ICAP: concept et mise en œuvre du « réseau intelligent du Web »

Amar Nezzari
France Telecom R&D Direction des services Multimédias Internet et intranet
42, rue des Coutures 14066 Caen Cedex 4
amar.nezzariATfrancetelecom.com
date: 25 septembre 2003

Christophe Wolfhugel
France Telecom Transpac
40 rue Gabriel Crié, 92240 Malakoff
wolfAToleane.net

Résumé

Les fournisseurs de services Internet sont de plus en plus souvent confrontés à des contraintes liées à la croissance du nombre de postes connectés et des services Internet de plus en plus évolués. Pour y faire face les ISP ont souvent recours à des solutions logicielles proxy pour rendre leur service. Ces solutions sont bien souvent incompatibles en termes de performances avec la multiplication des accès et économiquement rédhibitoires en termes de retour sur investissements. Pour résoudre cette équation technico-économique, le protocole iCAP (Internet Content Adaptation Protocol) semble être la solution bien adaptée à ce contexte. Ce protocole a été crée sur l'initiative de Network Appliance en collaboration avec France Telecom R&D au sein de l'IETF. Il repose sur des échanges simples entre un client et un serveur iCAP agissant sur des requêtes ou des réponses http. La mise en oeuvre d'iCAP ouvre le champ à de nombreuses applications en cœur de réseau transparentes d'un point de vue de l'utilisateur et totalement sécurisées. Le transfert d'une partie de l'intelligence applicative des postes client vers le cœur de réseau ouvre la voie à de nouvelles conceptions de services alliant sécurité et performances (filtrage, anti-virus, compression de données). Plusieurs types d'architectures iCAP et non iCAP pour des services de filtrage d'URLs sont proposées et comparées. Dans tous les cas de figures une solution iCAP permet de doubler les performances par rapport à des solutions classiques à base de chaînage de proxy. Cette solution a été mise en œuvre de façon opérationnelle par France Telecom dans le cadre du projet E-Lorraine, qui vise à fournir des services de filtrage de contenus et d'URL's pour des accès Internet à tous les lycées de la région.

Mots clefs

Caches, iCAP, Internet, Filtrage, Proxy.

1 Introduction

Les fournisseurs de services Internet et gestionnaires de réseaux d'entreprise sont de plus en plus confrontés à des contraintes liées à la croissance du nombre de postes connectés et à des services Internet de plus en plus évolués. L'utilisation de solutions logicielles à base de « proxy » est bien souvent nécessaire à la fourniture de ces nouveaux services. Les solutions classiques sont bien souvent incompatibles avec les objectifs de performances et de coût imposés par les clients et opérateurs. Pour résoudre cette équation technico-économique, le protocole ouvert iCAP semble être la solution adaptée à ce contexte car il offre une interopérabilité totale entre les services et la nature des réseaux (Internet et Intranet) d'une part et permet des fonctionnalités d'adaptation et de transformation des contenus Web d'autre part.

2 Qu'est-ce qu'iCAP ?

2.1 Définition

Le protocole iCAP pour Internet Content Adaptation Protocol a initialement vu le jour suite aux travaux menés par Network Appliance [1] co-fondateur en décembre 1999 avec Akamai de l'iCAP forum [2]. Ces travaux visaient à offrir une solution alternative à la plate-forme de services de proxy-cache et de transformation pour les mobiles proposée par Inktomi. L'objectif du forum est de standardiser les échanges entre les équipementiers représentatifs des architectures du Web (serveurs, caches, répartiteurs de charges, routeurs, etc...) d'une part et les systèmes de transformation de contenus (filtrage,

adaptation, insertion, etc…) d'autre part. France Telecom R&D à participé activement aux efforts de standardisation à l'IETF en tant que membre associé et partenaire technologique de Network Appliance. Aujourd'hui, le protocole iCAP est défini par la norme IETF (soumis en novembre 2000) et relatif à la RFC 3507. Cette RFC n'est pas un standard Internet mais une RFC "Informationnel" en attendant les recommandation du groupe OPES [3].

2.2 Principe

Le principe du protocole iCAP (Figure 1) repose sur une base simple puisque : à toute requête http correspond une réponse http provenant d'un serveur Web ou d'un cache. Pour offrir un service de transformation de contenus il suffit de modifier soit la requête, soit la réponse. Le rôle du protocole iCAP n'est pas d'effectuer lui-même la transformation des contenus, il se limite à réaliser le lien entre des « clients iCAP » et des « serveurs iCAP ». D'un point de vue fonctionnel, le protocole iCAP est particulièrement bien adapté à http même si d'autres protocoles tels que FTP, SMTP, SSL font aujourd'hui l'objet d'implémentations sur des services liés au filtrage de mail ou à la sécurisation des transactions. Les « clients iCAP » se situent essentiellement au niveau des caches qui peuvent être de type logiciel ou matériel pour rendre sans dégradation de qualité de service les fonctions spécifiques telles que le contrôle anti-virus, filtrage, etc... et améliorer les performances.

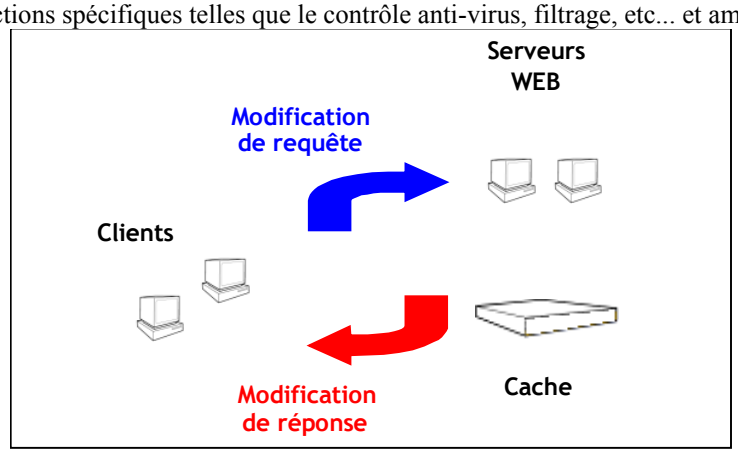

Figure 1 - *Principe de base d'iCAP*

En ce qui concerne le « serveur iCAP » il est quant à lui situé au niveau du serveur d'application (Figure 2). D'un point de vue implémentation l'appel d'un service iCAP se fait par l'intermédiaire d'un cache selon 4 modes (mode « pré-cache » et « post-cache » en modification de requête ou de réponse) en raison de la configuration classique du cache entre le poste client et le serveur de contenus de type proxy explicite ou transparent. Ces modes d'appels aux services sont tous définis au niveau de la norme iCAP, ils sont utilisés par exemple dans le cas d'un service de filtrage d'URL en mode modification de requête « pré-cache » ou un service de filtrage de contenus de type anti-virus en mode modification de réponse « pré-cache » (voir annexe « exemples de transactions iCAP »). On voit donc que le protocole iCAP peut être interprété au même niveau qu'une API de type réseau qui permet à un équipement situé en cœur de réseau de faire appel à un serveur externe agissant sur des flux entre client et serveur. De ce fait un serveur iCAP peut lui-même être connecté à des services iCAP différents sans modifier l'architecture grâce à l'utilisation proxy-cache qui permet au protocole iCAP de gérer l'appel aux services.

Les principaux avantages apportés par une architecture iCAP sont qu'elle permet de séparer les fonctions proxy et services, de concevoir une architecture de services distribués « en étoile » remplaçant ainsi une architecture de chaînage de proxies, l'utilisation d'ACL (Accès Control List) qui conditionnent les appels aux services par des paramètres tels que les types objets, @IP source, @IP destination etc. de « cacher » les traitements effectués par les services iCAP, enfin de distribuer par le proxy-cache des traitements vers différents serveurs d'applications. Pour améliorer les performances au niveau des traitements des services iCAP, l'utilisation des règles des ACL permet en autre de gérer les appels aux services iCAP en fonction des paramètres http. Dans ces conditions le proxy-cache ne renverra au service iCAP que les objets ayant besoin d'un traitement spécifique. Le proxy-cache permet aussi de réduire la charge sur les services par la fonction de cache des réponses iCAP en plus du cache des réponses HTTP. Ces performances restent toutefois très dépendantes de celles des services iCAP donc de la puissance des serveurs applicatifs d'une part et des proxies-caches utilisés d'autre part. Il est à noter que tous ces services iCAP, compte tenu de leur architecture, offrent l'avantage d'une solution de services centralisés, permettant ainsi de déporter une grande partie de l'intelligence applicative des postes clients en cœur de réseau sans configuration spécifique des paramètres de connexion des clients (proxy, browser, @IP, etc.) ni du réseau d'accès IP.

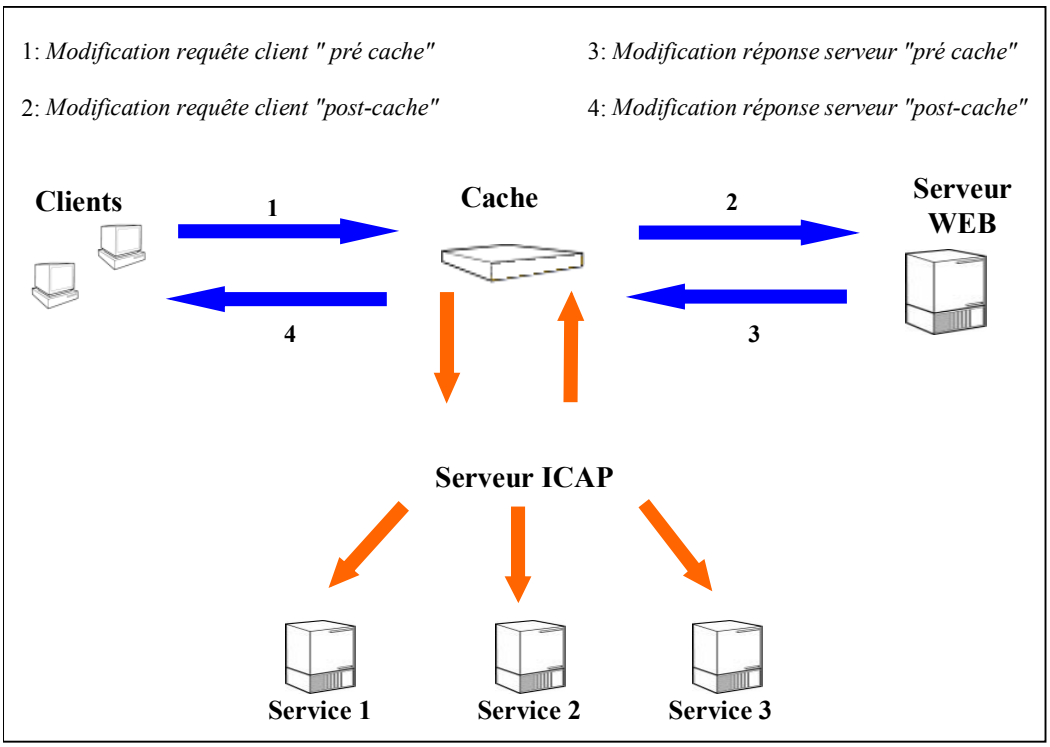

Figure 2 – *Modes d'appel aux services iCAP*

3 Mise en œuvre d'iCAP

La mise en œuvre du protocole iCAP ouvre le champ à de nombreuses applications en cœur de réseau complètement transparentes d'un point de vue de l'utilisateur final. Le transfert d'une partie de l'intelligence applicative des postes client vers le cœur de réseau ouvre la voie à de nouvelles conceptions de services alliant sécurité et performance à la fois tout en réduisant fortement les investissements en équipements. Malgré des débuts difficiles surtout face à des solutions à base d'API, la liste des applications et des produits compatibles avec le protocole iCAP ne cesse de croître ce qui permet maintenant d'envisager des déploiements opérationnels dans de nombreux cas de figure. Nous trouvons aujourd'hui des clients iCAP sur un grand nombre de proxy-cache (Network Appliance, BlueCoat, Array Networks), implémentés sur des machines spécialisées, ainsi que des logiciels en OpenSource tel que Squid par exemple. Même si la majorité des services iCAP visés repose sur des applications liées à la sécurité des contenus (anti-virus), des services liés au filtrage de contenus Web trouve notamment un large écho pour des applications pour le Grand Public et pour les Entreprises. A titre d'exemple France Telecom dans le cadre du projet E-Lorraine a déployé un service iCAP de filtrage de contenus et d'anti-virus sur un réseau IP de 200 sites géographiquement répartis.

3.1 Architecture à base de proxy-cache logiciel non-iCAP

De nombreuses applications client/serveur utilisent des architectures à base de proxy-cache non-iCAP pour rendre des services de filtrage ou anti-virus par exemple. Le principe de fonctionnement repose sur une architecture utilisant un proxy-cache qui dispose d'une API (Application Programmation Interface) permettant le développement de briques logicielles spécifiques pour rendre des services. Le plus connu et l'un des plus utilisé est le proxy-cache Traffic-Server d'Inktomi, l'API interne permet de dialoguer avec les services (Figure 3) externes. La solution reste propriétaire, ce qui présente un inconvénient pour les évolutions fonctionnelles des services d'une part, et tout ajout ou modification des services nécessite une modification de l'automate d'état de Traffic-Server d'autre part. Les logiciels de traitement de contenus anti-virus ou filtrage sont disponibles auprès des éditeurs, cependant ceux-ci n'ont jamais été utilisés simultanément sur le même proxy-cache.

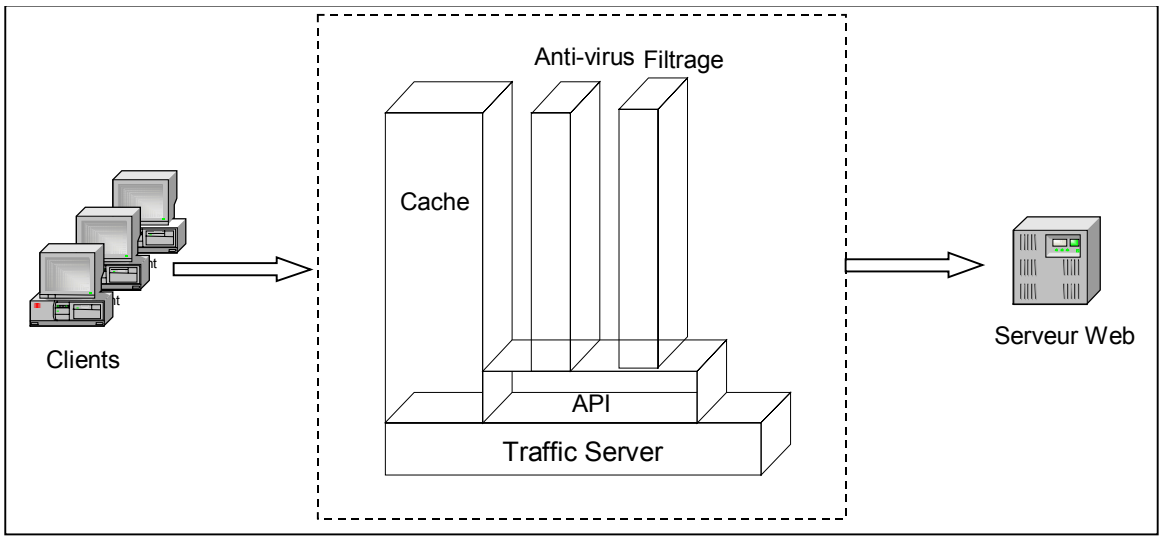

Figure 3 - *Architecture proxy-cache logiciel non-iCAP*

Il n'existe à ce jour que très peu d'informations faisant référence à la stabilité, aux performances et à la fiabilité d'un système dans une telle configuration. Si l'on souhaite avec ce type d'architecture proposer plusieurs services fortement différenciés le chaînage de proxy-cache s'impose, rendant le coût de la solution prohibitif pour obtenir des performances optimales.

3.2 Architecture à base de proxy-cache matériel non-iCAP

Pour palier aux inconvénients en terme de performances liés au proxy-cache logiciel, certains constructeurs comme BlueCoat, avant d'intégrer un client iCAP, ont proposé des solutions à base de proxy-cache matériels (Figure 4). Le principe de fonctionnement reste le même que pour une architecture à base de proxy-cache logiciel, à ceci près qu'il ne dispose pas vraiment d'une API (Application Programmation Interface). En effet, il intègre une grammaire et une syntaxe qui permettent d'agir sur des requêtes http provenant d'un client. Toutes ces règles sont basées sur l'écriture d'expressions régulières, sur les noms de host des serveurs Web, leur adresse IP ou encore sur les noms des utilisateurs et/ou les noms de groupes auxquels ils appartiennent.

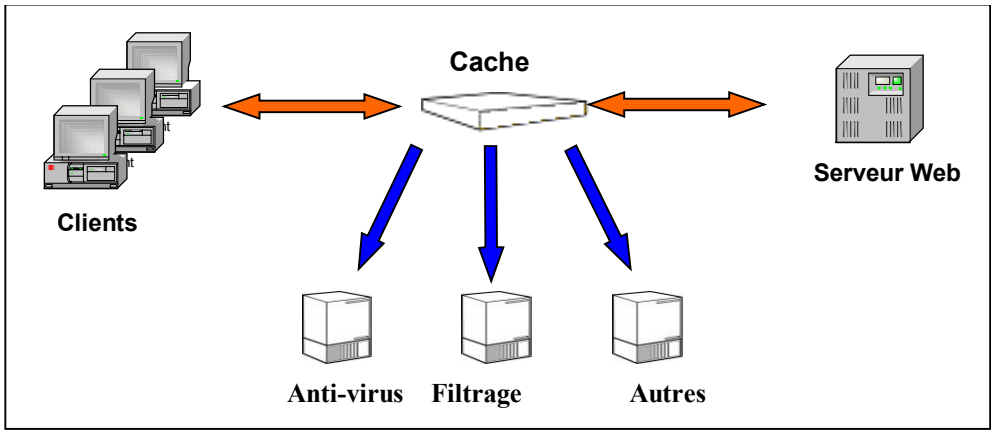

Figure 4 - *Architecture proxy-cache matériel non-iCAP*

La solution reste aussi propriétaire, et restreinte principalement à la manipulation et la réécriture d'URL, d'enrichissement de contenus Web, réorganisation des paramètres passés en ligne de commande dans la requête http ou suppression de certains « headers ». Ces types de proxy-caches matériels sont capables de gérer une base de données embarquée pour effectuer du filtrage de contenus sur la base des produits WebSense et SmartFilter par exemple. Il est possible de créer un ensemble de règles pour définir des profils simples d'activation du filtrage. Cette association est effectuée à la connexion sur le cache, lors de l'authentification de l'utilisateur. Cependant, la syntaxe ne permet pas d'agir sur le contenu des réponses provenant des serveurs Web d'origine, mais seulement sur les « headers » retournés dans cette réponse. Pour implémenter des services de type anti-virus, les constructeurs comme BlueCoat mettent en œuvre un chaînage de proxy. En effet, le cache est chaîné à un autre proxy (en général un serveur sur lequel tourne le module d'anti-virus) qui est lui-même chaîné au

cache (une règle de routage spécifique évitant alors que le système boucle). Le cache passe donc la requête du client vers le proxy anti-virus, qui la retourne au travers du cache pour rechercher le contenu sur Internet. Cette action n'est pas cachée et la réponse est directement envoyée au module anti-virus.

3.3 Etude comparative d'une solution iCAP et non-iCAP

Après une description fonctionnelle des architectures à base de proxy-cache, nous nous proposons dans ce paragraphe de montrer l'intérêt d'une architecture de services utilisant le protocole iCAP par rapport à des solutions utilisant le chaînage de proxy. Pour notre étude de performance, nous avons choisi un service de filtrage d'URL qui permet d'interdire ou d'autoriser l'accès à des URL de sites Web en fonction de critères définis par l'administrateur sous la forme d'une liste thématique (sport, loisir, sexe, …). Le logiciel de filtrage retenu est celui de l'éditeur allemand WebWasher, il permet l'analyse en temps réel des URL demandées par l'utilisateur et la compare à une base de données contenant plusieurs millions d'URL référencées dans une cinquantaine de thèmes. En ce qui concerne les proxy-caches nous avons opté pour une solution Network Appliance et réalisé cette étude en laboratoire sur une plate-forme de tests spécifique. Nous avons envisagé deux scénarii de tests spécifiques, pour le premier correspond à une charge du système à 500 requêtes http/s, le second à 1000 requêtes http/s et ce pendant 8 heures.

3.3.1 Solution non-iCAP

La solution non-iCAP est basée sur une architecture de ferme de caches Network Appliance (Figure 5) avec la fonction de filtrage activée et assurée via une version allégée du logiciel de filtrage de WebWasher dite « *in the box* » car intégrée sur les caches. Dans cette configuration, seule la fonction de filtrage DynaBLocator de WebWasher est présente. Il s'agit d'une base de plusieurs millions d'URLs catégorisées suivant une cinquantaine de thèmes. Chaque thème peut être autorisé ou interdit via l'interface d'administration des caches NetApp.

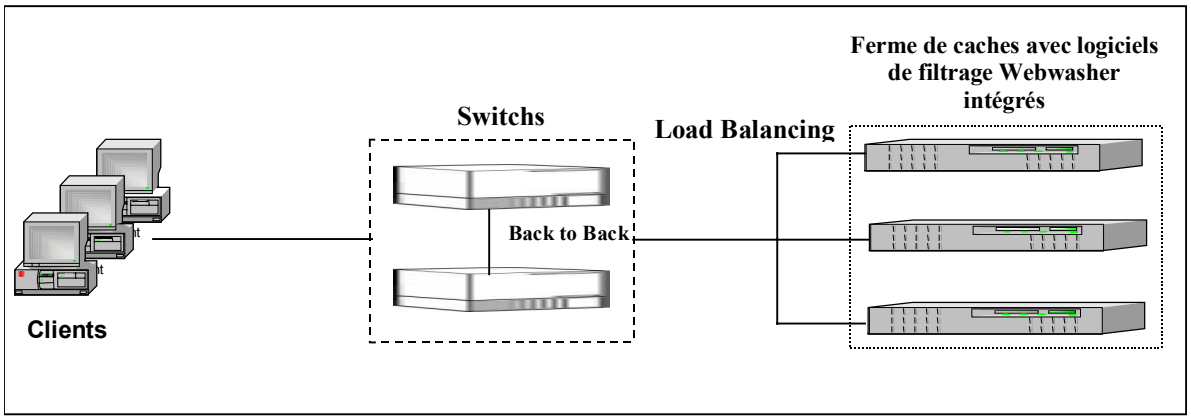

Figure 5 - *Architecture de tests non-iCAP*

Les fonctionnalités du logiciel de filtrage « *in the box* » permettent outre le filtrage des URLs , une mise à jour automatique de la base des URLs, une re-direction des URLs filtrées vers une URL de redirection permettant d'indiquer les raisons du filtrage et enfin la possibilité d'utiliser les résultats de catégorisation avec les ACL du cache. Sur cette solution, on peut noter en particulier l'absence de gestion d'une liste d'URLs personnalisées. Cette solution a pour inconvénient d'associer **1** logiciel de filtrage à **1** cache, contrairement à une solution iCAP qui permet d'associer **n** logiciels de filtrage à **1** cache.

3.3.2 Solution iCAP

La solution iCAP quant à elle est basée aussi sur une ferme de caches Network Appliance (Figure 6), mais à la différence de la solution précédente sur une ferme de serveurs hébergeant les logiciels de filtrage. Ici, les services de filtrage dialoguent avec les caches selon le protocole iCAP décrit précédemment.

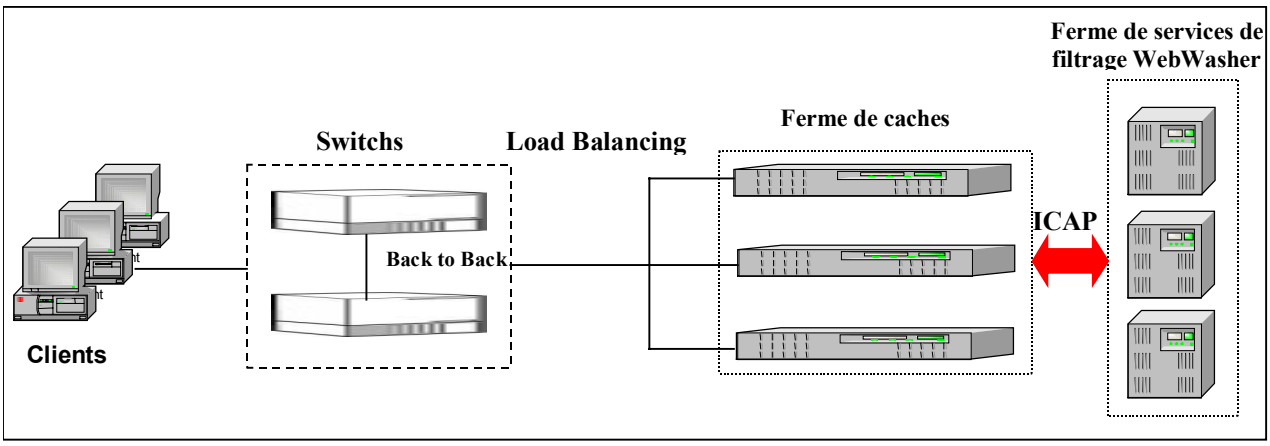

Figure 6 - *Architecture de tests avec iCAP*

Dans cette configuration nous utilisons la version « *out of the box* » de WebWasher. Cette version intègre toutes les fonctionnalités de WebWasher (Dynablocator, filtrage des publicités, filtrage sur liste personnalisée d'URLs, filtrage sur mots clés, etc.). L'utilisation du protocole iCAP permet d'une part d'être indépendant des choix de partenaire effectué par Network Appliance pour assurer le service de filtrage, et d'autre part d'ajouter de nouveaux services iCAP (modification de requête ou l'adaptation de réponse http, détection de virus, etc..).

3.3.3 Analyse des performances

Les tests de performances réalisés sur les architectures décrites aux paragraphes 3.3.1 et 3.3.2 ont pour objectif d'évaluer et comparer la capacité en charge d'une architecture composée d'une ferme de caches et d'une ferme de services iCAP. Pour ces tests nous nous plaçons dans les conditions suivantes :

- les clients, les caches, les serveurs WebWasher et les serveurs de contenus sont sur le même commutateur Ethernet (100 Mb/s) ;
- le cache est dans mis des conditions réalistes (taux de succès de l'ordre de 20%, taille de document suivant une distribution réaliste) mais avec des serveurs très proches (temps de réponses faibles) qui disposent de 72 Go de données ;
- la ferme de serveurs WebWasher est mise dans les conditions les plus pessimistes c'est à dire qu'il y a une requête ICAP vers le serveur pour chaque requête cliente (pas de cache des réponses ICAP), la proximité des serveurs WebWasher du point de vue des caches est réaliste et ne peut être interprétée comme étant des conditions avantageuses pour les tests. La proximité des serveurs de contenus en mode modification de requête n'avantage en rien les serveurs WebWasher qui n'interagissent qu'avec les caches. Dans des conditions réalistes, ce seront les même temps de réponses ICAP qui seront mesurés ;
- le filtrage par URL repose sur des algorithmes de recherche dans des arbres ou tables hashés. L'utilisation de noms d'URL spécifiques aux tests peut simplifier le travail de ces algorithmes. Or la latence perçue par le client est surtout induite par la gestion des connexions. Ceci a pu être vérifié en complexifiant le service WebWasher par des tests sur des expressions régulières et ce sans détérioration de la QoS.

On constate que l'ajout d'un service ICAP à une ferme de caches apporte une dégradation aux capacités en charge de la ferme. La dégradation mesurée est dans nos conditions la plus pessimiste. Enfin, l'algorithme de distribution de charge entre les 2 serveurs WebWasher est du type « Least Usage Based ». Sur le tableau ci-dessous (Figure 7) nous avons représenté les résultats des tests de performances par rapport au paramètre [nombre de requêtes/s / débit en Mb/s]. Nous avons comparé les résultats théoriques issus des données constructeurs, les résultats expérimentaux ainsi que des extrapolations à partir des résultats précédents. La valeur mise en évidence dans ce tableau est celle qui semble répondre au premier scénario, c'est à dire supportant une charge de 500 requêtes par secondes. Par manque de données concernant une architecture de caches composée de 2 caches C2100, nous ne pouvons donner de valeurs. Cependant, nous supposons qu'une telle architecture avec au moins 4 serveurs WebWasher serait théoriquement en mesure de tenir une charge de plus de 1000 requêtes par secondes. Cette supposition serait à valider par des expérimentations supplémentaires. On constate que le comportement des serveurs en cas de surcharge se dégrade (la charge supportée peut chuter temporairement à quelques requêtes par seconde). Aussi, il est important de sur dimensionner la ferme de serveurs de manière à ce que les caches soient le seul « goulot d'étranglement » en cas de surcharge de l'architecture mise en place. Les résultats indiquent q'une solution basée sur ICAP dotée d'une architecture composée de 2 fermes de services présente des performances meilleures (plus du double) qu'une solution de services intégrés sur le proxy cache. Si l'on double dans un même temps le nombre de serveurs, on double dans

un même le débit et le nombre de requêtes/s. Par conséquent la solution iCAP est de loin la plus adaptée de par ses performances et ses capacités d'évolution tant au niveau du dimensionnement qu'aux niveau services.

Requêtes s-1 / Mbit s-1		Ferme de Caches Network Appliance		
		C1105	2*C1105	C2100
Ferme WebWasher	Pas de serveur	C/D théo: 360/33 C/D expé: 460/28	C/D extra: 920/56	C/D théo: 1000/90
	Webwasher Intégré à NetApp	C/D expé: 110/7	C/D extra: 220/14	C/D extra: 280/18
	WebWasher sur IBM server P3 1Ghz 512Mo (M1)	C/D expé: 250/16	C/D extra: 250/16	C/D extra: 250/16
	WebWasher sur IBM server Bi-pro 1Ghz (M2)	C/D expé: 310/19	C/D extra: 310/19	C/D extra: 310/19
	WebWasher sur M1+M2	C/D expé: 450/26	**C/D extra: *500/29***	C/D extra: 500/29
	WebWasher sur 2*(M1+M2)	C/D extra: 450/26	C/D extra: 900/52	C/D extra: 900/52

Figure 7 – *Tableau de synthèse des performances*

3.4 Application au projet E-Lorraine

Le Conseil Régional de Lorraine a lancé le projet « E-Lorraine » [4] en 1999 afin de donner aux établissements scolaires de sa responsabilité accès à des services utilisant les nouvelles technologies, dont l'Internet. En 2002 le projet devient « E-Lorraine Haut-Débit » afin de marquer l'évolution des technologies de raccordement et des débits associés. La plupart des lycées publics, privés, EREA et agricoles ainsi que les CFA de Lorraine accèdent à l'Intranet E-Lorraine par des liaisons de raccordement à haut débit (xDSL, liaisons spécialisées, …) et disposent pour leur activité de navigation sur l'Internet d'un service centralisé mettant en œuvre la technologie iCAP afin de rendre deux services :

- du filtrage d'URL avec un logiciel du marché ;
- un contrôle antivirus sur les objets visualisés.

La mise en œuvre de ces technologies en contexte opérationnel permet d'effectuer des observations fort intéressantes sur les usages et les performances. Une maquette de laboratoire, même avec d'excellents outils de simulations donnerait des résultats moins pertinents.

La navigation se fait en chaînant des postes utilisateurs, ou plus souvent des « proxy-cache » d'établissements à la plate-forme de navigation centralisée (Figure 8) qui est composée de caches Network Appliance (les clients iCAP) et de serveurs Intel Linux pour rendre le service de filtrage d'URL et de contrôle anti-virus.

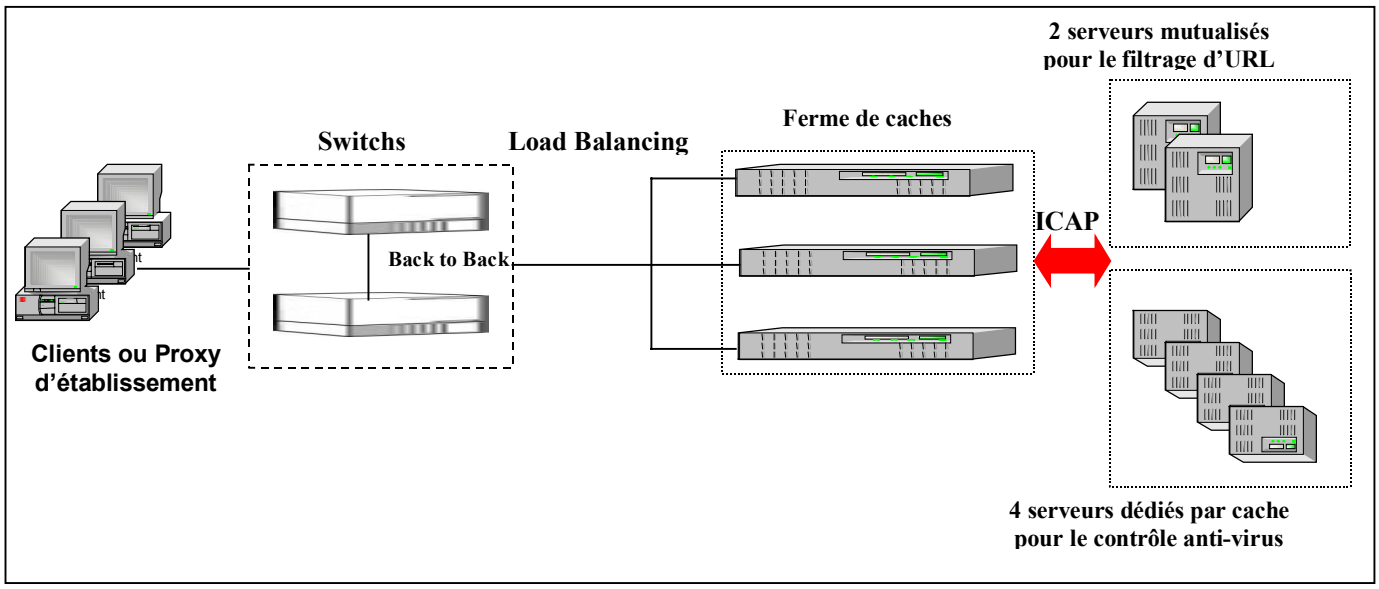

Figure 8 - *Architecture des services iCAP du service E-Lorraine*

Le trafic constaté aux heures de pointe de cette installation est de l'ordre de 400 URL par seconde pour un débit réseau correspondant compris entre 30 et 40 Mbit/s.

3.4.1 Performances – filtrage d'URL vs. Contrôle anti-virus

Comme prévu, le filtrage d'URL est très performant et peu consommateur de ressources. Les machines aujourd'hui déployées sur ce service permettraient sans aucun problème d'absorber de 3 à 4 fois plus de trafic puisque le traitement a effectuer à chaque requête est en effet fort simple :

- peu de données sont transférées entre le client et le serveur iCAP (la requête de l'utilisateur final plus la requête au protocole iCAP) et en retour la réponse qui dans 90% des cas sera un « ok pour afficher la page » ;
- la recherche dans les bases de classification d'URL est très rapide (à condition d'avoir fait le choix d'une grande quantité de mémoire sur les serveurs) ;
- les requêtes renvoyées aux caches sont elles-mêmes cachées, ainsi deux utilisateurs accédant aux mêmes URL dans un temps rapproché ne donnerons lieu qu'à une seule requête iCAP.

Le contrôle anti-virus est au contraire bien plus exigeant puisqu'il faut transférer au serveur iCAP l'intégralité de l'objet afin de pouvoir l'analyser, puis éventuellement le réparer et dans ce cas renvoyer le contenu modifié. La charge due au logiciel anti-virus dépend de deux critères :

- la taille de l'objet ;
- dans le cas d'archives (du « zip » ou du « tar.gz ») le facteur le plus pénalisant est le nombre de fichiers – analyser une archive contenant des milliers de fichiers va prendre plusieurs minutes.

Pour la navigation courante les temps de traitement sont fort heureusement faibles et ne se voient pas à la navigation. Le tableau ci-dessous (Figure 9) donne quelques indications chiffrées sur nos observations.

Echantillon : environ 16,1 millions d'objets.		*Durée moyenne du traitement en secondes*		
		ICAP + Transfert	*Filtrage d'URL*	*Antivirus*
Tous objets confondus	100,0 %	1,10	0,002	0,05
Objet < 32 Ko	96,7 %	1,04	0,002	0,04
32 Ko <= Objet < 128 Ko	3,0 %	1,49	0,001	0,15
128 Ko <= Objet < 1 Mo	0,2 %	8,17	0,001	0,80
Objet >= 1 Mo	0,1 %	83,08	0,001	4,52

Figure 9 - *Temps de traitement des requêtes HTTP et iCAP*

3.4.2 Utilité des diverses fonctions « cache »

Le proxy-cache assure par définition en plus de sa fonction de mandataire une fonction de cache : un contenu déjà chargé peut être proposé à nouveau à un utilisateur qui le demande ultérieurement. A ce cache d'objets s'ajoute un cache des réponses iCAP – aussi bien pour le contrôle anti-virus que pour le filtrage d'URL. Nous observons, toujours sur le même échantillon de 16 millions d'objets les différents taux de « cache » pour les requêtes iCAP :

- le nombre de requêtes iCAP « filtrage d'URL » dont la réponse était déjà dans le cache ;
- le nombre de requêtes iCAP « anti-virus » dont la réponse était déjà dans le cache.

La répartition des valeurs obtenues selon la taille des objets n'est pas surprenante :

Echantillon : environ 16,1 millions d'objets.	*Pourcentage de transactions qui ont trouvé une réponse en cache*	
	iCAP « anti-virus »	*iCAP « filtrage »*
Tous objets confondus	30,5 %	89,2 %
Objet < 32 Ko	31,2 %	89,4 %
32 Ko <= Objet < 128 Ko	11,2 %	82,1 %
128 Ko <= Objet < 1 Mo	15,2 %	76,4 %
Objet >= 1 Mo	14,1 %	73,4 %

Figure 10 - *Taux de succès dans le cache*

La baisse des taux de succès lorsque la taille de l'objet augmente s'explique par deux éléments :

- les objets de grande taille sont consultés par moins d'utilisateurs ;
- l'échantillon n'est plus forcément significatif en taille (quelques milliers d'objets seulement dont la taille est supérieure à 1 Mo).

3.4.3 Partage de charge et dimensionnement

Les clients iCAP peuvent proposer plusieurs algorithmes de partage de charge pour leurs requêtes. Ces éléments sont particulièrement importants pour un service très consommateur comme le contrôle anti-virus. Nous avons essayé deux des solutions proposées par Network Appliance :

- « round-robin » : le NetCache envoie les requêtes iCAP en séquence à chacun des serveurs déclarés ;
- « least usage » : par une méthode qui est peu documentée le NetCache détermine quel serveur iCAP est le moins chargé et le plus apte à répondre dans les meilleurs délais.

Les résultats obtenus montrent que le « round-robin » fonctionne bien, mais ne tolère pas qu'un ou plusieurs serveurs iCAP de la ferme ne répondent plus ou seulement très lentement (lors de l'utilisation de logiciels compliqués comme un moteur anti-virus, il arrive tout simplement qu'un serveur iCAP ne réponde plus du tout !). Il est préférable alors d'utiliser l'algorithme « least usage », dont nous étudions actuellement les détails d'implémentation sur le client iCAP. L'expérience confirme également qu'il faut très largement prévoir les ressources adéquates sur les serveurs iCAP afin d'éviter un écroulement du système. Ainsi pour le contrôle anti-virus, dans la plupart des situations un seul serveur suffit pour traiter toute la vérification anti-virus d'un NetCache (soit environ 90 objets par seconde). Il arrive cependant ponctuellement qu'il soit nécessaire d'utiliser la puissance de deux voire trois machines afin de ne pas écrouler le système et conserver de bons temps de réponse, c'est notamment le cas lors d'un afflux de téléchargements de fichiers plus volumineux qu'à l'habitude.

3.4.4 Comparaison avec une solution non iCAP

Des expériences passées pour rendre des services similaires, mais sans l'usage du protocole iCAP nous permettent de constater le progrès et les bénéfices de cette nouvelle technologies notamment par rapport à une solution de chaînage de proxy :

- pour des services iCAP en modification de requête, comme le filtrage d'URL, la nouvelle architecture est sans comparaison possible avec les anciennes : plus simple, beaucoup plus performante et plus économique (pour les équipements) ;
- pour le service iCAP en modification de réponse qu'est le service anti-virus, là également les progrès sont visibles : la solution est plus performante, plus fiable et plus simple à exploiter – il serait cependant vain de croire que cela permet de réduire de façon très importante le nombre de machines anti-virus nécessaires, ce traitement est très coûteux en temps de calcul et en entrées-sorties, que ce soit avec ou sans iCAP.

4 Conclusions

L'émergence du protocole iCAP est aujourd'hui une réalité puisque tous les constructeurs de caches intègrent cette technologie sur leurs équipements et que l'on trouve de plus en plus de services compatibles tels que le filtrage de contenus, d'URL, antivirus, etc... La solution iCAP est de loin plus performante et plus évolutive qu'une solution à base de chaînage de proxy traditionnel, de plus elle allie à la fois sécurité et confort d'utilisation ce qui augmente le potentiel de son développement. Aujourd'hui fort de ces atouts, France Telecom exploite cette technologie de façon opérationnelle. Enfin, son intégration en cœur de réseau offre un brassage plus riche des flux d'information, permettant ainsi un fonctionnement statistique et un apprentissage plus fiable des usages pour le développement à terme de nouveaux services à forte valeur ajoutée.

Annexe : exemple de transaction iCAP

Mode Modification de requête http "GET" : l'exemple ci-dessous illustre l'action d'un serveur iCAP qui va modifier l'URL qui lui est soumise – l'URL accédée est http://www.origin-server.com/ et l'URL qui est retournée au cache est http://www.origin-server.com/modified-path.

Requête iCAP

```
REQMOD icap://icap-server.net/server?arg=87 ICAP/1.0
Host: icap-server.net
Encapsulated: req-hdr=0, null-body=170

GET / HTTP/1.1
Host: www.origin-server.com
Accept: text/html, text/plain
Accept-Encoding: compress
Cookie: ff39fk3jur@4ii0e02i
If-None-Match: "xyzzy", "r2d2xxxx"
```

Réponse iCAP

```
ICAP/1.0 200 OK
Date: Mon, 10 Jan 2000  09:55:21 GMT
Server: ICAP-Server-Software/1.0
Connection: close
ISTag: "W3E4R7U9-L2E4-2"
Encapsulated: req-hdr=0, null-body=231

GET /modified-path HTTP/1.1
Host: www.origin-server.com
Via: 1.0 icap-server.net (ICAP Example ReqMod Service 1.1)
Accept: text/html, text/plain, image/gif
Accept-Encoding: gzip, compress
If-None-Match: "xyzzy", "r2d2xxxx"
```

Références

[1] Network Appliance, Akamai. *ICAP Internet Content Adaptation Protocol*. IETF Internet Draft, Mars 2000.
[2] Site Web officiel iCAP, http://www.i-cap.org.
[3] *Evaluating the ICAP protocol regarding the OPES callout protocol requirements*. IETF Internet Draft, Juin 2002.
[4] Site Web du projet E-Lorraine Haut-Débit : http://www.e-lorraine.net/.