solutions
réseaux

FRANCIS IA
OLIVIER MENAGER

Optimiser et sécuriser son trafic IP

Avec la contribution de
Jean-Marc Barozet,
Pascal Delprat et
Olivier Seznec,
de Cisco Systems,
et la collaboration de
Olivier Salvatori

ISBN : 2-212-11274-2
EYROLLES

4

L'accélération de flux IP

Que ce soit dans l'environnement Internet ou au sein du système d'information des entreprises, la course à la performance vise de plus en plus à prendre en compte « l'expérience » de l'utilisateur pour la rendre la plus agréable et la plus efficace possible. On commence à parler outre-Atlantique de QoE (Quality of Experience) à la place de la fameuse QoS (Quality of Service).

Lorsqu'on analyse cette expérience dans la chaîne de communication reliant le client au serveur — depuis le tronçon LAN du serveur jusqu'à celui du LAN vers le client en passant par l'accès au réseau WAN et le réseau WAN lui-même —, on se rend compte que chaque tronçon peut constituer un goulet d'étranglement ou à tout le moins une zone de ralentissement de la performance de bout en bout de l'expérience de l'utilisateur vis-à-vis du serveur qu'il cherche à atteindre.

Ces différents tronçons présentent des problématiques et des contraintes techniques d'optimisation à la fois communes et spécifiques.

Les réseaux locaux ne sont plus aujourd'hui des points de blocage de la performance. La plupart d'entre eux mettent en œuvre des technologies majoritairement Ethernet, même si des résidus de Token Ring ou de FDDI (Fiber Distributed Data Interface) se rencontrent encore, dans l'attente d'une migration vers Ethernet. Les débits atteints sont de l'ordre de 10-100 Mbit/s pour le poste de travail ou serveur et supérieurs à 1 Gbit/s pour la commutation dans le cœur du LAN d'interconnexion.

Dans la plupart des cas, c'est dans la zone des serveurs, au niveau même des machines, que le bât blesse. La complexité grandissante des applications exige de plus en plus de puissance de traitement de la part des machines sur lesquelles elles résident. Elle impose en outre des fonctions de traitement réseau performantes pour pouvoir transmettre et recevoir à très haut débit les flux de trafic. C'est la raison pour laquelle apparaissent des solutions d'optimisation éliminant le goulet d'étranglement sur le serveur. C'est le cas des solutions de commutation 4-7 pour les

architectures d'équilibrage de charge *(voir le chapitre 1)* et des solutions de cache ou de chiffrement-déchiffrement SSL (Secure Sockets Layer).

Il est important de bien identifier les points de faiblesse des machines employées. La puissance de traitement de l'unité centrale, ou CPU, doit être dimensionnée en fonction des besoins des applications qu'elle héberge. Des études démontrent que la capacité CPU est souvent consommée par des traitements pour lesquels elle n'est pas prévue, telles que la gestion de sessions de communication simultanées, TCP/IP ou SSL, engendrant une dégradation de la performance globale du serveur. La CPU passe alors son temps à gérer les ouvertures et clôtures de sessions et n'a plus beaucoup de ressources pour exécuter les traitements logiques applicatifs. Il faut donc réduire le nombre de sessions simultanées par serveur tout en tentant de les maintenir ouvertes le plus longtemps possible.

L'accès au réseau WAN offre aujourd'hui de nombreuses possibilités, allant de l'accès mobile, pour les utilisateurs nomades, par exemple, aux classiques accès téléphoniques RTC, en passant par le haut débit à prix abordable, ADSL ou câble. Dans le cas du bas débit, le réseau d'accès peut constituer un point de blocage. Avec le haut débit, l'accès au WAN déporte la problématique de goulet d'étranglement à l'intérieur du réseau WAN. Dans tous les cas, la bande passante de bout en bout doit être suffisante pour le bon fonctionnement des applications communicantes.

Le LAN du client est la zone normalement la moins impactée par les soucis de performance.

En résumé, ce sont les zones des serveurs et le WAN (accès et transit) qui nécessitent une optimisation afin de garantir la performance globale des applications et la qualité d'expérience des utilisateurs. L'accélération de flux IP est la solution la plus satisfaisante à cette problématique.

Les différents paramètres de performance à prendre en compte sont les suivants :

- temps de réponse ;
- débit et bande passante disponibles ;
- latence et durée de traversé du réseau ;
- durée de téléchargement des applications transactionnelles telles que le Web.

La situation économique tendue que connaissent nombre d'entreprises utilisatrices les contraint à extraire le plus de valeur possible de leurs applications à un coût le plus bas possible.

Parmi les approches qui se font jour en ce sens, citons celle consistant à retravailler le code de l'application afin d'en accroître la vitesse d'exécution. Les capacités de l'ingénieur sont alors préférées à l'ajout de ressources matérielles. C'est là un retour en arrière, qui ramène aux systèmes mainframe et aux programmes très optimisés afin de tirer du codage un maximum de performance d'exécution.

D'une façon générale, les approches d'optimisation logicielles restent relativement pauvres en fonctionnalités et surtout limitées en performance comparées aux optimisations matérielles. Concernant la performance, il est essentiel de ne pas constituer un nouveau goulet d'étranglement dans le réseau. C'est la raison pour laquelle cet ouvrage se penche essentiellement sur la description des solutions matérielles.

Les technologies d'accélération IP

Le domaine de l'accélération des flux IP est en pleine explosion aujourd'hui. En atteste la multitude de solutions en tout genre qui apparaissent sur le marché.

Au même titre que la sécurité et la haute disponibilité, la performance est au cœur des préoccupations. Ralentissement économique aidant, la mise en œuvre des solutions d'amélioration de la performance est toutefois différente de celle des années glorieuses. La QoS de bout en bout, par exemple, n'est plus à l'ordre du jour. Les solutions retenues doivent n'engager qu'un minimum d'investissement et ne pas remettre en cause l'architecture existante, ni les technologies en place.

Dans ce contexte, l'intérêt des solutions d'accélération de flux IP réside dans l'insertion au sein de l'infrastructure existante d'équipements spécifiques, comme les serveurs de cache, dotés de fonctionnalités de gestion de trafic (traffic shaping), de priorisation de flux, de compression ou de suppression de duplication.

Principes de base de l'accélération IP

Dans son principe, l'accélération de flux IP consiste à faire parvenir l'information le plus rapidement possible d'un point vers un autre du réseau.

Les applications de nature interactive sont à la fois les plus exigeantes en terme de bande passante et les plus sensibles au temps de latence réseau. Dans un contexte IP, la multiplicité des applications en présence entraîne une hétérogénéité de la taille des paquets transmis, qui rend caduque la théorie de l'équivalence entre la bande passante allouée et le temps de réponse. Par exemple, une session applicative interactive peut disposer d'une bande passante suffisante sans pourtant obtenir un temps de réponse satisfaisant.

Par ailleurs, ces applications doivent être accessibles non seulement *via* le réseau local de l'entreprise (LAN) mais aussi *via* le réseau longue distance (WAN) pour les agences distantes ou les utilisateurs nomades. Les technologies LAN déployées dans les entreprises fournissent généralement une bande passante et un débit suffisants pour les applications qui s'y exécutent et ne constituent pas un point de dégradation des performances d'accès.

Il n'en va pas de même du WAN. L'approche consistant à allouer aux trafics interactifs une bande passante supérieure à celle théoriquement nécessaire améliore certes sensiblement la performance mais au prix d'une utilisation peu efficace des ressources réseau. C'est pourquoi l'approche consistant à ajouter des mécanismes tels que le cache, la compression ou la diminution de la charge des serveurs est privilégiée.

Ces fonctionnalités d'accélération de trafic aident le système d'information à :

- résorber les goulets d'étranglement ou de congestion sur le chemin de données des sessions ;
- améliorer la performance de traitement ;
- maîtriser la consommation de bande passante réseau ;
- préserver l'intégrité des données.

Selon les applications à optimiser, les mécanismes d'accélération peuvent être plus ou moins sophistiqués et performants. Dans tous les cas, ils doivent être déployées à la fois sur le serveur et sur le client.

Comme expliqué précédemment, les services d'accélération visent, dans le contexte de l'accès distant *via* le WAN au système d'information de l'entreprise, à prévenir la dégradation des performances d'accès aux applications, voire, dans certains cas, les dysfonctionnements de ces applications.

La figure 4.1 illustre une architecture type d'accès aux applications d'une entreprise. La zone à optimiser est celle du WAN qui sépare les clients des serveurs d'applications.

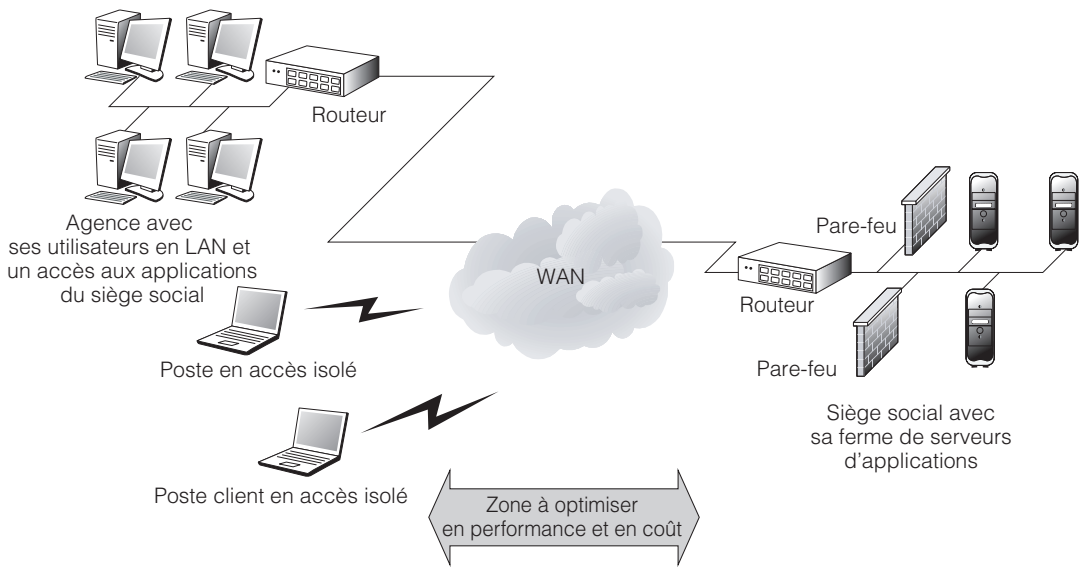


Figure 4.1

Architecture d'accès aux applications de l'entreprise

Les services qui participent à l'accélération du trafic réseau visent à réduire le volume de données à transmettre réellement dans le réseau, à réduire les méfaits causés par le taux d'erreur bit du fait de la mauvaise qualité du réseau ou encore à diminuer la sollicitation des différents maillons de la chaîne de communication du serveur jusqu'au client.

Les techniques génériques d'accélération

Les techniques génériques d'accélération agissent par définition au niveau le plus bas possible de la pile de protocoles OSI afin d'améliorer la performance du plus grand nombre d'applications.

L'équipement d'optimisation est situé à chaque extrémité d'entrée du réseau longue distance WAN, de façon à agir sur l'ensemble du trafic à transmettre dans le réseau.

Certaines implémentations travaillent au niveau IP, offrant ainsi des mécanismes d'accélération à l'ensemble des applications IP communiquant à travers le réseau. Ces implémentations sont transparentes à la fois pour le réseau, les clients et les serveurs. En revanche, les applications autres que IP, telles que SNA, Decnet, IPX, etc., ne peuvent en profiter.

Une autre approche consiste à travailler au niveau de l'octet afin d'agir sur la totalité des flux de données du réseau, quelle que soit l'application concernée. Les sections qui suivent dressent l'inventaire de l'ensemble des techniques d'accélération des flux de données.

Mécanisme de vérification et de correction des erreurs paquet

Afin de préserver l'intégrité des données et des transactions applicatives, il est possible de détecter les erreurs sur paquets et, à l'aide de mécanismes dédiés, de les corriger.

Les mécanismes de vérification et de correction d'erreur paquet s'appuient sur des fonctions mathématiques de calcul d'intégrité pour générer un résultat. La valeur obtenue est ensuite concaténée pour envoi avec l'élément de données, datagramme IP ou segment TCP ou UDP. À réception, une fonction mathématique inverse s'appuie sur l'information concaténée afin soit de contrôler l'intégrité de la donnée associée, soit de corriger les erreurs survenues lors du transport à travers le réseau.

Le contrôle sans correction d'erreur est surtout adapté aux réseaux haut débit et aux applications non-temps réel. Il consiste à demander à l'émetteur une retransmission sur détection d'erreur, ce qui suppose que l'application ne soit pas perturbée par le temps de latence généré par la retransmission.

Pour les applications temps réel, telles que les applications multimédias, et les environnements de réseau bas débit, il est plus intéressant de mettre en œuvre le mécanisme de correction d'erreur, qui évite les retransmissions sur le réseau. Inspiré des méthodes de correction d'erreur développées pour les réseaux satellite, il s'est adapté aux réseaux WAN classiques afin de réduire la bande passante consommée et, par voie de conséquence, l'investissement financier.

Dans l'idéal, les mécanismes de contrôle et de correction des erreurs doivent s'adapter aux différents profils de trafic à gérer, émis par la multitude des applications connectées au réseau. Par exemple, pour les applications multimédias temps réel, la retransmission n'est pas de mise, ni la correction d'erreur si elle est supportée par l'application elle-même. Certains protocoles multimédias intègrent par ailleurs un code correcteur, et il serait néfaste à la qualité de transmission du flux d'en superposer un autre.

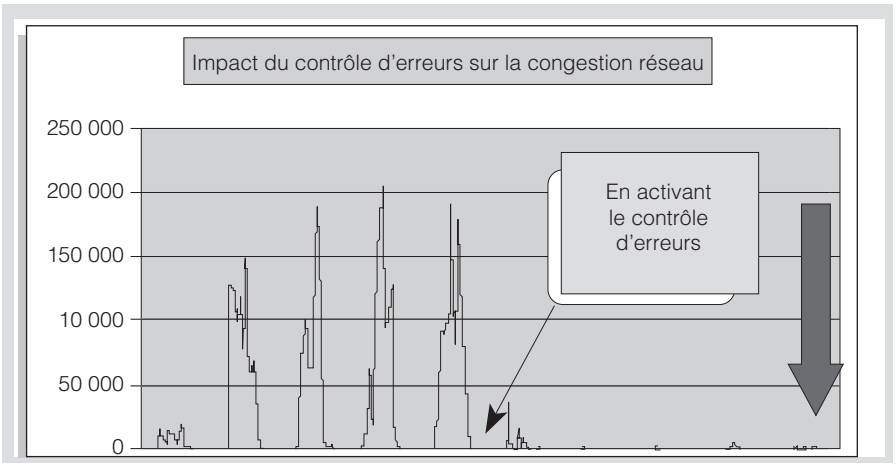
Pour le cas des flux s'appuyant sur TCP pour le transport, le mécanisme de demande de retransmission n'est pas obligatoire. Il peut toutefois être intéressant d'activer la fonction de correction d'erreur pour limiter le nombre de retransmission dans le réseau. Cela revient à insérer dans un paquet des informations de contrôle portant sur le paquet précédent et à autoriser à l'arrivée si nécessaire la reconstruction du paquet précédent à partir de ces informations. Bien que cette méthode consomme de la puissance machine, puisqu'il s'agit de travailler au niveau de chaque paquet, cet investissement en coût calcul machine est dans la plupart des cas largement compensé par le gain en performance et par l'élimination des pertes de paquets en transmission.

Une troisième option consiste à mettre en œuvre un mécanisme dédié de gestion de la retransmission de paquet en combinaison avec un code correcteur d'erreur, de façon à compenser la non-prise en compte de ces mécanismes par des protocoles tels qu'UDP.

La figure 4.2 illustre l'impact positif du contrôle d'erreur sur la congestion réseau.

Figure 4.2

Impact du contrôle d'erreur sur la congestion réseau



Les mécanismes de contrôle ou de correction d'erreur concourent à préserver la bande passante des abus de consommation dus à des erreurs et pertes de paquets lors des échanges de données, limitant la fréquence des retransmissions et des suppressions de paquets erronés.

Accélération WAN par cache

Le principe classique du cache consiste à placer en mémoire locale une quantité d'informations que l'on souhaite mettre plus rapidement à disposition des utilisateurs.

Comme le montre la figure 4.3, le cache peut travailler en local vis-à-vis des serveurs qu'il optimise afin de leur apporter une fonction de mémoire tampon, diminuant le nombre de sollicitations directes. Cette méthode est appelée cache serveur, ou reverse proxy. Une implémentation complémentaire consiste à travailler en cache vis-à-vis des utilisateurs dans le but de minimiser la quantité de données à transmettre à travers le réseau séparant les postes client des serveurs. Appelée cache client, cette méthode vise à mettre à disposition, au plus près des postes client, les informations fournies par les serveurs, améliorant ainsi les temps d'accès aux données, ainsi que la bande passante nécessaire à leur transport.

Dans les deux cas — caches client ou serveur —, les techniques employées sont limitées aux applications supportées par les solutions considérées. La plupart des solutions cache travaillent sur les flux Web mais peuvent également supporter les données provenant d'applications comme les serveurs de transfert de fichiers FTP, les serveurs d'information de type News ou encore les serveurs de flux vidéo de type streaming. Chaque support d'une nouvelle application nécessite un développement supplémentaire et une mise à jour logicielle de la machine serveur de cache.

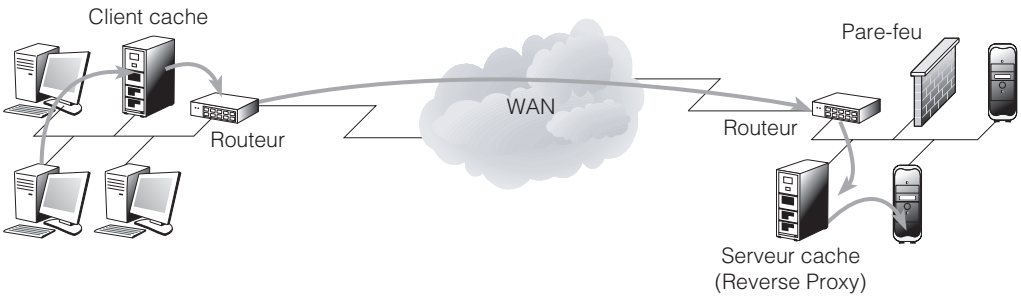


Figure 4.3

Techniques de cache traditionnelles client-serveur

Une analyse de l'environnement informatique d'une entreprise donnée montre que le paysage des applications qu'elle héberge est d'une grande diversité, générant une non moins grande variété de données transitant dans le réseau. Les répétitions de données étant très nombreuses dans un tel environnement, il est clair que les solutions de cache ne peuvent s'appliquer à la totalité des applications de l'entreprise, ce qui n'aide guère cette dernière à optimiser ses coûts et performances dans le WAN.

C'est pourquoi, en complément des implémentations de cache mentionnées précédemment, une nouvelle approche de cache plus optimisée permet d'accélérer les flux des applications d'entreprise à travers le WAN. Cette approche est défendue par toute une génération de jeunes pousses, comme Expand ou Peribit, persuadées qu'il existe une demande réelle des entreprises à solutionner le problème d'accès distant aux applications critiques tout en optimisant la consommation de bande passante.

Duplication des données

Les responsables informatiques en entreprise sont généralement surpris d'apprendre que plus de 90 p. 100 des données qui transitent sur leur réseau sont des répétitions. Les raisons à cela sont qu'une entreprise applique des processus communs de fonctionnement et que ses employés utilisent souvent les mêmes données (fichiers texte, tableaux, images, etc.). Qui n'a un jour, dans le cadre de son travail, repris un document existant pour servir de base à l'élaboration d'un nouveau document ?

Ce processus incrémental de génération de nouveaux objets dans le système d'information de l'entreprise participe de la duplication et de la répétition de données dans les différentes communications. Cela concerne en premier lieu les séquences de texte qui se répètent, comme les coordonnées de l'entreprise dans ses courriers, son logo ou encore le paragraphe de confidentialité inséré dans les documents sensibles.

Les éléments d'information dupliqués ne concernent pas que le texte. Avec l'avènement des moyens de communication multimédias, les images, le son et la vidéo définissent un autre ensemble de données répétitives qui transitent dans l'infrastructure réseau de l'entreprise, entraînant un gâchis dans la consommation de bande passante.

Avec l'évolution des modes de travail dans l'entreprise — télétravail, travail collaboratif —, les données dupliquées sont demandées plusieurs fois à travers le réseau par les utilisateurs. Lorsqu'un groupe de travail, par exemple, s'échange des documents *via* la messagerie, chaque document joint à un message est recopié autant de fois qu'il y a de destinataires à adresser puis envoyé dans le réseau avec toutes les copies nécessaires vers les utilisateurs concernés. Un autre exemple est l'accès à la base de données de l'entreprise, où les informations les plus demandées transitent mainte fois à travers le réseau sans qu'une optimisation soit apportée pour diminuer le volume de trafic répétitif.

Les applications informatiques elles-mêmes sont de nature à créer des duplications de données. Par souci de synchronisation et d'intégrité des données, le client applicatif a tendance à dialoguer fréquemment avec son serveur afin de mettre à jour ses données *via* l'infrastructure réseau. Les communications entre serveurs pour synchroniser leur base de données sont sources de redondance d'information transitant à travers les liens de communication.

Cette surconsommation de bande passante du fait de la redondance des données n'est pas grave en environnement LAN mais devient rapidement critique lorsqu'il s'agit de communications à travers le WAN, la bande passante n'étant plus gratuite.

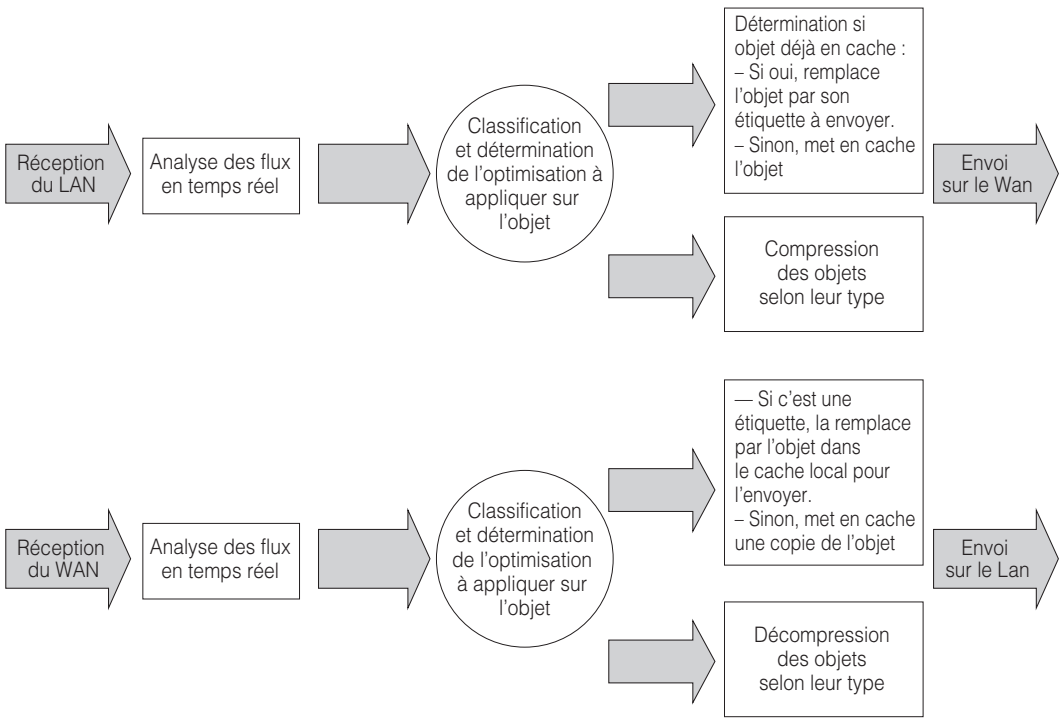


Figure 4.4

Séquence de traitements en optimisation

Pour pallier ces problèmes, il est possible d'analyser au plus près le train d'octets du trafic IP transitant à travers le réseau afin d'identifier les similitudes et les duplications de motifs pour les mettre sélectivement en mémoire cache.

Le mécanisme de cache d'applications d'entreprise combine souvent plusieurs éléments technologiques, incluant un moteur d'analyse de données, un cache sélectif et une technique de compression-décompression adaptative.

La figure 4.4 illustre ces différentes tâches d'optimisation entre l'émetteur et le récepteur.

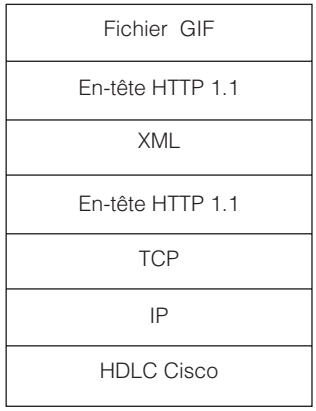
Lors de la réception du flux du réseau local, l'émetteur commence par analyser les caractéristiques de la donnée à traiter en optimisation. Cette phase d'analyse permet de classifier l'objet et de lui associer un mécanisme d'optimisation adapté. Il est alors possible de déterminer si une mise en cache de l'objet est nécessaire, combinée ou non à une compression afin de tenter de réduire sa taille. Enfin, l'objet est envoyé dans le WAN.

À réception par l'équipement homologue à l'autre bout du WAN, une analyse au fil de l'eau est effectuée afin de classifier à nouveau les éléments reçus et de déterminer les actions à appliquer. Si l'objet est compressé, il s'agit de le décompresser pour retrouver sa forme d'origine avant de l'expédier vers son destinataire dans le LAN. Si une étiquette est reçue à la place de l'objet réel — dans le cas du cache —, le récepteur remplace l'étiquette par l'objet correspondant qui se trouve dans son cache local pour l'envoyer vers son destinataire final.

Analyse des flux en temps réel

Dans l'ordre d'exécution des processus, le moteur d'analyse de données agit en premier. Il passe en revue en temps réel le train de données afin d'identifier les différents segments de protocoles et d'applications qui le composent. En règle générale, la segmentation effectuée a pour but de faire ressortir les différentes parties de la pile protocolaire, comme illustré à la figure 4.5.

Figure 4.5
Exemple de découpage en protocoles d'un flux de données



Le moteur d'analyse s'appuie généralement sur une base de connaissance protocolaire, avec une compréhension légère de certains protocoles. Le moteur peut ainsi compartimenter l'ensemble des flux puis les ventiler vers les autres mécanismes pour des traitements adaptés.

La base de connaissance évolue en fonction de l'apprentissage effectué sur les flux de données examinés. En d'autres termes, le moteur d'analyse apprend au fil de l'eau et détermine les motifs répétitifs. Les répétitions sont mises en mémoire cache dans l'ensemble des équipements d'optimisation à chaque extrémité du WAN et se voient associer un identifiant unique.

La base de données des correspondances objet-identifiant-étiquette est alors synchronisée entre tous les équipements de cache répartis dans le WAN.

Cache à la volée

L'algorithme proprement dit de cache doit être très sophistiqué afin de distinguer les données qui méritent d'être mises en mémoire pour réutilisation ultérieure — données répétitives — et les autres — données rares car pas souvent répétées. La plupart du temps, les données mises en cache sont des objets complets, comme un fichier d'image GIF, ou partiels, comme la palette de couleurs d'une image GIF, un code JavaScript encapsulé dans une page HTML, ou encore une image bitmap envoyée plusieurs fois à la demande d'utilisateurs travaillant sur le même contenu. Certaines solutions peuvent mémoriser des séquences d'octets et travailler en cache à un niveau de granularité de l'objet plus fin.

La figure 4.6 illustre le principe de fonctionnement d'un cache intelligent générique. Lorsque le moteur de cache identifie un objet intéressant à mettre en mémoire, il le copie pour un usage ultérieur et l'enregistre en lui assignant un identifiant unique. Dans le même temps, tous les moteurs à chaque extrémité du WAN synchronisent leurs caches afin d'avoir les mêmes objets en mémoire, ainsi que leurs références associées.

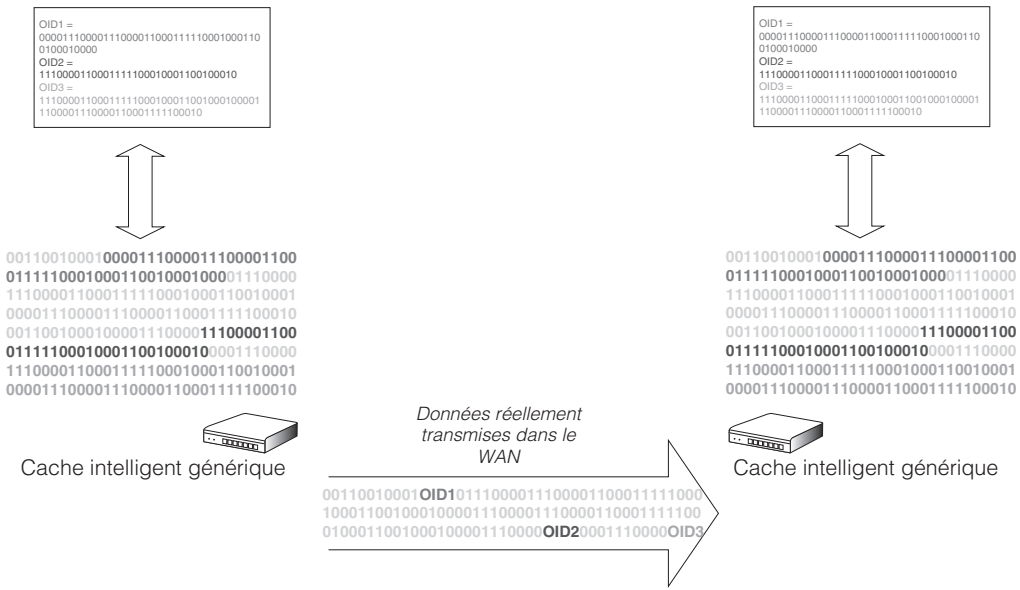


Figure 4.6

Fonctionnement d'un cache intelligent générique

S'il rencontre à nouveau l'objet, le moteur peut uniquement se contenter d'envoyer la référence à l'objet à la place de l'objet complet. À l'autre extrémité, son homologue est capable d'identifier rapidement l'objet et de le récupérer sur son disque local afin de l'attribuer correctement au destinataire. La restitution de l'ensemble du contenu nécessite un mécanisme intelligent capable de reconstruire l'ensemble des flux en temps réel en recombinant correctement les données envoyées réellement à travers le réseau WAN avec les données provenant de son cache local *(voir la section « Réassemblage intelligent avec restitution au destinataire », un peu plus loin dans ce chapitre).*

Accélération WAN par compression générique de données

La compression vient en complément du cache pour optimiser les flux présentant peu de redondance. La compression peut être plus ou moins performante selon le type de flux. Elle permet de réduire le volume de trafic à transmettre réellement à travers le réseau, même si le cache n'a pu agir dessus en optimisation.

Compression adaptative

La compression adaptative consiste à réduire le volume de données à transmettre à travers le réseau WAN en appliquant un algorithme mathématique adapté à la nature de la donnée à traiter. Pour être générique, la compression doit agir au niveau du paquet.

La compression adaptative des paquets est particulièrement puissante lorsqu'elle est employée en conjonction avec l'analyse dynamique des flux et le cache sélectif mentionnés précédemment. De moindre efficacité que le cache, notamment lorsqu'il travaille au niveau 3, la compression est uniquement activée sur les flux qui n'ont pu être mis en cache.

C'est au moteur d'analyse de données que revient la responsabilité de déterminer les types de données à compresser ainsi que le type de compression à appliquer. On distingue, par exemple, la compression du HTML de celle du SQL ou du JavaScript.

Réassemblage intelligent avec restitution au destinataire

Pour compléter la panoplie des mécanismes d'accélération générique des trafics IP, il est impératif de leur associer un mécanisme de réassemblage.

Le réassemblage consiste à restituer au vrai destinataire le trafic d'origine émis par la source réelle. Dans ce dessein, le mécanisme doit prendre en considération les différents objets à combiner provenant du réseau WAN (transmis à travers le réseau et souvent compressés) ainsi que ceux qui sont récupérés du cache local grâce à l'identifiant de l'objet transmis *via* le réseau. Bien entendu, le réassemblage intègre l'ensemble des mécanismes de décompression nécessaires pour restituer les objets.

Performances obtenues

Les techniques d'accélération générique de flux IP à travers un WAN donnent des résultats assez impressionnants. Le tableau 4.1 récapitule les données annoncées par les fournisseurs de ce type de solution, essentiellement Expand et Peribit).

Tableau 4.1 Performance de réduction en accélération générique de trafic IP

	Valeur moyenne de réduction de données sur le WAN	Valeur maximale de réduction de données sur le WAN
Annonce haute	85 %	95 %
Moyenne totale	65 %	90 %

La valeur moyenne de réduction de données sur le WAN se mesure en faisant le rapport entre le volume de flux de données en entrée de l'accélérateur et celui du flux réellement transmis dans le WAN en moyenne dans une journée. Ce paramètre ne suffit pas à lui seul car, idéalement, la réduction est à son maximum lorsque le volume de trafic à traiter est à son pic. Plus il y a de données à compresser et à cacher, plus les performances en réduction de volume sont élevées. C'est la raison pour laquelle le paramètre « Valeur maximale de réduction de données sur le WAN » est fourni.

Selon les applications et les contextes techniques plus ou moins idéaux pour les tests, on a distingué les valeurs hautes annoncées par les constructeurs des valeurs moyennes de toutes les valeurs fournies.

La valeur moyenne de 65 p. 100 indique que les solutions d'accélération peuvent plus que doubler la capacité de bande passante du réseau WAN. Pour un lien de 128 Kbit/s, l'accélération offre des capacités proches de 360 Kbit/s, soit 2,8 fois plus de capacité. Les valeurs maximales de réduction montrent qu'en cas d'échanges volumineux à travers l'infrastructure réseau, l'augmentation de capacité peut aller jusqu'à 10 fois la valeur du lien WAN en place. Dans une entreprise, les pics de trafic sont souvent constatés en début de journée, lorsque tous les employés se connectent au système d'information. C'est à ce moment que les performances doivent être au rendez-vous afin d'offrir un confort d'utilisation à l'utilisateur pour une productivité maximale.

Accélération des flux Web (HTTP et HTTPS)

Application développée au départ pour accéder aux informations réparties sur Internet, le Web, ou WWW (World-Wide Web), s'est complètement démocratisé pour tendre aujourd'hui à devenir un client utilisé pour accéder aux applications critiques de l'entreprise. Il suffit d'observer la multiplication des nouvelles technologies logicielles, comme les services Web, XML, SOAP (Simple Object Access Protocol), etc., pour mesurer la diversité des moyens qui permettent de transformer tout type d'application métier en une nouvelle version accessible *via* un navigateur Web.

L'entreprise peut transformer une application existante en lui ajoutant une interface HTTP ou développer une nouvelle mouture de l'application en s'appuyant sur ces technologies. Dans les deux cas, l'investissement est souvent assez lourd, aussi bien en temps qu'en ressources et moyens. Le coût n'est souvent pas négligeable. C'est pourquoi, malgré les bénéfices certains que peut procurer la migration, les entreprises ne s'empressent pas de se lancer dans ce type de projet. De premiers exemples de réalisation démontrent cependant que le processus est en marche et que cette tendance a de grandes chances de se développer à moyen terme.

Si l'on ajoute à cela que les grands éditeurs de logiciels d'ERP, de CRM, de systèmes de gestion des ressources humaines, etc., proposent tous une interface de développement pour client Web visant à simplifier le déploiement, la maintenance et la gestion opérationnelle du système d'information globale, on mesure aisément que le client navigateur Web va devenir le mode majoritaire d'accès aux applications critiques de l'entreprise.

Les points techniques structurants pour la généralisation de l'accès Web au système d'information de l'entreprise sont en premier lieu la sécurité et la performance. Dans la balance de décision d'adoption d'une application par l'utilisateur final, vient d'abord la performance. Comme expliqué précédemment, ce qu'on appelle de plus en plus l'expérience utilisateur, qui inclut rapidité d'exécution et d'accès à l'application et à ses données, est le critère d'évaluation par excellence de la qualité de l'application par son utilisateur. On estime que l'utilisateur ne tolère sur Internet qu'une attente inférieure à huit secondes, par exemple. Ses exigences sont toutefois supérieures lorsqu'il s'agit de l'accès à ses outils informatiques de travail.

Au même titre que la performance, la sécurité est cruciale pour le système d'information. La « webisation » de ce dernier ne fait qu'accroître ce besoin puisque les technologies Web permettent un accès de n'importe quel poste client aux applications critiques de l'entreprise.

Une véritable armada de technologies de sécurité sont à considérer afin de respecter au plus prêt les contraintes fortes de sécurité du système d'information. L'authentification, le contrôle d'accès, la confidentialité ou encore l'intégrité de données sont autant de services à mettre en œuvre afin de se conformer aux critères de fonctionnement de l'informatique de l'entreprise. Le protocole HTTPS (HTTP sécurisé) est justement conçu pour cela. Or la performance est un élément critique du protocole HTTPS.

Le navigateur Web client

L'entreprise qui souhaite homogénéiser l'accès à ses multiples applications doit tenter d'uniformiser la partie cliente pour une meilleure exploitation du parc informatique. Si le choix se porte sur le navigateur Web client, cela peut imposer de développer certaines applications spécifiques.

La plupart des éditeurs offrent la possibilité d'accéder aux serveurs en utilisant l'interface utilisateur du navigateur. Dans tous les domaines logiciels, que ce soit l'ERP (Enterprise Resource Planning), le CRM (Customer Relathionship Management), les systèmes de gestion de ressources humaines ou de gestion de base de données, le client peut être soit spécifique de l'application elle-même, soit un client Web standard. Le choix entre les deux n'est pas toujours évident. Même si le client propriétaire n'est pas universel, il reste souvent plus performant et mieux intégré.

L'architecture de la plate-forme applicative est composée d'un premier niveau de serveur Web pour gérer l'interaction entre le client et l'application, suivi d'un deuxième niveau, comprenant le serveur applicatif nécessaire à l'exécution de la logique opérationnelle, et de la zone de stockage et d'accès aux données, incluant le serveur de gestion de base de données.

L'accès aux différentes applications de l'entreprise devient simple en utilisant comme client le navigateur Web. Quel que soit l'éditeur considéré, essentiellement Microsoft et Sun iPlanet (ex-Netscape), le nombre de fonctionnalités ne cesse de croître. Les dernières versions supportent, par exemple, aussi bien des algorithmes de compression au fil de l'eau, généralement Deflate et

GZIP, que des méthodes de cache en disque local ou encore la capacité à exécuter des programmes encapsulés dans du HTML comme du JavaScript.

L'idée de certaines jeunes pousses de l'industrie, créées pour adresser ce marché de niche, consiste à s'appuyer au maximum sur ces capacités afin d'en tirer profit dans une optique d'optimisation des flux à faire transiter à travers le WAN.

Les données Web d'entreprise

Une page Web est par essence composite. La page est un assemblage de plusieurs objets de différents types. Les plus répandus sont le texte, aussi appelé source HTML, les images au format JPEG ou GIF et les codes de programmes en langage Java ou JavaScript, le source du programme étant dans ce dernier cas entièrement visible dans la page HTML.

Chaque objet de la page possède sa propre durée de vie, durant laquelle l'information qu'elle représente est pertinente. Par exemple, les données du tableau sont amenées à changer beaucoup plus souvent que les articles de presse ainsi que les photos associées.

Les données de la page qui ne varient pas dans le temps sont dites statiques, les autres étant des données dynamiques. La répartition entre objets statiques et dynamiques dans les pages Internet est de l'ordre de 80-90 p. 100 pour les premiers et de 10-20 p. 100 pour les seconds.

Cette répartition diffère pour les contenus Web d'applications d'entreprise, la part des éléments dynamiques augmentant dans certains cas jusqu'à devenir majoritaire par rapport aux éléments statiques.

L'optimisation de toutes les données Web peut tirer parti de cette différence entre les différents objets de la page.

L'approche HTTP

À l'image de l'accélération générique IP décrite précédemment, l'accélération Web HTTP se compose d'éléments équivalents, mais cette fois les mécanismes agissent au niveau 7 OSI, c'est-à-dire au niveau de l'application Web et de son protocole HTTP. Des jeunes pousses porteuses de cette approche sont, par exemple, Redline Networks et Crescendo Networks.

Cette approche permet de tirer parti des capacités d'optimisation de flux du navigateur afin d'atteindre des performances meilleures sans imposer d'équipement supplémentaire côté client. Le gain en termes de déploiement, de maintenance et de gestion est évident. Dans certaines solutions d'accélération *(voir ci-après),* il n'est même pas besoin d'installer sur le poste de logiciel client en plus du navigateur pour profiter de tous les avantages de l'accélération.

Le premier élément important est le moteur d'analyse de contenu, qui, ici, inspecte en temps réel le trafic afin d'identifier et de classifier les différents objets composant la page Web. Sont distingués les fichiers texte, image JPEG et les composants de programmes Java, JavaScript, etc. La classification a pour but d'associer un traitement d'optimisation *ad hoc* selon le type d'objet considéré.

La compression spatiale Web

La compression dite spatiale consiste à travailler en réduction de taille sur des objets au fil de l'eau, indépendamment les uns des autres et sans tenir compte des événements précédents.

La compression s'adapte au format de l'objet. Par exemple, le format texte fait l'objet d'une compression importante, que ce soit avec GZIP ou Deflate. Les images plus difficilement compressibles exigent l'application d'un algorithme de compression avec perte de qualité. Cette famille de méthodes de compression convertit l'image du format d'origine vers un autre de définition moindre. Par exemple, une image JPEG d'une résolution de 1 024 × 768 et en 16 millions de couleurs peut être transformée en une image 640 × 480 en 1 024 couleurs, avec au passage une réduction significative de la taille du fichier.

Le tableau 4.2 répertorie les performances que l'on peut obtenir en compression selon le type d'objet considéré.

Tableau 4.2 Taux de compression des objets composant la page Web (en %)

Type de contenu	Ordre de grandeur	Moyenne
HTML	300-5 000	2 000 (20 fois)
JPEG	120-500	200 (2 fois)
JavaScript	200-500	350
XML	300-500	400

Ces chiffres, donnés à titre indicatif, varient selon la puissance des matériels, les types de connectique proposés (100 Mbit/s, 1 000 Mbit/s) et les algorithmes de compression employés (Deflate est plus performant que GZIP).

L'inconvénient majeur de la compression spatiale est qu'elle ne détecte pas les redondances de données dans le temps. Dans certains cas, elle peut même s'avérer moins performante qu'un mécanisme de cache intelligent au niveau IP *(voir précédemment dans ce chapitre).*

Lorsqu'il s'agit de consultations Web sur Internet, la compression spatiale est performante pour réduire les volumes de trafic à envoyer à travers le WAN. Dans le cas d'applications intranet (CRM, ERP, etc.), en revanche, les pages contiennent plus d'informations répétitives, comme le logo de l'entreprise, le format du document, incluant souvent des champs de données alimentés par des requêtes sur une base de données. La répartition entre données variant peu dans le temps, ou données statiques, et données résultant d'une exécution de traitement, ou données dynamiques, joue plutôt en faveur des données dynamiques.

La compression temporelle Web

En complément de la compression spatiale, il peut être intéressant de mettre en œuvre une méthode de compression qui exploite le changement de valeurs selon le temps.

Pour illustrer la notion de compression temporelle, prenons l'exemple du MPEG (Motion Pictures Expert Group), le format standard pour la vidéo. La performance de MPEG s'appuie sur deux

méthodes de compression complémentaires, l'une spatiale, qui s'attache à compresser entièrement une image fixe, en prenant soin de mémoriser cette image comme référence, l'autre temporelle, qui consiste à comparer les images qui succèdent à celle de référence afin d'identifier les différences, c'est-à-dire ce qui a changé avec le temps. À l'affichage, le récepteur complète l'image de référence avec le delta transmis par l'émetteur pour restituer l'image d'origine.

C'est à peu près le même principe qui est utilisé dans le domaine Web pour la compression temporelle. La jeune pousse Crescendo a développé un algorithme en ce sens pour ses accélérateurs Web, qui repose sur l'analyse en temps réel des flux du réseau afin de détecter des points communs entre les pages Web qui y transitent. Ces pages servent de modèles de référence. Par exemple, dans une application CRM, la page principale reste toujours la même tout au long de la journée de fonctionnement, et seules les informations engendrées par des requêtes à la base de données diffèrent d'une consultation à une autre.

Pour tirer parti de ces nombreuses redondances d'information, l'algorithme de compression temporelle maintient une liste des pages les plus répétées et cherche en permanence à déterminer la page de référence optimale, c'est-à-dire celle qui est le plus souvent demandée. À chaque mise à jour de sa base de connaissance, l'algorithme pousse les nouvelles pages de référence vers les navigateurs pour les forcer à les mémoriser dans leur cache local. Ces pages de référence sont les données statiques à traiter. Cela sous-entend évidemment que le navigateur client possède une fonction de cache local.

Il ne reste plus qu'à envoyer les objets dynamiques, qui viennent compléter la page de référence. Les données dynamiques sont compressées selon leur type puis transmises dans le réseau vers le destinataire.

La figure 4.7 illustre chaque étape du processus de compression temporelle.

Pour le réassemblage, un programme JavaScript, également envoyé par l'accélérateur Web vers les clients, récupère les objets delta reçus du WAN et les combine avec la page de référence utilisée afin de restituer la page complète de manière transparente pour l'utilisateur.

Il est évident que la compression temporelle combinée à la compression spatiale surpasse une simple compression spatiale, surtout lorsque le trafic de données présente un grand nombre de redondances des objets dans le temps. Dans un cadre intranet, la compression temporelle affiche des performances plus importantes qu'une approche spatiale.

En addition à cette astuce judicieuse de compression de flux WEB, Crescendo Networks a fait le choix unique dans ce domaine de l'accélération de flux de construire des solutions matérielles en se basant sur les puissances de calcul apportées par des processeurs réseau (Network Processors, ou NP). Ces NP permettent de gagner en performance par rapport à l'approche traditionnelle reposant sur des processeurs généralistes (du type INTEL Xeon, par exemple), tout en offrant une évolutivité des fonctionnalités beaucoup plus souple comparée à une approche reposant sur des processeurs ASIC (Application Specific Instruction Code), qui offrent peu de possibilités d'évolution.

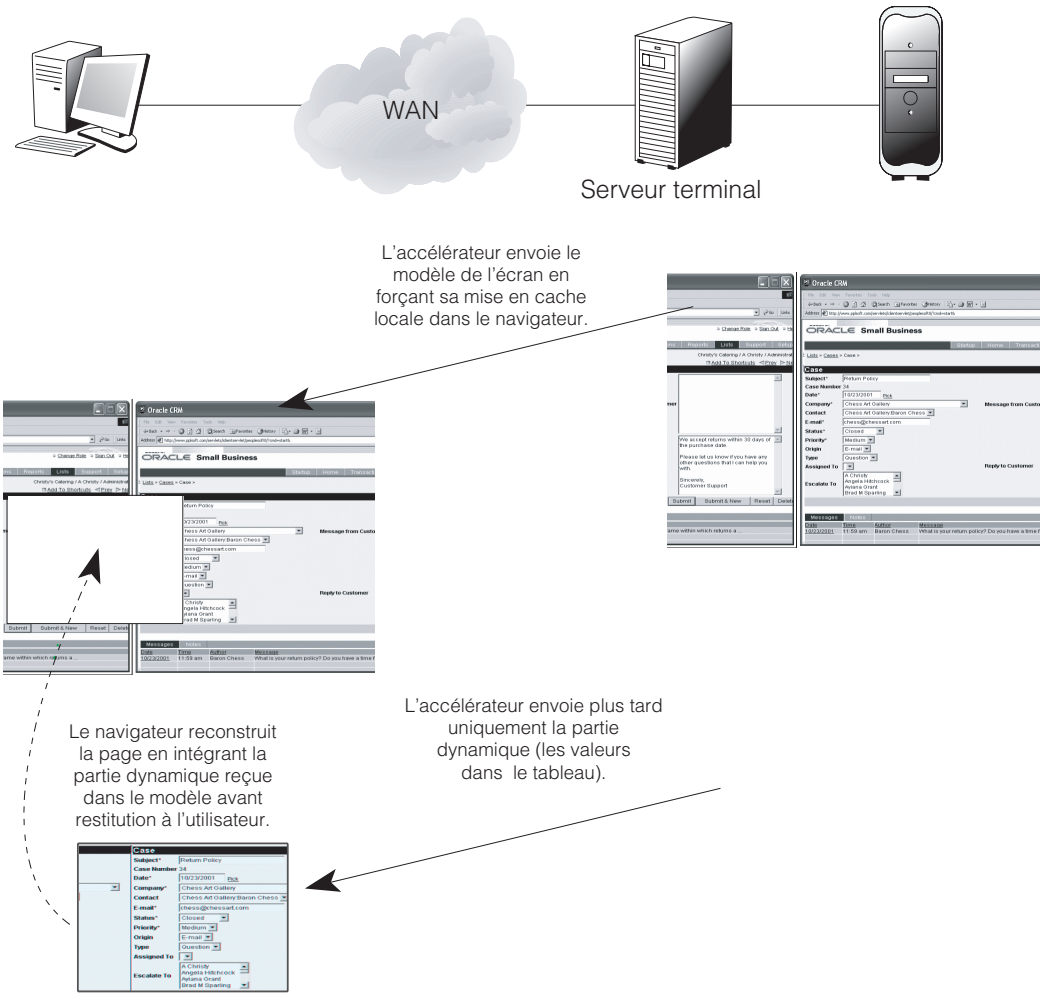


Figure 4.7

Fonctionnement de la compression temporelle Web

L'approche HTTPS

La plupart des équipements d'accélération Web complètent leur panoplie de protocoles supportés en ajoutant à HTTP sa version chiffrée HTTPS au moyen du protocole SSL (Secure Sockets Layer). Cela permet de traiter les flux chiffrés provenant des serveurs Web avec les mêmes performances qu'en HTTP.

Le protocole HTTPS peut être utilisé par les applications de commerce en ligne dans un environnement Internet grand public, mais son principal usage reste les intranets et extranets de

l'entreprise, qui permettent d'exploiter le réseau public pour relier les utilisateurs aux applications critiques. On parle parfois en ce cas de VPN à la demande.

L'un des principaux atouts de HTTPS, qui explique en grande partie son succès en entreprise, est son intégration dans le navigateur, qui n'impose pas ou peu de logiciel supplémentaire pour l'établissement de sessions totalement sécurisées.

Son déploiement est en outre plus simple et plus rapide qu'un VPN IPsec, lequel nécessite l'installation d'un client IPsec sur le client, ou qu'un VPN destiné à connecter les clients au site central des serveurs. Des soucis de gestion de versions des clients IPsec selon les systèmes d'exploitation considérés des postes client caractérisent les aléas majeurs de cette solution. HTTPS élimine cette contrainte.

L'approche HTTPS nécessite de fournir des services de connexion rapides, hautement sécurisés et robustes. Comme expliqué précédemment, les inconvénients du protocole HTTP se retrouvent dans le protocole HTTPS et s'ajoutent à ceux du protocole SSL de sécurisation des sessions.

Pour accélérer les sessions HTTPS, il faut agir aussi bien sur la session SSL de transport sécurisé des flux qu'au niveau des données HTTPS à transmettre à travers les tuyaux chiffrés.

Accélération du protocole de transport SSL

À l'image de TCP, l'établissement d'une session SSL nécessite plusieurs échanges entre le client et le serveur. Cette phase d'établissement de session sécurisée a pour but de déterminer les clés de chiffrement des sessions. Elle consiste à négocier dans un premier temps les clés de session puis à les échanger afin de démarrer les dialogues sécurisés.

Cette phase d'établissement est la plus gourmande en consommation CPU. Sur un serveur qui ne possède pas de carte accélératrice HTTPS, c'est au système d'exploitation de traiter les sessions. Il s'ensuit une rapide dégradation des performances de la machine, qui peut aboutir à un blocage de ce dernier.

Pour remédier à ces désagréments, des solutions matérielles dédiées ont été développées. La carte adaptateur réseau intégrant des capacités de traitement SSL est une de ces solutions. Elle reste toutefois limitée en matière d'extensibilité et peut s'avérer très coûteuse en cas d'architecture multiserveur.

Une autre solution consiste en la mise en place de boîtiers externes dédiés aux traitements SSL-HTTPS. Dans l'architecture illustrée à la figure 4.8, le boîtier se positionne près des serveurs qu'il gère et termine l'ensemble des sessions SSL-HTTPS provenant du client distant.

Les échanges de données entre le client distant et le boîtier de traitement SSL-HTTPS se font en toute confidentialité grâce au chiffrement des sessions. Du côté du réseau local, les communications avec les serveurs se font en clair. Si nécessaire, certaines solutions autorisent le chiffrement des sessions internes avec les serveurs pour une plus grande confidentialité au sein du réseau d'entreprise. Le choix se porte alors sur une configuration statique figée de la clé de chiffrement à utiliser de façon à ne pas trop détériorer les performances des sessions.

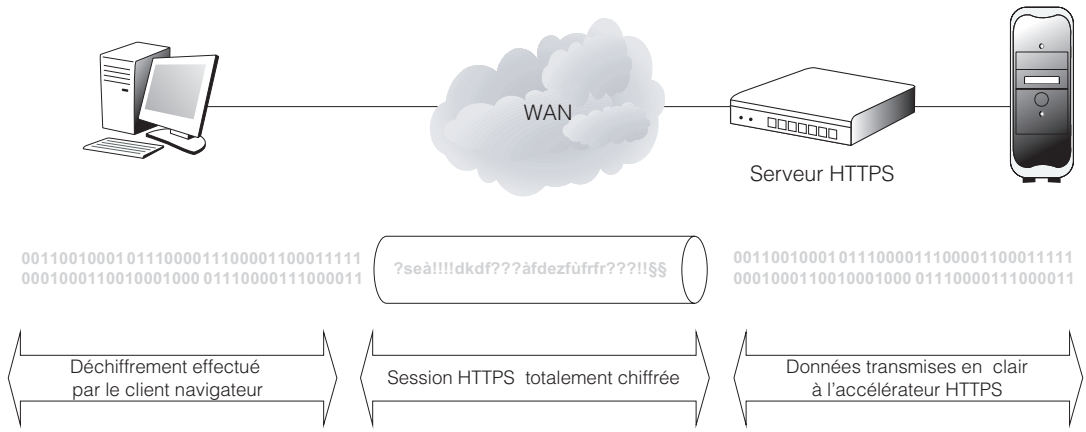


Figure 4.8

Accélération HTTPS via *un équipement dédié*

Accélération des échanges HTTPS

Le fait de soulager les serveurs du traitement contraignant du SSL garantit une meilleure accélération de bout en bout du temps de réponse de l'utilisateur.

Certaines offres d'accélération HTTPS proposent d'aller encore plus loin dans l'optimisation en ajoutant à la gestion des sessions SSL par du matériel dédié la combinaison des mécanismes d'accélération déjà employés sur les flux HTTP. Autrement dit, avant d'être injectés dans le tuyau chiffré, les flux HTTPS subissent les mêmes traitements d'optimisation que les flux HTTP, à savoir compression spatiale et temporelle, forçage de cache, etc., de façon à bénéficier des mêmes gains de performances.

L'approche VPN-HTTPS

La technique consistant à s'appuyer sur le protocole HTTPS afin d'offrir aux utilisateurs distants un moyen d'accès aux applications critiques de l'entreprise procure des avantages indéniables par rapport à une approche VPN IPsec classique au niveau réseau.

Les contraintes bien connus des VPN IPsec, telles que l'obligation de déployer soit un client logiciel VPN, soit un matériel sur le site client et de le maintenir et de le gérer en fonction de l'évolution des versions des logiciels, des systèmes d'exploitation, etc., sont complètement éliminées dans la solution de VPN en HTTPS.

Cela n'est toutefois possible que si toutes les applications critiques joignables de l'extérieur sont développées en technologie Web. Or la réalité est bien différente. L'entreprise vit sur son historique applicatif, composé d'applications métier développées en interne et de progiciels de toute sorte qui ne peuvent basculer du jour au lendemain en accès Web. Il suffit de citer comme exemple les applications hébergées sur les serveurs centraux de type mainframe, les applications sur des serveurs UNIX, etc.

Dans ces cas, il convient d'élargir le nombre de services accessibles *via* des sessions HTTPS. L'utilisateur, qui n'a besoin que de son navigateur, peut, grâce à un portail hébergé sur un serveur dédié, ou appliance, accéder à une page d'accueil lui donnant la possibilité d'utiliser toutes les applications *via* l'interface Web sécurisée.

L'appliance s'interface avec les différentes applications *via* des protocoles comme RDP (Remote Display Protocol) pour les serveurs Microsoft Terminal Server, X.11 pour les applications s'exécutant sur plate-forme UNIX, ou encore Telnet en émulation 3270 pour les serveurs mainframe IBM.

Cette solution évolue pour intégrer de nouvelles interfaces de communication avec de nouvelles applications et services internes, tout en les adaptant au protocole HTTPS pour les rendre accessibles aux utilisateurs Web.

Accélération des applications critiques

On appelle applications critiques les applications indispensables au bon fonctionnement de l'entreprise. L'efficacité et la productivité de l'entreprise dépendent de l'efficacité de ces applications, véritables piliers du système d'information.

La multiplication des applications d'entreprise a conduit les directeurs informatiques à prendre conscience qu'il était important d'homogénéiser l'ensemble du parc informatique en démarrant par les clients applicatifs. Faute de cela, la gestion et la maintenance du parc informatique, où chaque application possède son propre client propriétaire, peuvent devenir un cauchemar.

Des cabinets d'étude tels que le Gartner Group ont mis en évidence l'importance du concept de TCO (Total Cost of Ownership), ou coût total de possession. Leurs études ont montré que le coût de fourniture, de déploiement et de maintenance des applications informatiques pour un groupe d'utilisateurs donné dépassait largement le coût d'acquisition initial d'un PC et des logiciels. Les évaluations récentes du même Gartner Group évaluent le TCO d'un PC en réseau exécutant un système d'exploitation Windows 98 à environ 50 000 euros sur ses cinq années de vie.

Les composantes de ce TCO sont les suivantes :

- investissement en matériel, réseau et logiciels (environ 30 p. 100) ;
- gestion du système et des réseaux ;
- support technique (environ 16 p. 100) ;
- coûts générés par les utilisateurs finals, tels que perte de temps et diminution de la productivité d'un utilisateur tentant de résoudre lui-même le problème technique (jusqu'à 40 p. 100).

Face à un tel constat, il existe aujourd'hui deux possibilités de client universel : le navigateur Web et le client fin. L'approche Web consiste à mettre en œuvre des architectures à plusieurs niveaux, appelées multitiers. Cette solution n'est néanmoins pas la plus courante, car elle impose comme contrainte forte d'investir en développement logiciel pour basculer les applications en accès Web.

La solution client fin n'impose aucune contrainte pour l'application cible. Les deux fournisseurs principaux de ce type de solution sont Citrix, avec Metaframe, et Microsoft, avec .Net.

Cette solution vise les objectifs suivants :

- permettre un déploiement facile et automatisé des applications ;
- offrir des moyens centralisés de supervision et d'administration ;
- banaliser le poste de travail et éviter la surenchère perpétuelle de ressources, telles que mémoire, CPU ou bande passante réseau, imposées par les nouvelles applications.

Elle consiste à insérer une architecture client-serveur destinée à banaliser le poste de travail. Dans ce modèle, les applications sont déployées, gérées, supportées et exécutées sur le ou les serveurs d'applications.

Le modèle serveur centralisé utilise un système d'exploitation multiutilisateur permettant à plusieurs utilisateurs de se connecter simultanément à un serveur et d'exécuter des applications dans des sessions indépendantes protégées. Ce modèle utilise une technologie permettant de séparer la logique applicative de l'interface utilisateur et de distribuer la présentation de cette interface vers un poste client. Dans le cas de Citrix, le protocole utilisé est ICA (Independent Computing Architecture).

Contrairement à l'approche Microsoft, qui consiste à n'adopter que la seule famille des postes clients Microsoft, Citrix présente l'énorme avantage de supporter tout type de poste client. Cela permet de gérer un environnement totalement hétérogène, en installant sur chacun des postes le client ICA, gratuit, correspondant au système d'exploitation client.

Selon le cabinet Zona Research, avec ce modèle de client fin le TCO peut être réduit de 57 p. 100 sur cinq ans.

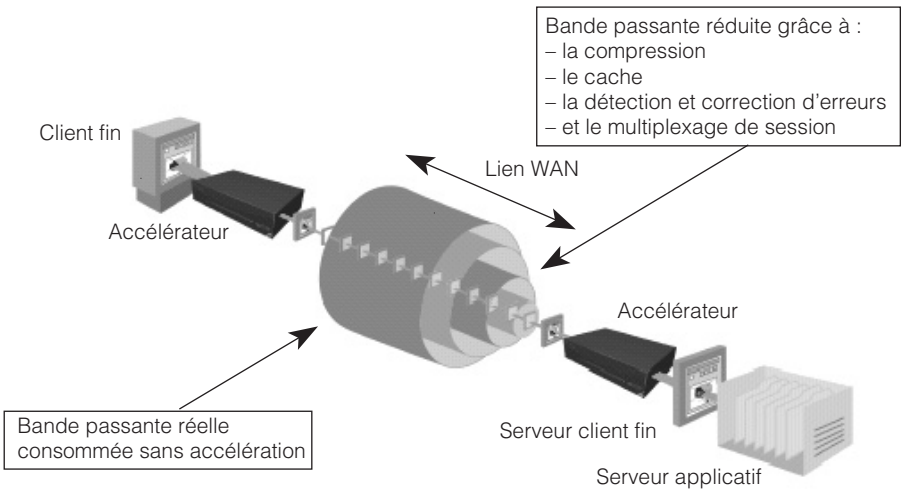
Concernant le réseau, seuls les rafraîchissements d'écran, les saisies clavier et les clics de souris sont transmis. La technologie client fin se comporte très bien dans un environnement de réseau local LAN. Ses performances baissent lorsque l'environnement réseau possède des bandes passantes moindres. Des ralentissements perceptibles et parfois inacceptables pour l'utilisateur se ressentent lorsque le poste client est séparé du serveur par un réseau WAN bas débit ou très chargé, du fait qu'un grand nombre de sessions simultanées y transitent. Force est cependant de constater que dans la plupart des déploiements, les clients fins sont séparés des serveurs par un réseau longue distance WAN souvent bas débit.

Les paramètres techniques à surveiller sont le débit moyen et le volume total de données à transmettre, car ils déterminent le temps de réponse ainsi que la durée d'exécution des sessions applicatives.

Pour optimiser l'ensemble des flux client fin, il est nécessaire de mettre en œuvre les mécanismes génériques d'accélération décrits précédemment dans ce chapitre, et ce à chaque extrémité du lien WAN (côtés agence distante et siège social), comme l'illustre la figure 4.9.

En complément de ces mécanismes, qui traitent aussi bien les flux client fin que les autres flux IP, il est intéressant d'ajouter des mécanismes spécifiquement développés pour accroître l'accélération des flux client fin. L'approche souvent adoptée pour cela consiste à coupler aux mécanismes de compression et de cache génériques, un outil de gestion de bande passante.

Figure 4.9
Architecture d'accélération des clients légers



L'une des raisons du mauvais fonctionnement des clients fins réside dans la cohabitation anarchique des applications. Par exemple, un utilisateur distant peut être amené à ouvrir plusieurs sessions, telles que son client fin, son Web et son client FTP, pour télécharger un gros fichier. Chaque application obtient dès lors une quantité de bande passante arbitraire, ne correspondant pas le plus souvent à ses besoins. Les applications les plus importantes peuvent de la sorte se retrouver dépourvues de débit réseau et ne plus fonctionner.

La gestion de bande passante est donc indispensable, en sus de la compression et du cache, pour prioriser les différents flux selon leur niveau de criticité et leur garantir la bande passante nécessaire à leur fonctionnement optimal.

Reste un cas pour lequel il n'existe pas de solution pour résoudre un dysfonctionnement : celui où le client fin est employé pour accéder à l'ensemble des applications de l'entreprise, critiques et non critiques. La distinction entre les applications devient plus difficile car il faudrait pour cela analyser à l'intérieur du tuyau de client fin les différentes sessions applicatives. Or, aujourd'hui, les mécanismes de différentiation de trafic ne travaillent qu'aux niveaux 3 et 4 OSI, ce qui ne leur donne aucune possibilité d'analyser à l'intérieur d'une session applicative client fin.

Le futur défi de ces acteurs de la QoS sera de répondre aux besoins réels d'analyse et de différenciation des trafics au niveau applicatif.

En résumé

L'accélération de flux consiste non seulement à améliorer le temps de réponse utilisateur mais aussi à réduire le volume de données transmis à travers le réseau longue distance WAN. Des mécanismes tels que la compression ou la mise en cache ont pour rôle d'éviter de transmettre la totalité du trafic à travers le WAN. Ces mécanismes sont particulièrement adaptés lorsque la bande passante est limitée (accès bas débit, par exemple) ou chère (liens WAN internationaux, par exemple).

Plusieurs mécanismes ont été imaginés par de jeunes pousses plus imaginatives les unes que les autres pour atteindre ces objectifs. Ces mécanismes peuvent travailler aussi bien au niveau réseau qu'au niveau applicatif.

L'intérêt de ces solutions dépend au finale du coût de la bande passante WAN. Même lorsque cette dernière est chère ou limitée, l'accélération de flux reste utile pour l'optimisation des performances.