passerelles (CIG, *Content Interconnecting Gateway*), elles-mêmes reliées à la A\*Star maître de leur CDN respectif. Ainsi, si un nouveau document est ajouté au CDN français, la CIG en est informée et peut envoyer le contenu, si nécessaire, à la CIG du CDN anglais. Cette dernière se chargera alors de la transmettre à la A\*Star maître, qui diffusera le contenu au sein du CDN.

## 5 Objectifs scientifiques

Ce stage fait office de stage de fin d'études à l'ESSI<sup>4</sup> (option *Systèmes et Applications Réparties*), ainsi qu'au DEA *Réseaux et Systèmes Distribués*. Mon objectif est d'effectuer un travail de recherche ainsi que de développement dans un environnement industriel. J'ai donc intégré le département R&D de la société *ActiVia Networks* à Sophia Antipolis, où j'ai en charge l'étude puis le développement du "**Content Peering**", c'est-à-dire l'interconnexion de CDN<sup>5</sup>. Ce projet contient donc une partie analyse/conception, ainsi qu'une partie implémentation de la solution.

Cette technologie étant émergente (il n'existe pas encore de RFC sur l'interconnexion entre les CDN), j'ai la chance de pouvoir prendre ce projet dès son début et ainsi participer à sa conception. Dans un premier temps, des tests (statiques) d'interconnexion sont mis en place entre *AT&T* et *ActiVia Networks*, et on m'en a confié la charge, sous la direction de Martin May. Puis dans une seconde partie, le but sera d'implémenter un protocole d'interconnexion des CDN, dont la spécification est en cours de normalisation par le *Content Distribution Internetworking Working Group* (IETF).

Ce projet est particulièrement intéressant, car il s'inscrit dans le cadre d'un projet international. En effet, il est réalisé dans le cadre d'un (futur) groupe IETF. J'observe ainsi les différentes étapes du processus de normalisation, grâce à la contribution de la communauté scientifique du monde des réseaux, mais également la création de drafts Internet.

J'ai également contribué au développement de scripts Perl qui interviennent dans la gestion des fichiers de configuration (XML) de la A\*Star.

---

<sup>4</sup>Ecole Supérieure en Sciences Informatiques

<sup>5</sup>Content Distribution Networks

## 6 Plan de travail

Voici le plan du travail réalisé durant mon stage :

- Comprendre ce qu'est un CDN
- Fonctionnalités nécessaires pour l'interconnexion de CDN
- Mise en place de l'interconnexion entre deux, puis plusieurs CDN
  - Compréhension de DNS
  - Tests avec AT&T et Nortel Networks
- Conception d'un protocole d'interconnexion entre les systèmes de redirection
- Développement du protocole à partir du draft en cours d'élaboration
- Intégration à la A\*Star développée par ActiVia

Mon stage se limite au routage entre les CDN, en supposant que la distribution des contenus est déjà effectuée.

## 7 Outils utilisés

Les outils utilisés sont en nombre réduit, puisque une bonne partie du travail réalisé consiste en un travail de documentation.

Par la suite les outils réseaux classiques de **Linux** ont été nécessaires : `dig`, `ifconfig`, `arp`, `route`, `traceroute`, `ping`, `ssh`.

Et plus tard, `gcc` et `Perl` lors de l'implémentation du protocole de routage appliqué à l'interconnexion des CDN. Le logiciel libre GNU `Zebra`<sup>6</sup> sous Linux, implémentant divers démons de routage, a servi de base au développement du protocole.

---

<sup>6</sup><http://www.zebra.org>

## 8 Travail réalisé

Mon stage a débuté le 17 avril, c'est-à-dire après la fin des examens de ma dernière année à l'ESSI. En introduction au sujet, j'ai lu une bonne partie des drafts IETF traitant des *Content Distribution Networks* ([www.ietf.org](http://www.ietf.org)), puis ceux se rapportant à l'interconnexion des CDN ([www.content-alliance.org](http://www.content-alliance.org)). La liste exhaustive figure dans la bibliographie.

### 8.1 Interconnexion de CDN

L'interconnexion de CDN sur laquelle j'ai travaillé se décompose en trois parties :

**Request Routing Peering** Il route les requêtes vers le cache adéquat de l'un des CDN interconnectés. Pour cela chaque CDN interagit avec les autres par l'intermédiaire d'annonces afin de publier de nouveaux contenus, ou bien au contraire en supprimer. Ainsi chaque CDN possède une table régulièrement mise à jour des différents contenus disponibles. Un contenu peut être disponible sur plusieurs CDN, des métriques permettent d'aider dans le choix du CDN le plus apte à répondre à la requête. Interviennent également les filtres, permettant de n'échanger certains contenus qu'avec certains caches de certains CDN bien précis, en général pour satisfaire la politique de routage, ou bien pour ne cibler que certaines zones géographiques.

Il peut être de deux types : Basé sur DNS, ou bien basé sur HTTP. L'avantage de cette seconde méthode est que l'on connaît la totalité de la requête ainsi que l'adresse IP du client, alors qu'avec la méthode basée sur DNS on ne connaît que l'adresse IP du *resolver* du client (et non celle du client lui-même !). Par contre l'avantage de DNS est d'être absolument transparent pour l'utilisateur, les redirections s'effectuant avec des CNAME. Alors qu'au contraire, la méthode basée sur HTTP implique que le navigateur de l'utilisateur connaît la redirection HTTP. C'est la solution basée sur DNS qui est mise en avant par ActiVia.

Lors de l'établissement de l'interconnexion entre deux CDN, ceux-ci doivent annoncer quelles sont leurs capacités : Web statique et/ou dynamique, streaming audio, streaming vidéo (Windows Media, Real, ...).

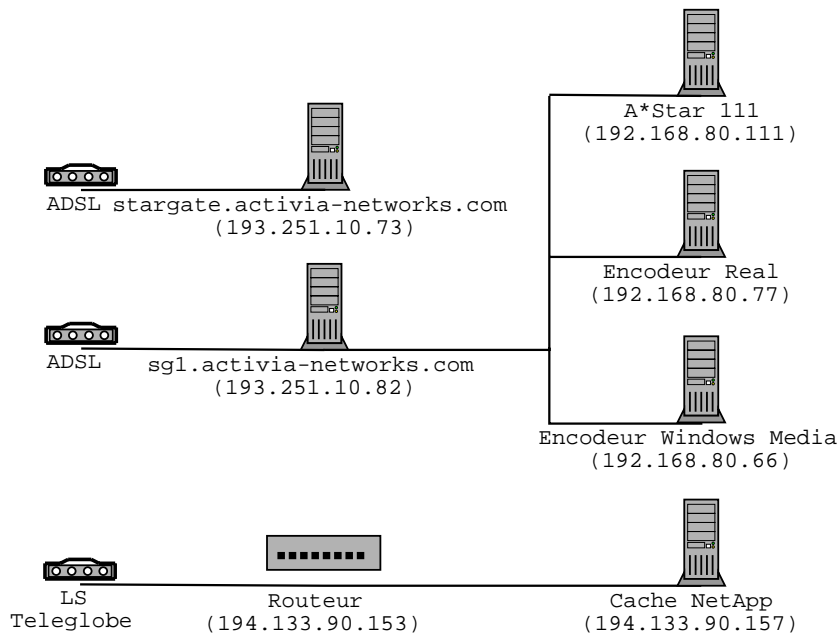
**Content Distribution** Cette partie permet la distribution des contenus entre les différents CDN (la distribution à l'intérieur même du CDN est gérée par ce dernier). Durant la distribution, il faut tenir compte du fait que certains contenus (régionaux, par exemple) ne doivent être délivrés que sur certains points de présence bien précis et non sur la

totalité des caches qui composent le CDN.

**Accounting** Cette dernière partie s'occupe de l'analyse des flux et des statistiques sur les contenus. On peut ainsi permettre la facturation d'un contenu, même s'il est consulté depuis un CDN autre que le CDN autoritatif sur ce contenu. Des statistiques sur la consultation d'un ou plusieurs contenus sont également possibles.

## 8.2 Test d'interconnexion avec AT&T

J'ai participé à la mise en place de l'interconnexion entre le CDN d'*ActiVia Networks* en France et celui d'*AT&T* aux Etats-Unis : Mode de fonctionnement<sup>7</sup>, topologie du réseau, etc... Voici le schéma de l'architecture côté *Activia Networks* :



`stargate.activia-networks.com` est le serveur DNS autoritatif sur le domaine `content-federation.net`. Toute requête reçue sera renvoyée, par un "CNAME", vers un autre serveur DNS : `sg1.activia-networks.com`. Cette machine est configurée de telle sorte que les paquets reçus sur le port 53 sont renvoyés sur une autre machine (grâce à l'option de *Port Forwarding* du noyau de Linux), c'est-à-dire vers un routeur de contenu connectée au réseau local : Il s'agit de la **A\*Star 111**. En fonction de l'adresse IP ayant

<sup>7</sup>Pour plus d'informations, vous pouvez consulter la page suivante : <http://www.cditestbed.net>



2. Le résolveur DNS du client trouve le serveur de nom autoritatif (le redirecteur *ActiVia*, un routeur de contenu *A\*Star*)
3. Le redirecteur *ActiVia* détecte que la requête vient d'une région interconnectée avec *ActiVia*, et redirige donc la requête vers le serveur DNS d'*AT&T* en utilisant un **CNAME** = `activia1.target25.com`. Le CNAME utilisé a bien sûr été négocié au préalable.
4. Le résolveur DNS du client *AT&T* résout la nouvelle requête et la renvoie alors au redirecteur *AT&T*, autoritatif sur le domaine `target25.com`.
5. Le redirecteur *AT&T* répond à la requête du DNS du client *AT&T*.
6. La réponse finale est renvoyée au client *AT&T*.
7. Enfin, le client *AT&T* peut effectuer une requête HTTP GET avec l'adresse IP retournée par le DNS *AT&T*.

### 8.3 Protocole de Content Peering

Les tests actuels ont un défaut : Ils sont statiques. C'est-à-dire que les différents CDN interconnectés contiennent chacun une copie du contenu des autres. Mais que se passera-t'il si une modification intervient dans l'un (voir plusieurs) des contenus ? Dans l'état actuel des choses, nous aurons à mettre à jour le nouveau contenu manuellement dans tous les CDN. Ceci n'est bien évidemment pas exploitable à plus grande échelle.

Il est nécessaire d'établir une communication directe entre les deux (puis  $n$ ) CDN, afin qu'ils puissent s'informer mutuellement de leur état, et se transférer les différentes évolutions des contenus.

De plus, dans le cas où d'autres CDN viendraient se greffer à ces deux premiers, il faudrait penser à une manière efficace de gérer le routage inter-CDN grâce à un protocole : *Request Routing Peering Protocol (RRPP)*.

Enfin, il faudra également prévoir la possibilité de facturer les autres CDN pour que l'on distribue leur contenu, ou bien tout simplement pouvoir gérer des statistiques sur les demandes liées aux contenus distribués par les différents CDN. Cette partie n'a pas été traitée durant ce stage.

#### 8.3.1 “Request Routing Peering Protocol”

Ce protocole a été en partie conçu au sein d'*ActiVia*. Pour faciliter le déploiement du système d'échange d'informations sur le routage des requêtes, nous avons décidé d'implémenter ce système dans le protocole BGP/MGBP. Ce protocole de routage inter-domaine est utilisé dans l'Internet depuis de nombreuses années et fait preuve d'une grande fiabilité et stabilité. Une version du draft (en cours d'élaboration) est disponible en annexe.

Notre choix s'est tourné vers GNU Zebra<sup>8</sup>, logiciel libre implémentant différents protocoles de routage : BGP4, BGP4+, OSPFv2, OSPFv3, RIPv1, RIPv2, et RIPng. C'est bien sûr son implémentation de BGP/MBGP qui sera utilisée.

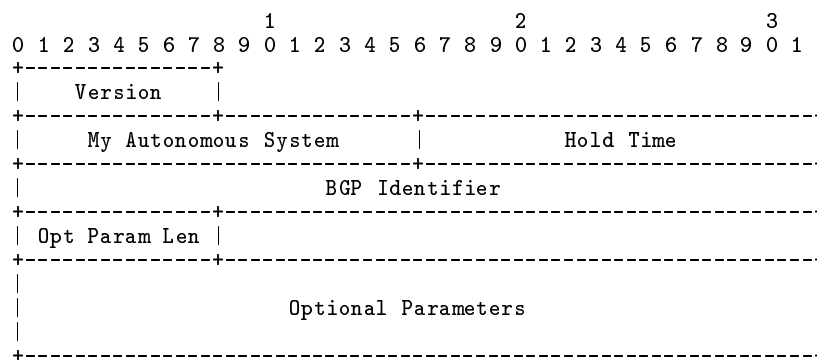
Dans un premier temps, j'ai donc du analyser le code (en langage C) du démon de routage afin de comprendre sa structure et son fonctionnement. A titre d'information, il y a plus de 35000 lignes de code pour le démon BGP, et 50000 lignes pour les bibliothèques communes aux différents démons de GNU Zebra.

Une fois les différents éléments importants isolés, j'ai ensuite implémenté la phase de négociation du protocole RRPP : Ajout d'une *capacité* Routage de Contenu, gestion des URL dans le champ *NLRI* des messages MBGP, ajout de commandes au terminal virtuel de bgpd ainsi qu'au fichier de configuration.

### 8.3.2 Implémentation du protocole

Nous avons implémenté les mécanismes d'échange d'informations sur le contenu qui sont décrit dans le draft en cours délaboration : "Request Routing Peering Protocol". Pour faciliter le déploiement et la validation de ces mécanismes dans un environnement réel, nous avons choisi de les implémenter dans le protocole BGP/MBGP.

Lors de l'établissement de la connexion entre deux démons BGP distants, un paquet de type OPEN est envoyé. En voici la structure :



- Paquet OPEN de BGP -

Nous n'entrerons pas dans le détail de la signification de ces champs, pour plus de détails reportez vous au RFC correspondant [RL95]. La fin de l'entête représente les paramètres optionnels, dont voici le format :

---

<sup>8</sup><http://www.zebra.org>

1															
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5
Parm. Type					Parm. Length					Parm. Value (variable)					

- Format d'un OPTIONAL PARAMETER -

Le type 2 correspond à CAPABILITY [CS00], c'est-à-dire une fonctionnalité particulière qui est disponible. La valeur associée à ce paramètre est du même type que précédemment :

1															
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5
Capa. Code					Capa. Length					Capa. Value (variable)					

- Format d'une CAPABILITY -

Le Code 1 correspond à Multiprotocol Extensions (MBGP, [BRCK00]). Nous avons défini notre propre code 150, correspondant au CONTENT PEERING. Son format est le suivant :

1										2										3											
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Content Family Identifier										RRS Identifier																					
Content Type										Distribution Type																					
Metric Code										Metric Update Frequency																					
Content Type										Distribution Type																					
Metric Code										Metric Update Frequency																					

- Format des parametres du CONTENT PEERING -

Le message est composé de deux premiers champs de 16 bits chacun, suivis de  $n$  fois 64 bits, chaque bloc correspondant à un type de contenu. Voici une description des différents champs :

**Content Family Identifier** : Cet entier non signé sur 2 octets indique quel type de RRS<sup>9</sup> est associé aux différentes capacités qui suivent. Les valeurs actuellement définies sont : CFI=1 pour une redirection basée sur DNS, et CFI=2 pour une redirection basée sur HTTP.

**RRS Identifier** : Cet entier non signé sur 2 octets représente l'identificateur de nœud RRS de l'expéditeur. Cette valeur est déterminée à l'établissement de la connexion et est la même pour chaque connexion du/au CIG.

---

<sup>9</sup>Request Routing System

**Content Type** : Ce champ de 2 octets représente un type de contenu associé à une métrique particulière. Le type de contenu correspond à une qualité de service particulière, par exemple un petit document Web requiert une faible latence alors que pour un gros document on privilégiera plutôt une large bande passante. Les types de contenus actuellement définis sont les suivants : Web statique (1), Web statique de grosse taille<sup>10</sup> (2), Web dynamique (3), Streaming audio à la demande (4), Streaming audio live (5), Streaming vidéo à la demande (6), Streaming vidéo live (7).

**Distribution Type** : Ce champ fournit des informations complémentaires sur le type de distribution utilisé lors de la phase de diffusion du document au sein du CDN. Chaque bit de ce champ de 2 octets représente un type de distribution : push (bit 16), alm (bit 17), multicast (bit 18), pull (bit 19).

**Metric Code** : Ce champ représente la métrique utilisée par le type de contenu concerné. Les codes définis sont : RTT min (0), RTT moyen (1), RTT cumulé (2), bande passante maximale disponible (3), bande passante moyenne disponible (4), bande passante cumulée disponible (5), nombre d'intermédiaires<sup>11</sup> (6), nombre d'intermédiaires cumulés (7).

**Metric Update Frequency** : Cet entier non signé sur 2 octets correspond au nombre de secondes qui séparent 2 envois de mesures de la métrique utilisée. La CIG faisant tourner le protocole doit calculer la valeur de la MUF en utilisant la plus faible valeur, entre les MUF configurées et celles reçues dans les messages de type NEGOCIATION de BGP. Toute implémentation doit pouvoir rejeter des connexions en se basant sur la MUF. La valeur minimale calculée indique le nombre maximal de secondes qui peuvent s'écouler entre 2 réceptions de messages de type ADVERTISEMENT par l'expéditeur, et fournit ainsi un mécanisme naturel de "keepalive" (i.e. "vivacité" d'une session).

---

<sup>10</sup>supérieur à 100 Ko de données, par exemple

<sup>11</sup>*hops*, en anglais

## 9 Conclusion

Les réseaux d'acheminement de contenu (CDN) permettent une utilisation optimale des ressources réseaux actuellement disponibles sur l'Internet. Grâce à un routage "intelligent" et en rapprochant le contenu des utilisateurs finaux, ils apportent ainsi la qualité de service nécessaire aux nouvelles applications multimédia. Toute la force du CDN réside dans le fait que sa mise en œuvre ne nécessite aucune modification de l'architecture actuelle de l'Internet, elle essaie au contraire d'en tirer le meilleur parti, et également de participer à sa décongestion.

A l'aide du protocole RRPP en cours d'élaboration, on pourra interconnecter un ou plusieurs réseaux d'acheminement de contenus, permettant ainsi une plus vaste couverture des utilisateurs. A l'aide d'informations dynamiques à propos de l'encombrement du réseau ou bien encore de la charge des caches, l'utilisateur sera ainsi redirigé vers un cache ayant les performances optimales pour lui délivrer le contenu désiré.

A l'issue de ce stage, après la mise en place de tests statiques d'interconnexion entre deux CDN (ActiVia Networks et AT&T), j'ai implémenté une partie du protocole "Request Routing Peering Protocol", protocole en partie développé au sein d'ActiVia. Et plus précisément la phase de "Négociation". Le protocole BGP a servi de base à mon travail de développement, car il a prouvé sa stabilité et sa flexibilité durant ses nombreuses années d'utilisation dans l'Internet, de plus notre protocole utilise des mécanismes très similaires. D'autre part l'implémentation a été réalisée dans BGP/MBGP dans un but de déploiement rapide et de passage à l'échelle.

## Annexe

# A Request Routing Peering Protocol

Voici le draft sur lequel je me suis basé pour implémenter le routage de contenu. Il est en cours d'élaboration et sera présenté à la prochaine réunion de l'IETF. Mon travail a été d'implémenter cette spécification dans le protocole BGP actuel. Je me suis arrêté à la phase de "négociation".

Network Working Group  
Internet-Draft  
Expires: December 30, 2001

D. Kaplan  
ActiVia Networks  
July 2001

A Request-Routing Peering Protocol (RRPP)  
draft-kaplan-req-routing-peer-proto-00.txt

## Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/1id-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

This Internet-Draft will expire on December 30, 2001.

## Copyright Notice

Copyright (C) The Internet Society (2001). All Rights Reserved.

## Discussion List & Archives

This document and related documents are discussed on the cdn mailing list. To join the list, send mail to [cdn-request@ops.ietf.org](mailto:cdn-request@ops.ietf.org). To contribute to the discussion, send mail to [cdn@ops.ietf.org](mailto:cdn@ops.ietf.org). The archives are at [ftp://ops.ietf.org/pub/lists/cdn.\\*](ftp://ops.ietf.org/pub/lists/cdn.*).

## Abstract

This draft describes the design of the RRPP protocol that meets the requirements defined in [6] for interconnection of Request-Routing Systems (RRS). The content information exchange implemented by RRPP has some analogy with the IP information exchange implemented by the external protocol BGP [1]. The RRPP protocol exchanges capabilities, content and network advertisements, using similar mechanisms as capabilities and updates underlying BGP/MBGP [1][2], mechanisms which have been validated by several years of operation in the Internet. The RRPP protocol also relies on BGP concepts to enforce inter-CDN request-routing policy rules.

## Table of Contents

1.	Introduction . . . . .	3
2.	The RRPP protocol principles . . . . .	5
3.	Content Routing Information . . . . .	7
3.1	Content Routes . . . . .	7
3.2	Content Routing and Forwarding . . . . .	8
3.3	Interaction with the Distribution System . . . . .	9
4.	Request-Routing System Peering Overview . . . . .	11
4.1	NEGOCIATION of Capabilities . . . . .	11
4.2	ADVERTISEMENT of AREA and CONTENT information . . . . .	12
5.	RRPP Messages . . . . .	14
5.1	Message Header Format . . . . .	14
5.2	NEGOCIATION Message Format . . . . .	15
5.3	ADVERTISEMENT Message Format . . . . .	19
5.4	NOTIFICATION Message Format . . . . .	23
5.4.1	Metric Update Frequency Expired error handling . . . . .	23
6.	Aggregation of Content Routes . . . . .	24
7.	Decision Process . . . . .	25
8.	Metrics . . . . .	26
8.1	Defined Metrics . . . . .	26
9.	Redirection Policy . . . . .	27
10.	Implementation Considerations . . . . .	28
11.	Practical Design . . . . .	29
12.	Acronyms . . . . .	30
13.	Acknowledgements . . . . .	31
	References . . . . .	32
	Author's Address . . . . .	32
	Full Copyright Statement . . . . .	33

## 1. Introduction

To face the explosion in size of the Internet two major steps have been done:

- o First step to scale routing in the Internet in 1988 (besides hierarchy in the IP address) was to divide the network into Autonomous Systems and define a hierarchical routing (BGP for inter-AS/IGP for intra-AS).

- o Second step to scale the Internet is made by the caching technology which has been overwhelming the network since several years (Web Caching and Splitting).

The "Bring the Network to the Edge" paradigm solves backbone congestion and server overload problems, satisfying both:

- o the end users by reducing their response time,
- o the network operators by reducing their bandwidth needs.

More recently, a number of CDNs - Content Distribution Network - have emerged in order to organize the distribution of content and perform the redirection of client requests to the "best" surrogate in an efficient way. Surrogates are also called "reverse proxy" or "accelerator server"; they are intermediaries that act on behalf of an origin server. The CDNs are separately-administered content networks, where a content network is under the same management authority. They are generally composed of a set of surrogates and three principal architectural components: a Distribution System, a Request-Routing System and an Accounting System.

As the number of CDNs increases, the need for their interconnection [4] becomes crucial. A content provider outsources the distribution of its content to a set of CDNs in one of the following ways:

1. If the content provider is not a CDN he relies on a given Authoritative CDN to outsource the distribution and request-routing systems. The Authoritative CDN becomes responsible for the peering with other CDNs.
2. If the content provider is a CDN (ie is equipped at least with a request-routing system) he is himself the authoritative CDN which peers with other CDNs.

The peer CDNs themselves peer with other CDNs to increase the SCALE and REACH parameters described in [4], resulting in an increase of the content provider audience. We suppose in the rest of the draft that the content provider is a CDN as stated in case 2. Two approaches are possible to provide the peering mechanisms:

1. Define a new protocol.
2. Use MBGP with a new address family (Content Network Address) to specify the content routing policy rules.

We propose to take the first approach and develop a new protocol, namely RRPP, relying on BGP/MBGP concepts [1][2]. The benefit of such an approach is twofold:

1. A "BGP-like" protocol relies on well known and safe mechanisms to vehicle content routing information and provide content networks peering.

2. A "BGP-like" protocol should require little modification to be integrated in the classical BGP for rapid deployment.

In the following we call an aggregation of content a "content volume" (ie a set of DNS names or a set of URIs). Using RRPP, two CIGs (Content Internetworking Gateway - [4]) exchange content reachability information including network coverage (content network capacity, availability, reliability, etc.), distribution used for the content (distribution type, content type, origin, etc.) and content volumes.

We address here the design of both DNS based and in-line/http based request-routing systems [5] peering in a uniform framework. Nevertheless, the examples and scenarios we explicit to illustrate our framework, are taken most of the time from the dns based request-routing peering.

## 2. The RRPP protocol principles

Content routing does not follow exactly the same principles as classical IP routing. A "content route" is characterized by both destination and source locations. A single source may be replicated on a set of surrogates which belong to several CDNs. The aim of content routing is to find the "best" surrogate to serve a given client request, according to current service and network conditions.

To carry on the comparaison, let us define informally the IP routing function by:  $F(\langle \text{dest} \rangle) = \langle \text{ipaddr} \rangle$ , where  $\langle \text{dest} \rangle$  is the IP address of the destination (end user), and  $\langle \text{ipaddr} \rangle$  is the IP address of the next hop towards the destination. Then the content routing function is informally defined by:  $F_c(\langle \text{dest} \rangle, \langle \text{content} \rangle) = \langle \text{cname} \rangle$ , where  $\langle \text{dest} \rangle$  is the IP address of the destination,  $\langle \text{content} \rangle$  is the DNS name or URI that represents the requested content, and  $\langle \text{cname} \rangle$  is the DNS CNAME which should be resolved to the best surrogate ip address. We get off the traditionnal external routing in the sense that  $\langle \text{cname} \rangle$  points at the final CDN that will serve the content, not the "next-hop" CDN. In other words we use a recursive method where a RRS directs a request to the next-best RRS but expects an answer to return to the client [6].

The alternative would have been to follow the well-known "hop-by-hop" routing paradigm and would require some loop avoidance mechanism (like concatenation of CDNs identifier in the DNS name for example). In our framework, the RRPP messages propagate any change in "best" surrogates to all the Authoritative Content Providers. The resulting inter-CDN redirection process effectivily stops after the first step initiated by the Content Provider. Note that RRPP messages are themselves protected from loops by the CDN-PATH attribute (similar to the AS-PATH attribute of BGP).

The design of RRPP is guided by the requirements defined in [6]. Two phases are identified. First the NEGOCIATION phase exchange some capability information such as the request-routing system supported

(dns or http based), the content types distributed, etc. When the NEGOCIATION phase has been successfully handled, the ADVERTISEMENT phase begins. This second phase involves the exchange of both AREA and CONTENT information.

The RRPP protocol is used to define inter-CDN request-routing policy the same way BGP is used to define interdomain routing policy. As RRPP relies on BGP mechanisms it also fullfills the following protocol design requirements:

- o minimizing the routing table space,
- o minimizing the number of messages exchanged,
- o avoiding loops and oscillations (robustness),
- o using optimal paths.

Moreover the final state machine of RRPP maps the BGP FSM (section 8 in [1]) which has been running for several years in the Internet.

### 3. Content Routing Information

#### 3.1 Content Routes

Basically, the content routing management with RRPP involves the following components:

- o The Content Topology Data Base.
- o The Routing Computation process.
- o The Local Content Routing Matrix (local to the CDN).
- o The Content-Forwarding Information Base (Content-FIB) built using both The Content Routing Matrix of RRPP and the Local Content Routing Matrix.

The content routes are stored in the Content Topology Database [6] which consists of three distinct parts:

1. The content routing information that has been learned from inbound ADVERTISEMENT messages. They carry routes that are available as an input to the decision process.
2. The content routing information on which the CDN is Authoritative and that the CIG has selected by applying its local policies to the routing information in 1 (according to the RRPP Decision Process described later) and stored in the Content Routing Matrix.
3. The information that the CIG has selected for ADVERTISEMENT to its CDN peers.

The CIG is the top level content router dedicated to CDN peering. It contains the Local Content Routing Matrix, the Content-FIB and the RRPP Content Routing Matrix.

The Routing Computation process (for example the top level dns redirection system) is in charge of:

- o getting the newly learned AREAs and CONTENTs that can be reached by other CDNs,
- o getting the newly distributed CONTENTs/AREAs of the CDN that have to be advertised to content level peers, and
- o update the content-FIB accordingly.

The main difference with BPG is that the unique CIG of a CDN is related with several other peers, whereas each BGP speaker of an AS has only one direct peer.

RRPP also provides a mechanism by which a CIG can inform its peer that a previously advertised content route is no longer available for use.

The request-routing policy is implemented by tools that filter and control content routing (route maps, access lists, etc). The CONTENT ADVERTISEMENT is expressed by the CONTENT attribute which is managed separately and benefit of all the administrative machinery for expressing the policy that network operators and content providers desire in their inter-CDN routing environment. The dns based and http based content peering define both their own content routing policy rules.

### 3.2 Content Routing and Forwarding

The ADVERTISEMENT phase exchanges AREA (list of ip prefixes) and CONTENT Volumes (DNS names or URIs). Upon reception of an update from cdnA concerning a cidr\_i/content\_j entry, a Computation Routing process Authoritative for content\_j creates a new entry if it does not exist or compare the metric and cost parameters with existing entry to keep the "best" one. The DNS name output of the content routing function encode both the type of content and the area covered. For instance, "covs.cdnA.net" represents the type of CONTENT "static-web" and the AREA "210.15.1.0/24".

The Content Routing Matrix (CRM) has rows indexed by contents and columns indexed by prefixes. Each cell represents the CNAME and associated parameters. The Content Forwarding Matrix (CFM) has rows indexed by contents and columns indexed by prefixes. Each cell represents the CNAME which should be resolved to the best surrogate. Note that these two matrices should be implemented by one dimension arrays because lots of cells should remain empty.

Let us consider cndA and cdnB, two content level peers. The Content Forwarding Matrix of cdnB is for example:

	cidr1	cidr2	cidr3
content1	covy.cdnA.net	covx.cdnB.net	covx.cdnB.net
content2	covz.cdnA.net		covu.cdnB.net

The domains covx and covy correspond to the content type of content1 (static web for ex.) and they represent a different set of prefixes reached (covx for prefixes cidr2 and cidr3 reached in cdnB and covy for prefixe cidr1 reached in cdnA). The domains covz and covu correspond to the content type of content2 (streaming on demand with the real tool for ex.) and they represent a different set of prefixes reached (covz for prefixe cidr1 reached in cdnA and covu for prefixe cidr3 reached in cdnB). Both covy and covz include the prefix cidr1, covx includes the prefix cidr2, and both covx and covz include the prefix cidr3. We guess that cidr1 belongs to cdnA whereas cidr2 and cidr3 belong to cdnB but it is not sure. This depends on the current service and network conditions as well as on the decision process involved. According to that CFM, the dns responsible of a given content redirects towards the best surrogate (the one that is associated with the best metric). The implementation of the matrices will depend on the naming conventions used.

A content route is injected in the relevant Content Routing Matrix according to its target Request Routing System type.

When a service becomes unavailable the content route must be immediately suppressed in the CRM and the authoritative peers must be immediately informed.

### 3.3 Interaction with the Distribution System

The source CDN outsources its content in order to increase its SCALE and REACH parameters. The trivial case is at the original root of the distribution tree where the source CDN is the Content Provider, that usually owns only one RRS and always outsources its distribution.

For one content outsourced within a given CDN there exists only one Content Tree (CT) determined by the set of surrogates involved in the distribution of this content. Usally this CT corresponds to a type of content rather than a particular content (ie static web, dynamic web, stream-od-real, stream-live-real, stream-od-wm, stream-live-wm, etc.). The content tree should be independent from the service used to deliver the content. The effective paths taken by the distribution process define a subtree of this CT which represents an AREA associated to a given CONTENT.

The Request-Routing Peering Protocol has to be aware of the distribution trees to advertise them to the peers. As many paths

towards a given destination are available in traditional routing, many content paths are available to serve a given client. The cost reflect the policy of the CDN and might be based on previous Distribution System settlement. Note that the cost information/settlement is learned from the distribution system and should remain opaque to external CDNs. The metric is objective whereas the cost is dependent of the distribution tree and resources used (cache appliances, systems and services used for the distribution). The decision of redirection should be based on dynamic metrics such as the availability/conditions of service and network and on the cost of the distribution subtree that is used.

The distribution system injects newly distributed contents and withdrawn ones into the Computation Routing process in such a way that ADVERTISEMENT messages are triggered towards Content Peers as soon as the information is learned.

#### 4. Request-Routing System Peering Overview

This section sketches the main steps done by RRPP (inspired by BGP) to perform Request-Routing System Peering. Let us suppose that cdnB outsources some of its content to cdnA and both want to peer with request-routing systems of same type. They have to proceed as follows:

1. inform each other they are going to exchange some content information thanks to capability parameter and enable content mode associated with dns based redirection, and
2. begin the content exchange: cdnB advertises the prefixes it can reach for a given content entailing that cdnA updates its Content Routing Matrix, and vice versa.

During the first phase of NEGOCIATION, RRPP verifies that the two peers agree to perform content level peering. During the second phase of ADVERTISEMENT, RRPP exchanges AREA and CONTENT information. We give some insight on what these two phases proceed in the following.

##### 4.1 NEGOCIATION of Capabilities

In order to exchange Content Reachability Information (CRI), two RRPP content routers must have the capabilities to process such CRI. The Capability Attribute of the NEGOCIATION phase codes the different setup parameters that have to be agreed by CDN peers to enable content level peering. It can be roughly represented by the vector:

```
| <RRS Type> | <Data Encoding Type> | <Content Type> (1) | <Metric> (1) |
| <Metric Update Frequency> (1) |
| ..... | <Content Type> (n) | <Metric> (n) |
| <Metric Update Frequency> (n) |
```

The NEGOCIATION phase succeeds if and only if the following conditions between the two vectors exchanged are satisfied:

- o The RRS Types are the same,
- o The set of Content Types are the same,
- o The Metrics associated to a given Content Type are the same, and
- o If the Metric Update Frequencies differ, the highest one is kept.

#### 4.2 ADVERTISEMENT of AREA and CONTENT information

For example, if cdnA distributes the content c1, then the ADVERTISEMENT message comprises both AREA and CONTENT information and can be roughly represented by the vector:

```
| <content> | <area> | <autho> | <CFI> | <toc> | <metric> | <cost> | <cname> |
```

where Content Reachable Information <content> represents a content volume, the Area Reachable Information <area> represents a set of ip prefixes, <autho> states wether the originating CDN is authoritative for that content or not, <CFI> is the Content Family Identifier (RRS type) , <toc> is the Type of Content and <cname> represents the DNS CNAME given by the content peer. During a dns-based RRS peering a possible exchange of vectors could be:

Message 1 (cdnA to cdnB):

- o <content>: content1.cp1.fr
- o <area>: 129.42.17.10/24, 163.121.107.0/24
- o <autho>: yes
- o <CFI>: dns-based RRS Peering
- o <toc>: Static Web
- o <stoc>: http
- o <metric>: 100ms (aggregated by areas)
- o <cname>: static-web1.cdnA.fr

Message 2 (cdnB to cdnA):

- o <content>: content1.cp1.fr
- o <area>: 129.42.17.10/24
- o <autho>: no

- o <CFI>: dns-based RRS Peering
- o <toc>: Static Web
- o <stoc>: http
- o <metric>: 50ms
- o <cname>: static-web12.cdnB.fr

Message 3 (cdnB to cdnA):

- o <content>: content23.cp1.fr, content12.cp1.fr
- o <area>: 129.42.17.10/24,171.159.240.0/20
- o <autho>: yes
- o <CFI>: dns-based RRS Peering
- o <toc>: Stream On-Demand
- o <stoc>: http&(real|windows media)
- o <metric>: 1Mbps (aggregated by areas ans type of content)
- o <cname>: stream-od-real3.cdnB.net,stream-od-wm1.cdnB.net

Note that the DNS CNAME encode the type of content as recommended in [6].

As we will see in the next section, the CONTENT volume "content23.cp1.fr, content12.cp1.fr" travels in the CONTENT Reachable Information field of the CONTENT attribute, the prefixes (129.42.17.10/24,171.159.240.0/20) travel in the AREA Reachable Information field of the ADVERTISEMENT message, the cnames () travel in the field next-hop of the CONTENT Attribute. The CONTENT advertised belongs to a given Content Family (here a set of DNS names).

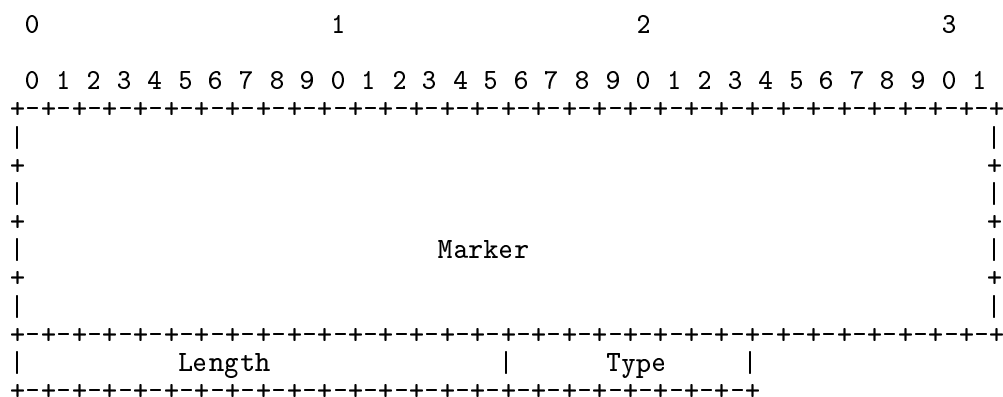
## 5. RRPP Messages

A RRPP session is opened over a TCP connection for reliability, stability and authentication scheme availability reasons. The RRPP protocol uses 4 packet types:

1. NEGOCIATION: setup
2. ADVERTISEMENT: update
3. Keepalive: to monitor a session liveness
4. Notification: to report errors or end connection

## 5.1 Message Header Format

The message header format is the same as the BGP one. Each message has a fixed-size header. There may or may not be a data portion following the header, depending on the message type. The layout of these fields is shown below:

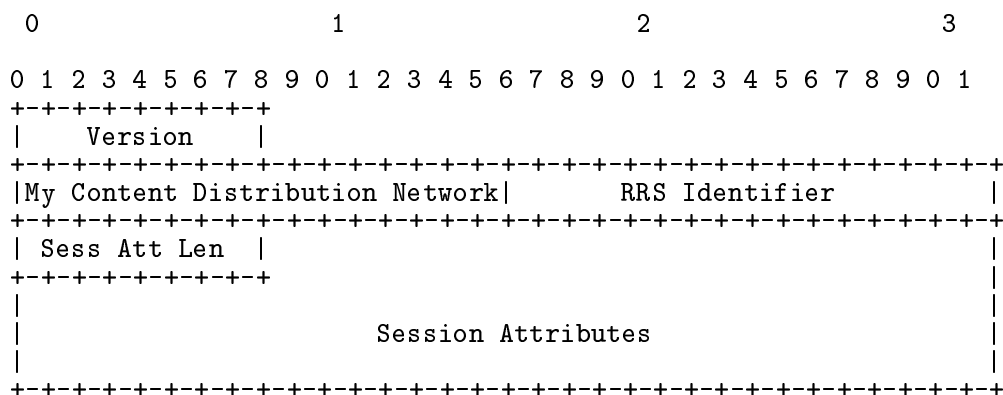


- o Marker: This 16-octet field contains a value that the receiver of the message can predict. If the Type of the message is NEGOCIATION, or if the NEGOCIATION message carries no Authentication Information (as a Session Attribute), then the Marker must be all ones. Otherwise, the value of the marker can be predicted by some computation specified as part of the authentication mechanism (which is specified as part of the Authentication Information) used. The Marker can be used to detect loss of synchronization between a pair of CIG, and to authenticate incoming RRPP messages.
- o Length: This 2-octet unsigned integer indicates the total length of the message, including the header, in octets. Thus, e.g., it allows one to locate in the transport-level stream the (Marker field of the) next message. The value of the Length field must always be at least 19 and no greater than 4096, and may be further constrained, depending on the message type. No "padding" of extra data after the message is allowed, so the Length field must have the smallest value required given the rest of the message.
- o Type: This 1-octet unsigned integer indicates the type code of the message. The following type codes are defined:
  1. NEGOCIATION
  2. ADVERTISEMENT
  3. NOTIFICATION

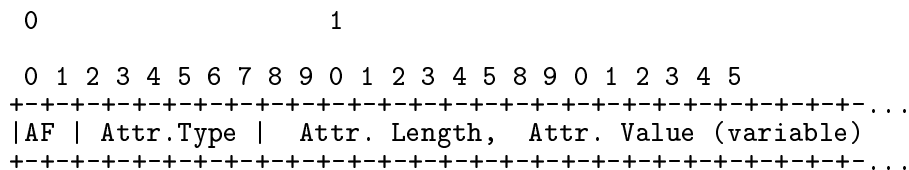
Note that no KEEPALIVE message is needed because a mechanism based on ADVERTISEMENT messages is provided to ensure the liveness of the connection.(see section ??).

## 5.2 NEGOCIATION Message Format

The NEGOCIATION phase initiates a set of agreed session parameters by the means of the Capability Attribute.



- o Version: This 1-octet unsigned integer indicates the protocol version number of the message.
- o My Content Distribution Network: This 2-octet unsigned integer indicates the Content Distribution Network number of the sender.
- o RRS Identifier: This 2-octet unsigned integer indicates the RRS node Identifier of the sender. The value of the RRS Identifier is determined on startup and is the same for every CIG peer.
- o Session Attributes Length: This 1-octet unsigned integer indicates the total length of the Session Attributes field in octets. The value of the Length field must always be at least ??.
- o Session Attributes: This field contain a list of session attributes, where each attribute is encoded as a <Attribute Type, Attribute Length, Attribute Value> triplet.



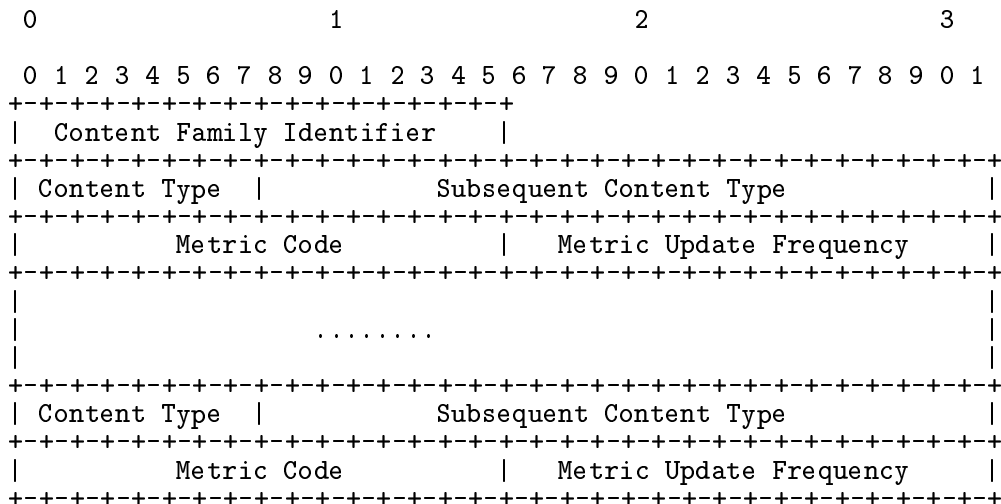
\* The high-order bit (bit 0) of the Attribute Flags octet is the

Optional bit. It defines whether the attribute is optional (if set to 1) or mandatory (if set to 0).

- \* The second high-order bit (bit 1) of the Attribute Flags octet is the Extended Length bit. It defines whether the Attribute Length is one octet (if set to 0) or two octets (if set to 1). Extended Length may be used only if the length of the attribute value is greater than 255 octets.
- \* If the Extended Length bit of the Attribute Flags octet is set to 0, the third octet of the Session Attribute contains the length of the attribute data in octets.
- \* If the Extended Length bit of the Attribute Flags octet is set to 1, then the third and the fourth octets of the session attribute contain the length of the attribute data in octets.
- \* Attribute Type is a 6-bits field that unambiguously identifies individual attributes. Attribute Length is a one or two octet field that contains the length of the Attribute Value field in octets. Attribute Value is a variable length field that is interpreted according to the value of the Attribute Type field.
- \* The remaining octets of the Session Attribute represent the attribute value and are interpreted according to the Attribute Flags and the Attribute Type.

Currently defined Attribute Types are listed in the following.

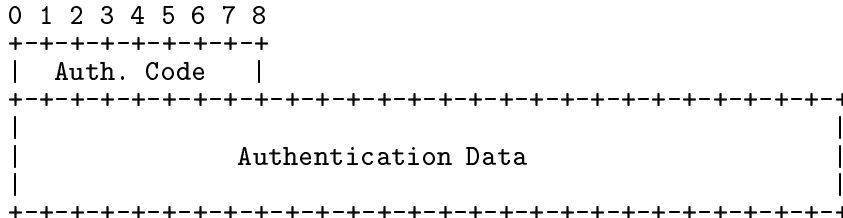
- o Content Features (Attribute Type 1): This mandatory attribute contains a 16-bit word followed by a list of 64-bit words:



- \* Content Family Identifier: This 2-octet unsigned integer indicates which type of request-routing systems is associated to these capabilities. Presently defined values are CFI=1 for DNS-based redirection (contents are specified by DNS names) and CFI=2 for HTTP-based redirection (contents are specified by URIs/URLs).
- \* Content Type: This 1-octet field represents a type of content associated to a particular Metric. This type of content corresponds to a particular QoS requirement, for example a small Web document rather requires small latency whereas a huge one rather requires big bandwidth. The Content Types presently defined are: Static Web (1), Huge Static Web (2), Dynamic Web (3), Audio Streaming On-Demand (4), Audio Streaming Live (5), Video Streaming On-Demand (6), Video Streaming Live (7). Huge can be for example interpreted as greater than 100000 bytes of data.
- \* Subsequent Content Type: This field provides additional information about the type of protocols used by the distribution system for that Content Type. Each bit of this 3-octet field denotes one protocol, and presently defined bits are: ftp (bit-8), http (bit-9), rtsp (bit-10), real (bit-11), windows media (bit-12). For example if video on demand fetched with real/rtsp and windows media/rtsp is supported, the SCT field has value (binary notation): "00111000 00000000 00000000". Note that for CFI=2 the http bit must always be set to one.
- \* ALTERNATIVE for SCT: This field provides additional information about the type of distribution used for content-delivery. Each bit of this 3-octet field denotes one distribution type, and presently defined bits are: push (bit-8), alm (bit-9), multicast (bit-10), and pull (bit-11).
- \* Metric Code: a non exhaustive set of metrics is described in section "Metric". We define the following initial Metric Codes in this document: min RTT (Code 0), mean RTT (Code 1), summarized RTT (Code 2), max available bandwidth (Code 3), mean available bandwidth (Code 4), summarized available bandwidth (Code 5), number of hops (Code 6), summarized number of hops (Code 7).
- \* Metric Update Frequency: 16-bit integer that represents the number of seconds between two Metric measurements that are sent. A RRPP speaker MUST calculate the value of the Metric Update Frequency by using the smaller of its configured Metric Update Frequency and the Metric Update Frequency received in the NEGOCIATION message. An implementation may reject connections on the basis of the Metric Update Frequency. The calculated value indicates a maximum number of seconds that may elapse between the receipt of successive ADVERTISEMENT messages by the sender. The minimum value among all calculated values plays the role of the Hold Timer and provides a natural

"keepalive" mechanism for RRPP.

- o Authentication Information (Attribute Flags 1, Attribute Type): This optional attribute (similar to BGP) may be used to authenticate a RRPP peer. The Attribute Value field contains a 1-octet Authentication Code followed by a variable length Authentication Data.



\* Authentication Code: This 1-octet unsigned integer indicates the authentication mechanism being used. Whenever an authentication mechanism is specified for use within RRPP, three things must be included in the specification:

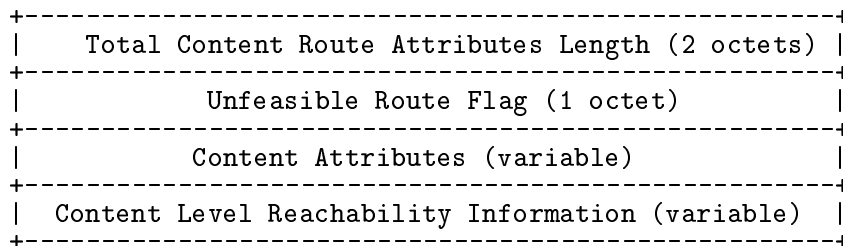
1. the value of the Authentication Code which indicates use of the mechanism,
2. the form and meaning of the Authentication Data, and
3. the algorithm for computing values of Marker fields.

Note that a separate authentication mechanism may be used in establishing the transport level connection.

\* Authentication Data: The form and meaning of this field is a variable-length field depend on the Authentication Code.

### 5.3 ADVERTISEMENT Message Format

The ADVERTISEMENT phase involves these principal fields:

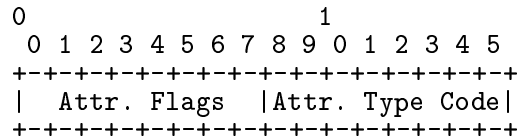


- o Total Content Route Attribute Length: This 2-octet unsigned integer indicates the total length of the Content Route Attributes field in octets. Its value must allow the length of the Content Level Reachability field to be determined as

specified below. A value of 0 indicates that no Content Level Reachability Information field is present in this ADVERTISEMENT message.

- o Unfeasible Route Flag: The high-order bit of the Unfeasible Route Flag octet indicates if the content routes present in the message are being advertised (1) or withdrawn from service (0). The lower-order seven bits of the Unfeasible Route Flag octet are unused. They must be zero (and must be ignored when received).
- o Content Attributes: A variable length sequence of content attributes is present in every ADVERTISEMENT. Each content attribute is a triple <attribute type, attribute length, attribute value> of variable length.

\* Attribute Type is a two-octet field that consists of the Attribute Flags octet followed by the Attribute Type Code octet.



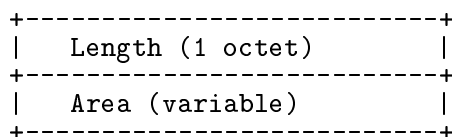
- \* The high-order bit (bit 0) of the Attribute Flags octet is the Optional bit. It defines whether the attribute is optional (if set to 1) or mandatory (if set to 0).
- \* The second high-order bit (bit 1) of the Attribute Flags octet is the Transitive bit. It defines whether an optional attribute is transitive (if set to 1) or non-transitive (if set to 0). For mandatory attributes, the Transitive bit must be set to 1.
- \* The third high-order bit (bit 2) of the Attribute Flags octet is the Partial bit. It defines whether the information contained in the optional transitive attribute is partial (if set to 1) or complete (if set to 0). For mandatory attributes and for optional non-transitive attributes the Partial bit must be set to 0.
- \* The fourth high-order bit (bit 3) of the Attribute Flags octet is the Extended Length bit. It defines whether the Attribute Length is one octet (if set to 0) or two octets (if set to 1). Extended Length may be used only if the length of the attribute value is greater than 255 octets.
- \* The lower-order four bits of the Attribute Flags octet are unused. They must be zero (and must be ignored when received).
- \* The Attribute Type Code octet contains the Attribute Type Code.
- \* If the Extended Length bit of the Attribute Flags octet is set

to 0, the third octet of the Path Attribute contains the length of the attribute data in octets.

- \* If the Extended Length bit of the Attribute Flags octet is set to 1, then the third and the fourth octets of the path attribute contain the length of the attribute data in octets.
  - \* The remaining octets of the Path Attribute represent the attribute value and are interpreted according to the Attribute Flags and the Attribute Type Code.
- o The currently defined Attribute Type Codes, their attribute values and uses are the following:
- \* **AUTHORITATIVE:** mandatory attribute that defines the authority of the CDN that originated the content route:
    1. **Authoritative:** the AREA belongs to an authoritative CDN for the CONTENT information.
    2. **Non Authoritative:** the AREA does not belong to an authoritative CDN for the content information.
    3. **Incomplete:** no content information is associated with the AREA ADVERTISEMENT.
  - \* **CDN-PATH:** mandatory attribute that is composed of a sequence of CDN numbers (similar to BGP AS\_PATH)
  - \* **COST:** optional - represent the cost of the distribution (content tree)
  - \* **CONTENT:** optional attribute to advertise CONTENT of a given Content Family. This attribute is similar to the MBGP MP\_REACH\_NLRI attribute which advertise different address families. It has the format:

```
+-----+
|      Content Family Identifier (1 octet)      |
+-----+
|      Length of Next Hop Content Location (1 octet) |
+-----+
|      Next Hop Content Location (variable)      |
+-----+
|      Content Type (1 octet)                    |
+-----+
|      Subsequent Content Type (3 octet)         |
+-----+
|      Metric Value (4 octet)                   |
+-----+
|      Content Level Reachability Information (variable)|
+-----+
```

- \* The Content Family Identifier identifies the RRS type mode which is enabled by the Capability attribute during the NEGOCIATION phase. Each Content Family has its own policy rules. This attribute contains a field that represents the next hop for the content (CNAME or URL) and a field to describe the type of distribution used. A Content Family which has not been negotiated will be silently disgarded.
  - \* The Content Type and Subsequent Content Type fields have been defined in section (??) and the Metric Value field is a 32-bit unsigned integer. Metrics can be aggregated per areas, content types, etc.
  - \* The Content Level Reachability Information is a variable length field that lists CLRI for the content routes that are being advertised in this attribute. Each CLRI is encoded according to the CFI as specified in section ??.
- o Content Level Reachability Information: This variable length field contains a list of IP prefixes (cidr) that represent AREAs. The length in octets of the Content Level Reachability Information is not encoded explicitly, but can be calculated as: ADVERTISEMENT message Length - 22 - Total Content Route Attributes Length, where ADVERTISEMENT message Length is the value encoded in the fixed- size RRPP header, Total Content Route Attribute Length is the values encoded in the variable part of the ADVERTISEMENT message, and 22 is a combined length of the fixed- size RRPP header, and the Total Content Route Attribute Length field.
  - o Content Level Reachability information is encoded as one or more 2-tuples of the form <length, prefix>, whose fields are described below:



- o The use and the meaning of these fields are as follows:
  - \* Length: The Length field indicates the length in bits of the AREA. A length of zero indicates a prefix that matches all AREAs (with prefix, itself, of zero octets).
  - \* Area: The Area field contains IP address prefixes followed by enough trailing bits to make the end of the field fall on an octet boundary. Note that the value of the trailing bits is irrelevant.

## 5.4 NOTIFICATION Message Format

### 5.4.1 Metric Update Frequency Expired error handling

## 6. Aggregation of Content Routes

Aggregation is the process of combining the characteristics of several different routes in such a way that a single route can be advertised. Aggregation can occur as part of the decision process to reduce the amount of routing information that will be advertised.

Content routes of same Content Family and same authoritative CDN may be aggregated, according to the following rules:

- o **CDN\_PATH** attribute: If routes to be aggregated have identical **CDN\_PATH** attributes, then the aggregated route has the same **CDN\_PATH** attribute as each individual route. We use the rules of BGP concerning **AS\_PATH** attribute adapted for RRPP aggregation.
- o **CONTENT** attribute: In the case of http based request-routing system peering, the HTTP1.1 [3], via its 301 and 302 responses for example, makes namespace redirection where a request on one URL is returned to the client with instructions to resubmit the same request to another URL. There is a tremendous number of URIs involved and the aggregation of URIs into Content Volumes should be performed to reduce this amount of data.
- o **METRIC** attribute: aggregation of the metric per content type, set of AREAs, set of contents is straight forward.

## 7. Decision Process

Only one step redirection decision is stored in the content-FIB of the CIG, ie a CNAME advertised by a given CDN should be resolved within the scope of this CDN.

RRPP selects one content route as the best one and then only forward this content route to its neighbors. To select a content route we propose to perform the following steps:

1. if the **ADVERTISEMENT** specifies a next hop that is inaccessible drop the message,
2. choose the content path with the best metric value according to its associated metric,
3. if values are the same, prefer the content route with the lowest origin type (Authoritative < Non Authoritative < Incomplete),
4. if the origin codes are the same, prefer the lowest cost,
5. if the costs are the same, prefer the internal path over the external path,

6. prefer the path with the lowest RRS node identifier.

## 8. Metrics

Each peering CDN should have the same notion of metric. For example, two peering CDNs in mode DNS might choose a metric based on rtt between the client resolver and the surrogates that can serve the client request. This means that every peer has to agree on a given metric, and that one CDN should have a uniform metric for all its peering point (ex: rtt to closest surrogate, , etc.). Nevertheless, the metric should be associated to a content type and choosed in order to perform efficient "content aware" routing.

The policy opaqueness is hidden in the cost parameter.

### 8.1 Defined Metrics

- o Minimum RTT.
- o RTT summarized on all relevant surrogates of the CDN.
- o Mean RTT over a period.
- o Maximum available bandwith.
- o Available bandwidth summarized on all relevant surrogates of the CDN.
- o Mean available bandwidth over a period.
- o Number of hops.
- o Number of hops summarized on all relevant surrogates of the CDN.

These metrics are not exhaustive. They can be combined to find the best surrogate.

## 9. Redirection Policy

On the one hand, the CDN has to apply a content routing policy to other CDN peers (including the content provider). The request-routing policy rules are implemented by the mechanisms underlying the BGP protocol: Community attribute , Route Maps and Access Control Lists. We manage content routing policy by configuring separate route maps under the content family. In addition, the Community attribute give the possibility to apply the same policy to a set of CDN peers, etc.

On the other hand, if a CDN wants to privilege a particular set of end clients, he uses the community attribute to apply a particular policy and implement its filters.

## 10. Implementation Considerations

The RRPP daemon runs on the Content Internetworking Gateways (CIG) [4] to provide interconnection of request-routing systems.

First solution: Description of implementation in actual BGP to be detailed. The content exchange provided by RRPP is directly implementable in actual BGP using MBGP [2] for content advertisement: the RRPP daemon is installed on BGP speakers, one per AS present in the CDN, instead of the CIG and content peering enabled between them. Capability Attribute 150

## 11. Practical Design

To illustrate our discussion let us consider an operational scenario. To be detailed.

## 12. Acronyms

- o RRS: Request-Routing System
- o RRP: Request-Routing Peering Protocol
- o DSP: Distribution System Peering
- o CN: Content Network
- o CRT: Content Routing Table
- o CT: Content Tree
- o CRM: Content Routing Matrix
- o CFM: Content Forwarding Matrix
- o CIG: Content Internetworking Gateway
- o CFI: Content Family Identifier
- o CRI: Content Routing Information
- o CTD: Content Topology Data Base

## 13. Acknowledgements

## References

- [1] Rekhter, Y. and T. Li, "A Border Gateway Protocol 4 (BGP-4)", RFC 1771, March 1995.
- [2] Bates, T., Rekhter, Y., Chandra, R., Katz, D. and Juniper Networks, "Multiprotocol Extensions for BGP-4", RFC 2858, June 2000.
- [3] Fielding, R., Gettys, J., Mogul, J., Nielsen, H., Masinter, L., Leach, P. and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999.
- [4] Day, M., Cain, B. and G. Tomlinson, "A Model for CDN Peering", Internet draft, draft-day-cdn-model-06.txt (work in progress), November 2000.
- [5] Cain, B., Douglis, F., Green, M., Hofmann, M., Nair, R. and D. Potter, "Known CDN Request Mapping Mechanisms", Internet draft, draft-cain-cdn-known-req-map-00.txt (work in progress), November 2000.
- [6] Cain, B., Spatscheck, O., May, M. and A. Barbir, "Request-Routing Requirements for Content Internetworking", Internet draft, draft-cain-request-routing-req-02.txt, July 2001.
- [7] Green, M., Cain, B. and G. Tomlinson, "CDN Peering Architectural Overview", Internet draft, draft-green-cdn-gen-arch-01.txt (work in progress), October 2000.
- [8] tam ere, "", RFC 2065.

## Author's Address

Delphine Kaplan  
ActiVia Networks  
Space Antipolis 5  
Parc de Sophia Antipolis  
2323 Chemin St Bernard  
06225 Vallauris, Cedex  
FRANCE

Phone: +33 4 97 23 46 66  
EMail: Delphine.Kaplan@activia.net  
URI: <http://www.activia.net/>

## Full Copyright Statement

Copyright (C) The Internet Society (2001). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

#### Acknowledgement

Funding for the RFC editor function is currently provided by the Internet Society.

## B Exemple de script Perl

J'ai programmé une trentaine de scripts Perl pour la A\*Star, soit 2500 lignes de code. Voici par exemple le script `routes.add.pl`, qui permet d'ajouter une entrée au fichier `routes.xml`.

```

                                - routes.xml -
<module name='routes' xmlns:xsi='http://www.w3.org/2000/10/XMLSchema-instance'
  xsi:schemaLocation='../xsd/routes.xsd'>
  <route device='lo' netmask='255.0.0.0' destination='127.0.0.0' />
  <route destination='0.0.0.0' netmask='0.0.0.0' gateway='192.168.8.251' />
  <route netmask='224.0.0.0' device='dummy0' destination='224.0.0.0' />
</module>

                                - routes.add.pl -

#!/usr/bin/perl -w

require 5.6.0;

use Getopt::Declare;
use ActiVia::XML::Updater;

$MODULE_NAME = "routes";
$COMMAND_LAYER = "admin";

sub _testIp {
  my ($address) = @_;
  if (! ($address =~ m/^\d+\.\d+\.\d+\.\d+$/)) {
    return 1;
  }
  @ip= split /\./, $address;
  foreach my $a (@ip) {
    if (($a<0) || ($a>255)) {
      return 2;
    }
  }
  return 0;
}

sub _ckeckIpAddresses {
  my ( $gateway, $netmask, $destination ) = @_;
  # checks ip addresses
  if ($gateway) {
```

```

if (_testIp($gateway) != 0) {
    print STDERR "<exception code='1'>Invalid ip address: " . $gateway .
        "</exception>";
    return 1;
}
}
if ($netmask) {
    if (_testIp($netmask) != 0) {
        print STDERR "<exception code='1'>Invalid ip address: " . $netmask .
            "</exception>";
        return 1;
    }
}
if ($destination) {
    if (_testIp($destination) != 0) {
        print STDERR "<exception code='1'>Invalid ip address: " . $destination .
            "</exception>";
        return 1;
    }
}
return 0;
}

sub main {
    my $args = new Getopt::Declare(q {
        device <device>The device name
        gateway <gateway>The ip address of the gateway
        netmask <netmask>The netmask [required]
        destination <destination>The ip address of the destination [required]

        [mutex: device gateway]
    }, [-BUILD]);

    if ($args->parse()) {
        # checks ip addresses
        if (0 != _checkIpAddresses($args->{'gateway'}, $args->{'netmask'},
            $args->{'destination'})) {
            # invalid ip addresses
            return 1;
        }
    }
    # Files locations

```

```

$xml = $ENV{AV_ACTIVIA_PATH} . "/etc/" . $COMMAND_LAYER . "/xml/" .
$MODULE_NAME . ".xml";
$xmlta = $ENV{AV_ACTIVIA_PATH} . "/etc/" . $COMMAND_LAYER . "/xta/" .
$MODULE_NAME . ".xta";

# First try to find a duplicate entry for the specified device
# xpath
$xmlpath = "/descendant-or-self::route[attribute::destination=" .
    $args->{'destination'} . "]"";
# parse the xml file
my $parser = ActiVia::XML::DOM::Parser2->new();
return 1 unless defined $parser;
my $selector = ActiVia::XML::XPath::Selector->new();
return 1 unless defined $selector;

# try to find the device
$parser->setXML($xml);
$parser->parse();
my $doc = $parser->getDocumentNode();
my $nl = $selector->select( $xmlpath, $doc->getElementNode() );
if ($nl) { # found
    print STDERR "<exception code='1'>A destination entry already exists: " .
    $args->{'destination'} . "</exception>";
    return 1;
}

# Second, now we can insert it
# xpath
$xmlpath = "self::node()";

my $updater = ActiVia::XML::Updater->new();
return 1 unless defined $updater;
# Create the node to insert
my $route = ActiVia::XML::DOM::ElementNode->new();
$route->setName("route");
# The attribute list
my $attrl = $route->getAttributeNodeList();

# mandatory attributes
# device attr
if ($args->{'device'}) {

```

```

my $device = ActiVia::XML::DOM::AttributeNode->new();
return 1 unless defined $device;
$device->setName("device");
$device->setValue($args->{'device'});
$attr1->addNodeTail($device);
}

# gateway attr
if ($args->{'gateway'}) {
my $gateway = ActiVia::XML::DOM::AttributeNode->new();
return 1 unless defined $gateway;
$gateway->setName("gateway");
$gateway->setValue($args->{'gateway'});
$attr1->addNodeTail($gateway);
}

# netmask attr
my $netmask = ActiVia::XML::DOM::AttributeNode->new();
return 1 unless defined $netmask;
$netmask->setName("netmask");
$netmask->setValue($args->{'netmask'});
$attr1->addNodeTail($netmask);

# destination attr
my $destination = ActiVia::XML::DOM::AttributeNode->new();
return 1 unless defined $destination;
$destination->setName("destination");
$destination->setValue($args->{'destination'});
$attr1->addNodeTail($destination);

return $updater->update( "add",
    $xpath,
        $xml,
        $xta,
        $route );
}
}

exit not main( @ARGV );

```



## Références

- [AAH00] B. Aboba, J. Arkko, and D. Harrington. Introduction to Accounting Management. *RFC 2975*, Octobre 2000.
- [ATS01] L. Amini, S. Thomas, and O. Spatscheck. Distribution Peering Requirements for Content Distribution Internetworking. *IETF Network Working Group, Internet Draft*, Février 2001.
- [BCD<sup>+</sup>01] A. Barbir, B. Cain, F. Douglis, M. Green, M. Hofmann, R. Nair, D. Potter, and O. Spatscheck. Known CDN Request-Routing Mechanisms. *IETF Network Working Group, Internet Draft*, Juin 2001.
- [BRCK00] T. Bates, Y. Rekhter, R. Chandra, and D. Katz. Multiprotocol Extensions for BGP-4. *RFC 2858*, Juin 2000.
- [CDN00] CDN Event 2001. *The Ins and Outs of Content Delivery Networks*. Stardust.com, Décembre 2000.
- [CS00] R. Chandra and J. Scudder. Capabilities Advertisement with BGP-4. *RFC 2842*, Mai 2000.
- [CSMB01] B. Cain, O. Spatscheck, M. May, and A. Barbir. Request-Routing Requirements for Content Internetworking. *IETF Network Working Group, Internet Draft*, Janvier 2001.
- [DCTR01] M. Day, B. Cain, G. Tomlinson, and P. Rzewski. A Model for Content Internetworking. *IETF Network Working Group, Internet Draft*, Mars 2001.
- [Del01] C. Deleuze. Les réseaux de distribution de contenu (CDN). Séminaire LIP6, Avril 2001.
- [DGH00] C. Deleuze, L. Gautier, and M. Hallgren. A DNS Based Mapping Peering System for Peering CDNs. *IETF Network Working Group, Internet Draft*, Novembre 2000.
- [DGR01] M. Day, D. Gilletti, and P. Rzewski. Content Internetworking Scenarios. *IETF Network Working Group, Internet Draft*, Mars 2001.
- [GCT<sup>+</sup>01] M. Green, B. Cain, G. Tomlinson, S. Thomas, and P. Rzewski. Content Internetworking Architectural Overview. *IETF Network Working Group, Internet Draft*, Mars 2001.
- [GNS00] D. Gilletti, R. Nair, and J. Scharber. CDN Peering Authentication, Authorization, and Accounting Requirements. *IETF Network Working Group, Internet Draft*, Septembre 2000.

- [Hui00] C. Huitema. *Routing in the Internet*. Prentice Hall PTR, 2nd edition, 2000.
- [Lu01] J. Lu. Propagated Content Delivery Protocol. *Internet Draft*, Février 2001.
- [Moc87a] P. Mockapetris. Domain Names - Concepts and Facilities. *RFC 1034*, Novembre 1987.
- [Moc87b] P. Mockapetris. Domain Names - Implementation and Specification. *RFC 1035*, Novembre 1987.
- [PA01] R. Penno and A. Albuquerque. User Profile Information Protocol. *IETF Network Working Group, Internet Draft*, Avril 2001.
- [Rad] Radware. Cache Directing Technology.
- [Ray01] E. T. Ray. *Learning XML*. O'Reilly, 2001.
- [RBR00] P. Rzewski, J. Bai, and N. Robertson. Origin/Access Content Peering for HTTP. *IETF Network Working Group, Internet Draft*, Novembre 2000.
- [RCR00a] P. Rzewski, B. Cain, and N. Robertson. Cross-Network Accounting for HTTP. *IETF Network Working Group, Internet Draft*, Novembre 2000.
- [RCR00b] P. Rzewski, B. Cain, and N. Robertson. Cross-Network Distribution of Content Signals for HTTP. *IETF Network Working Group, Internet Draft*, Novembre 2000.
- [Riv98] M. Riveill. Distributed Naming Services. *INPG / ENSIMAG*, Octobre 1998.
- [RL95] Y. Rekhter and T. Li. A Border Gateway Protocol 4 (BGP-4). *RFC 1771*, Mars 1995.
- [SOL00] *Securing and Optimizing Linux : RedHat Edition*. Number 1.3. OpenDocs Publishing, Juin 2000.
- [Sol01] Warp Solutions. Dynamic Content Distribution, Avril 2001.
- [Spr01] Sprint, Advanced Technology Labs. *The Basics of BGP Routing and its Performance in Today's Internet*. Nina Taft, Mai 2001.
- [TO99] I. Hyna E. Schlegl T. Oetiker, H. Partl. The Not So Short Introduction to LaTeX 2e, Janvier 1999.
- [WCKT01] B. Whetten, D.M. Chiu, M. Kadansky, and G. Taskale. Reliable Multicast Transport Building Block for TRACK. *IETF RMT Working Group, Internet Draft*, Mars 2001.