日本ソフトウェア科学会 (和文・英文) 投稿用表紙

原稿種別 (該当するものに×印)

□論文　□特集用論文 (　　　特集)

□チュートリアル　□ソフトウェア評論

□トピックス　☒フォーラム

□その他 (　　　)

題　目 インターネット防火壁の基礎技術と応用

– DeleGate の仕組み –

英文題目 Fundamental Technologies and Applications of Internet Firewalls

– Construction of DeleGate –

著者氏名 1 佐藤豊　会員番号

2

3

4

連絡著者および連絡先

氏　名 佐藤豊

宛　先 〒305 茨城県つくば市梅園 1–1–4

所　属 電子技術総合研究所

電話番号 0298–XX–XXXX 内線

Fax 番号 0298–XX–XXXX

Email Address ysato@etl.go.jp

投稿年月日 1996 年 XX 月 XX 日

刷上りページ数 [†] 約 7 ページ

事務局用

受付年月日　年　月　日

受付番号

[†] 2,000 字 (25 字× 80 行) で刷上り約 1 ページとなります．

フォーラム

標題

インターネット防火壁の基礎技術と応用
– DeleGate の仕組み –

Fundamental Technologies and Applications of
Internet Firewalls – Construction of DeleGate –

著者名及び所属

佐藤豊
Yutaka Sato

電子技術総合研究所
Computer Science Division, Electrotechnical
Laboratory (ETL)

概要

インターネットに接続する組織を侵入や攻撃から守るための、防火壁と呼ばれる防御システムが普及している。防火壁はIPルータと代理サーバの組合せで構成される。本稿では、防火壁の基本構成を概説し、電子技術総合研究所で開発され広く利用されている代理サーバである「DeleGate」について、そのマウント機能をはじめとする特徴的な防火壁機能を解説する。代理サーバの応用分野は、防火壁の構築に留まらない。応用プロトコルの中身を理解しながら中継する際にデータ変換やプロトコル変換を行い、付加価値を加えることにより、様々な応用可能性を持っている。広域網上での通信の効率化等の多様な応用例を挙げながら今後の研究開発課題を述べる。

キーワード

セキュリティ、防火壁、代理サーバ、応用ゲートウェイ、WWW、DeleGate

1　はじめに

インターネットに接続する組織は、インターネットからの侵入者によって組織内の情報・計算資源を荒らされるのを防ぐために、様々な関門や防護柵を築いて来た。今日、それらのセキュリティ技術はファイアウォール（防火壁）技術と総称されるようになり、体系化・標準化が進み[1] [2] [3]、商品化され、日常風景の一部となりつつある。

一般に防火壁は、インターネットプロトコル (IP) レベルの経路制御およびフィルタリングに、アプリケーション (応用) プロトコルレベルのフィルタリングを組み合わせて実現される。前者はルータ等の専用装置により実現され、後者はソフトウェアで実現されるのが普通である。前者は情報の容器を、後者は情報の内容を、それぞれに扱う。

クライアントとサーバとの間で交わされる応用プロトコルを中継するそれらのソフトウェアは代理サーバ (proxy server)、あるいは応用レベルゲートウェイと呼ばれる。応用層において応用プロトコルの構文や意味を理解して中継することにより、IP 層では得られないきめ細かで柔軟なフィルタリングや経路制御を加えることができる。さらに、中継に伴って積極的に付加価値を加えることができ、ここに様々な応用可能性が生まれる。

さて、筆者は 94 年 3 月以来、DeleGate と名付けたシステムを開発してきた[7] [8]。これはインターネット上の多様な応用プロトコルを中継するシステムであり、代理サーバとして防火壁を構築するための基礎的な機能を備えている。ただ、このシステムの開発を始めた動機はセキュリティを強化することではなかった。当時、未熟な防火壁装置・技術によるセキュリティ保護のために利用者の利便性が犠牲にされていた状況があり、利用者の一人として自らの利便性回復のための道具を作り始めたのがきっかけである。その後の World Wide Web (WWW) の急速な普及に伴い、DeleGate は主に HTTP (Hyper Text Transfer Protocol) 用の代理サーバとして広く利用されてきた。他の HTTP 専用の代理サーバと異なり、DeleGate では多様な応用プロトコルに対して共通に適用できる、一般的な中継機能や付加価値機能を見出すことに開発の力点を置いており、そこに DeleGate 独自の機能が生み出されている。

以下では、防火壁の実現方式と、DeleGateの機能を概観し、特に応用プロトコルレベルでこそ実現可能な機能についてDeleGateで実現されている防火壁機能を紹介し、最後に今後の研究・開発の課題を述べる。

2　防火壁とその実現方式

2.1　防火壁への要求 – 安全性と利便性の両立 –

防火壁は、互いに接するネットワークの間に置かれる通信のフィルタ装置である。悪意あるいは事故によって発生し得る、何らかの害を及ぼす通信が他のネットワークから侵入、あるい他のネットワークに侵出しないよう、濾過して除去するのが目的である。

一方防火壁は、強力なセキュリティを実現しさえすればそれで良いというものでは無い。一般にセキュリティシステムでは、安全性を強化するほど、利用者にとっての利便性を損なう。もともと便利に使うためのネットワークが、必要悪のような防火壁のためにがんじがらめにされ使いにくくなってしまって本末転倒である。安全性と利便性をいかに両立させるか、トレードオフするかが問題である。

従って、防火壁への要求としては、より強力なフィルタリング能力と、利用者から見た透明性の両立が挙げられる。フィルタリング能力としては、

・より高速な中継性能

・より高度な判定能力 (より豊富な判断材料)

が望まれる。利用者から見た透明性としては、

・利用者が防火壁の存在を意識しないで済むこと

・クライアントのプログラムを変更せずに済むこと

が望ましい。また、せっかく通信性能を劣化させてまで導入するからには、必要悪として存在するのではなく、

・より豊富な付加価値を加えられること

が望まれる。

2.2　防火壁の基本構成

上記のような要求に応えるために、まずルータ等により全般的なフィルタリング付き中継を高速に実現し、その上で代理サーバにより個々の応用や利用者の特殊なケースをきめ細かに制御する、というのが基本的な方針となる。インターネットにおける防火壁の基本的な構成は、以下のような段階で表すことができよう。

(0) 隣接するネットワークの間で、IPパケットの直接的な通過を部分的あるいは全面的に禁止する

(1) どのネットワークとも直接的に通信可能なホスト、あるいはネットワークを置く(これが防火壁)

(2) 防火壁上のホストにおいて、直接的な通過を禁止されたプロトコルに対して、応用プログラムにより中継を行う(これが代理サーバ)

(3) 中継に際して、通信プロトコルの変換、コマンド単位のフィルタ、データのキャッシュなどを行う

防火壁の具体的な構成方法は、使用するハードウェアやプラットフォームにより様々である[1] [2] [3]。一般にルータにおいては、IPパケット中の以下のような情報に基づき、通過の許可/不許可を決めてフィルタリングを行う。

・送信者/受信者のIPアドレス
・転送プロトコルの種別(TCP,UDP等)
・応用プロトコルの種別(TCP,UDPポート番号)
・送信/受信、TCPのコネクションの接続方向

2.3 代理サーバとクライアント

代理サーバは、クライアントとサーバとの間にあってその間の通信を中継するサーバである(図4-a)。

代理サーバは2種類に大別できる。まず、応用プロトコルの内容を見ないで、転送プロトコルレベルで得られる情報(クライアントやサーバのIPアドレスやポート番号等)だけを用いて中継を制御するもの(Socks, udprelayなど)。これらは「サーキットレベル」の代理サーバと呼ばれる。一方、HTTP用代理サーバ、FTP用代理サーバ、というように、個々の応用プロトコルの内容を解釈しながら中継するもの(CERN-HTTPD [4]やDeleGateなど)。これらは応用レベルの代理サーバと呼ばれる。

サーキットレベルの代理サーバは、応用プロトコルによらず中継でき、汎用である。ただし機能的には中継サーバとして以上の発展性は無い。一方応用レベルの代理サーバは、各応用プロトコル毎に作る必要がある。そのかわり、プロトコルの内容を理解しているため、応用プロトコルのコマンド単位でアクセス制御を実現できる。さらに、各種の付加価値、例えばキャッシュ、プロトコルの変換、データの変換、複数サーバの統合、など

を実現することができる。

サーキットレベルの代理サーバに対応するためには、クライアント側のプログラムに手を加えて、代理サーバとの間で目的のサーバに関する情報をやりとりするプロトコルを実行することが必要である。一方応用レベルの代理サーバでは、中継する元の応用プロトコルに対して互換性を保ってプロトコルを拡張することで、クライアントに手を加えずに拡張が可能な場合が少くない。例えば、FTP(ファイル転送プロトコル)や、POP(メールボックス管理プロトコル)では、接続の最初にユーザの認証を行うコマンドをクライアントからサーバに送信する。この時送られるユーザ名の形式は元のプロトコル上で、任意の文字列としてしか規定されていない。従って、通常は目的のサーバに以下の(1)の形式で送られるものを、(2)のような形式に拡張しても差し支えない。

(1) USER ユーザ名

(2) USER ユーザ名@server

代理サーバは(2)のようなコマンドを受け取って、目的のサーバ server に接続し、(1)の形式のコマンドを目的のサーバに中継すれば良い。

HTTPプロトコルの場合には、もう少し複雑である。HTTPでは、各種情報資源を、URL (Uniform Resource Locators) により、一様な形式で一元的に識別する。例えば 以下の(0) のようなものである。これは、HTTPサーバである server において、path で識別される資源を表す。代理サーバを使用しないクライアントは、server に直接接続して(1)の形式の要求を送る。一方、代理サーバ経由の場合には、代理サーバに接続後(2)の形式のコマンドを送る。従って、クライアントはこの区別に対応しなければならない。

(0) http://server/path

(1) GET /path

(2) GET http://server/path

これに対してDeleGateでは(2)の形式が普及する以前、これに未対応なクライアントのための中継を実現する必要があった。そこで、(0)に対してDeleGateを介してアクセスすることを明示的に表す(3)のような形式を導入した。この(3)は(0)の互換形式になっているので、クライアントには変更は必要ない。クライアントはdelegateをHTTPサーバと見て、(4)のような要求を

送出するのである。

(3) http://delegate/-_-http://server/path

(4) GET /-_-http://server/path

これを受け取ったDeleGateは、自身が代理サーバとして利用されていることを判別し、server に接続して (1) のような要求を送る。この方法による中継ではさらに、サーバからクライアントへの応答中に含まれる (0) の形式の URL を (3) の形式に書き換えなければならない。これは、この代理サーバ方式が機能しつづけるためには、クライアントからは常に (0) が (3) のように見える必要があるからである。

クライアントに手を加えずに済み、利用者毎に多様な手持ちのクライアントをそのまま利用できるということは、実用上大きな利点である。一方で、それは利用者に代理サーバを使用していることを意識させることになり、利用者から見た透明性の面では劣る。ただし、FTPやPOPの例のように、上位互換の拡張が利用者から見て自然ならば問題は少い。さらに、そのような自然な拡張をクライアントのプログラムがサポートするようになれば、代理サーバへの対応の移行が円滑に行えて理想的である。

2.4　サーバのための代理サーバ

これまで述べた代理サーバは、クライアントの代理として動作するものである (図 4-b(1))。クライアントは何らかの形でそれを代理サーバとして意識して利用している。これに対して、目的サーバの代理としての代理サーバの利用方法もある (図 4-b(2))。これは、クライアントからは目的のサーバそのものとして見え、代理サーバであることは意識されないものである。

サーバ側にもクライアント側と同様な、代理サーバへの要求がある。例えば、防火壁内のサーバの公開、負荷分散のために代理サーバのキャッシュを用いた並列ミラーサーバを提供する、などである。 WWWクライアントの日本語対応が進む以前の一時期、 DeleGateはサーバ側において、同一テキストに対する複数の日本語コードを提供するために用いられることも多かった。

クライアントのための中継においては、代理サーバは自身の存在が利用者に意識されないようにすることが求められるが、サーバのための中継においては、実際の

サーバ (複数かも知れない) が利用者に見えないようにし、 DeleGate 自身がサーバであるように見せることが求められる。このために DeleGate は、次章に述べるような、実際のサーバを隠蔽する「マウント機能」を実現している。これは、前節に述べた、要求 / 応答の中継に際しての書き換えにより実現されている。

クライアント側に従属する代理サーバ、サーバ側に従属する代理サーバの他に、他の代理サーバのための代理サーバという利用形態もある (図 4-b(3))。多段の防火壁で自然に生じる他、網状に接続して冗長経路における経路制御を行ったり、階層的に接続してキャッシュの階層化を行うなどの応用ができる。

3 DeleGate の機能概要

3.1 中継する応用プロトコル

DeleGate が応用レベルの代理サーバとして内容を解釈しながら中継を行っている応用プロトコルとしては、表 1(a) のものがある[10]。 DeleGate はまた、サーキットレベルの代理サーバとして、任意の TCP/IP 上の応用プロトコルを単一の目的サーバ向けに透過中継する機能、および Socks サーバとしての中継機能を持つ。この他、 DeleGate は、表 1(b) のデータ形式やプロトコルを解釈・使用する。

3.2 主な機能

DeleGate の提供する基本的な機能としては、以下のようなものがある。

代理サーバ

クライアントと実際の (任意の) サーバとの間を中継する (HTTP,FTP,POP,Telnet,Gopher)

アクセス制御

(クライアントホスト, サーバホスト, プロトコル) の組合せごとに、中継の許可 / 不許可を選択する。 Ident プロトコルによる利用者単位の認証にも対応している。

経路制御

(クライアントホスト, サーバホスト, プロトコル) の組合せごとに、どの上流の代理サーバ (DeleGate,Socks または各プロトコルの代理サーバ) に中継するか、あるいは直接目的のサーバに接続するかを選択する。同一目的サーバに対して複数の経路が与えられた時は、それ

らを順次試行することにより、不通の経路を迂回する。

データキャッシュ

一旦中継したデータをキャッシュに保持し、同一データに対する要求時に再利用する (HTTP,Gopher,FTP,NNTP)。

コネクションキャッシュ

一旦サーバに接続したらその接続を保持し、同一サーバに対する要求時に (複数のクライアント間で) 共有・再利用する (FTP および NNTP)。

プロトコル変換

クライアントのプロトコルとは異なるプロトコルのサーバと通信し、これを変換してクライアントに中継する (FTP,NNTP,Gopher,WAIS から HTTP へ)

マウント機能

複数のサーバが提供する情報を、選択、併合し、別の階層構造や情報の識別名に写像してクライアントに提供する (HTTP,FTP,NNTP,POP)

MIME コード変換

クライアント / サーバ間で交換されるインターネットメイルの MIME エンコード / デコードを行う (NNTP,POP,SMTP)

文字コード変換

クライアント / サーバ間で交換されるテキストデータに対して、3 種類の日本語文字コード (ISO-2022-JP, Shift-JIS, EUC-JP) の間の変換を行う。

外部フィルタ機能

利用者が用意したフィルタ (入力したデータを変換して出力するプログラム) を、 DeleGate とクライアント、あるいは DeleGate とサーバとの間に挿入し、通信データを加工する。

3.3 DeleGate の防火壁機能

DeleGate が応用レベルで通信を中継していることにより実現が可能になった、主に防火壁で利用される機能を紹介する。

3.3.1 サーバの運営を支援するための代理サーバ

– マウント機能 –

代理サーバは防火壁内部のクライアントが外部のサーバにアクセスするために使われることが多いが、逆に防火壁内部のサーバを外部のクライアントに提供するためにも利用できる。

安全水準を高く保たなければならない防火壁にサーバを置いて、多人数で維持作業を行うことは、安全性を低下させてしまい、維持作業にも不便を伴う。この問題を解消するために、安全な防火壁内部に実際のサーバを置いて自由に保守を行い、防火壁上では代理サーバによる中継だけを行う、という方法を取ることができる。

DeleGate の「マウント機能」は、このような目的に利用できる機能である。これは、ファイルシステムのマウント機能の考え方を一般化したものであり、基本的に以下の 3 つの機能を提供する。

・複数サーバの併合 (merge): 複数のサーバ上にある情報や機能を、すべて一つのサーバ上にあるかのように見せてクライアントに中継する。

・別名 (alias) と隠蔽: 元サーバでの識別名と異なる識別名に写像してクライアントに提供し、元サーバに関する情報を隠蔽する。

・情報 / 機能の絞り込み (filter): 元サーバが提供する情報や機能の一部分だけを中継する。

複数のネットワークインタフェース (ホスト名) を持つホストにおいて、 DeleGate は、どのインタフェースからクライアントが接続して来たかにより、マウントのしかたを選択することができる。これにより、単一の DeleGate サーバを複数の実サーバであるかのようにクライアントに見せることもできる。

DeleGate ではこれまで、 HTTP,NNTP,FTP および POP の各プロトコルに対して、マウント機能を実現している。 DeleGate がこれらの (POP 以外の) プロトコルに対してキャッシュ機能を持っていることにより、実際のサーバが停止していても代理サービスは継続できるという利点も得られる。

マウント機能は、クライアントからサーバへの要求、およびサーバからクライアントへの応答の中に含まれる、情報へのポインタを書き換えることで実現される。例えば HTTP においては情報へのポインタは URL という形で統一的に表現され、 HTTP プロトコルで構文が規定される要求 / 応答メッセージのヘッダや、 HTML 形式のボディの中に現れる。

例えば、実際には内部のサーバにある (1) のような情報を、クライアントに対しては防火壁上のサーバの (2) にあるように見せたいとする (* は任意長の文字列)。

(1) http://hostI/*
(2) http://hostX/I/*

これは、防火壁上でHTTPを中継するDeleGateにおいて、要求の中に現れる(1)のようなURLを(2)のように書き換え、逆に応答中に含まれる(2)のようなURLを(1)のように書き変えることで実現できる。このようにマウント機能は、2.3に述べた書き換えと基本的には同じ機構を利用しており、その自然な拡張として考案されたものである。

同様に、要求／応答メッセージに現れる情報のポインタは、FTPにおいてはディレクトリ名、NNTPにおいてはニュースグループ名であり、それらを双方向の中継において書き換える。

ただし残念ながら、プロトコル上規定されている部分以外にポインタが出現した場合には、この方式では対応できない。

3.3.2 一時的代理サーバの自動生成

– HTTPからのXクライアントの起動と中継 –

HTTPからCGI経由でHTTPサーバ上にX Windowのクライアントを起動し、利用者側(HTTPクライアント側)にあるXサーバに接続して画像情報を提供するサービスがある。ところがXプロトコルが防火壁を越えられない環境ではこれを実現できない。

ここで、HTTPクライアント側にあるXサーバの情報は、HTTPのPOSTメソッドによる要求の中に，FORM形式で送られるのが普通である。例えば以下の(1)のような形式である。

(1) display=hostname:display#&...

DeleGateはこのような要求を中継する際、FORMの内容を見て、hostname:display#で指定されるXサーバへの中継が必要だと判断し、そのために専用の一時的な時限代理サーバを自動生成する。そして、生成したX代理サーバのアドレスを用いて、クライアントからの要求を以下の(2)のように書き換え、HTTPサーバに中継する(図4)。

(2) display=proxy-hostname:proxy-display#& ...

この要求によりHTTPサーバ上に起動されたXクライアントは、要求中に示された代理Xサーバに接続し、その代理サーバが実際のXサーバに中継することになる。

これは、一種のマウント機能であるが、HTTP や HTML で規格化されていない情報の書き換えの例である。このような (X サーバのような) 資源へのポインタも、最終的には URL として標準化されることが望まれるが、防火壁内のサーバを公開する目的に DeleGate を使用する場合には、実サーバと DeleGate の間での合意があれば、標準化されていなくても利用することができる。

3.3.3 パスワードの代理暗号化

– POP ログインにおける USER・PASS と APOP の変換 –

盗聴の恐れのある公開のインターネット上に、生のパスワードを流すのは言うまでもなく危険である。例えば、出張先などから遠隔でメールを読み書きするために、POP プロトコルを使うような場合である。POP では、このような状況に対処するために、パスワードを暗号化して転送するコマンド (APOP コマンド) を用意している。ところが、最小実装規約に含まれていない APOP コマンドは実装されていない場合が少くない。

そこで、POP クライアントと隣接する DeleGate との間では、実装が保証されている認証コマンド (USER コマンドと PASS コマンド) でログインし、DeleGate と遠隔の POP サーバとの間でこれを APOP コマンドに変換してログインを行う機能を実現している (図 4)。

3.3.4 代理データ変換

– NNTP 中継における文字コード変換 –

ネットニュース上で交換する日本語テキストの文字コードは、ISO-2022 のエスケープコードを用いて ASCII と JIS の文字セットを混合してエンコードしたもの (ISO-2022-JP)、と規定されている。しかし、特に最近パーソナルコンピュータによるインターネットへの参加が増えるに伴い、投稿者側のツール (ニュースリーダ) での内部コードが流出し、読み手のツールに障害を引き起こす事故が増えている。

DeleGate では、NNTP を中継する際に、ネットニュースに送出される日本語テキストの文字コードを強制的に ISO-2022-JP に変換することにより、このような内部コードの流出事故を防ぐことができる。

3.3.5 TCP/IP の接続方向の変換

– FTP データ接続における PORT と PASV の変換 –

防火壁を越えてTCP/IPの接続を許す方向を一方向に制限している場合がある。一般に「内部から外部へ」の接続だけを許可することは、TCP/IP上の応用サービスに対して外部からの侵入を防ぎ、内部の利用者の利便性を損なわない簡潔な方法である。ところが、このような一方通行の接続によって、ファイル転送プロトコル(FTP)が利用できない状況が生じた。

FTPでは、コマンドのやりとりのための制御用の接続の他に、データ転送のために専用の接続を作成して使用する。制御用の接続は当然クライアントからサーバに向けて接続されるが、データ転送用の接続はサーバ側からもクライアント側からも張ることができる。伝統的には、サーバからクライアントへの接続(PORTコマンドによる)が使われて来た。これでは、上記のような一方通行のTCP/IP接続環境では、外部のFTPサーバから内部のFTPクライアントに向けて、データ転送用の接続を行えない。

この場合の問題は、クライアントからサーバに向けて接続を行うPASVコマンドを用いることで解決することができる。ところが、FTPの最小実装規約に含まれているPORTコマンドと異なり、PASVコマンドは(クライアントとサーバともに)実装されているとは限らず、使用できない場合がある。

また、これとは逆の状況として、防火壁の内部にあるサーバを公開する場合に、外部から内部への接続は制御用の標準ポートにしか接続を許さないという場合がある。この場合にはPORTコマンドで内部のサーバから外部のクライアントへデータ転送用の接続を行わなければならない。

このように、クライアントの希望する(可能な)接続方法が防火壁を越えられない場合、あるいは目的サーバが希望する(可能な)接続方法とクライアントのそれが一致しない場合、データ転送用の接続を確立することができないという問題が生じる。そこでDeleGateはこれらの問題を解決するために、クライアントがPORT,PASVのどちらでDeleGateに接続を要求して来たかと、DeleGate自身がサーバに対してPORT,PASVのどちらで接続を要求するかを、独立に処理している。サーバに対してはまずPASVを試行し、失敗した場合にはそれ移行PORTによる接続を行

うようにすることで、接続方向の不整合を自動的に吸収している。

4　おわりに

応用プロトコルレベルで通信を中継する技術そのものは新奇なものでは無く、DeleGateの基礎を構成しているのは古典的な技術の集積である。また、その応用としての防火壁の実現は、現時点で社会的な要請の高い重要な応用ではあるが、それはこの技術の応用分野の一部に過ぎない。

応用プロトコルレベルでの中継技術の発展的応用としては、今後も様々なものが現れて来ると考えられるが、DeleGateが現時点で検討している課題としては以下のようなものがある。

・遠隔地点間でのTCP/IP通信の効率化: データ転送に先立って接続の確立のための通信を必要とするTCP/IPでは、遠隔地点間での通信の際に接続のためのオーバヘッドが大きい[6]。接続済みのTCP/IP接続を広域的に共有できる資源と考えて、仮想的なネットワークとして管理し、再利用することができないだろうか？

・広域分散協調キャッシュ: 広域に分散した複数のキャッシュ付き代理サーバ間で、キャッシュを階層化・ネットワーク化し、協調的に動作させる。現在世界的に、Internet Cache Protocol(ICP) [5]を用いたキャッシュサーバ協調プロジェクトが進行中であり、DeleGateでも対応を検討している。広域に分散したサーバ間での制御情報の通信のコスト、能動的な更新を考慮してキャッシュ制御プロトコルを設計することが必要である。

・ATMの効率的な利用: ATMの上でIPを運ぶ「IP over ATM」では、輻輳時の通信効率の低下が問題となっている。IP over ATM 上での応用プロトコルの伝送を中継サーバとしてのDeleGateで制御する、あるいはIPを介さず、直接的にATM上でDeleGate-DeleGate間を結んで応用プロトコルを中継することで、改善を試みる。

・動画像形式の変換・蓄積・中継: CU-SeeMeのような実時間・揮発性の動画像転送プロトコルを、MPEGのような動画像形式に変換してHTTPに中継したり、

蓄積して検索させられるようにする。

・特殊分野のプロトコルの変換中継: 医療画像情報の形式、転送プロトコル、応用を包含する規約として、「DICOM」の標準化が進行している。その普及を促進するために、これをHTTPのような広く普及したプロトコルに変換して中継する。

・メール/ニュースの自動暗号化/自動署名: メールを暗号化して転送したいという要求は高く、多種多様なメールツールがそれぞれ対応するのを待っていられない。SMTPによるメールの送出、POPによるメールの受信のそれぞれにおいて、自動的に暗号化/復号化(おそらくPGP化)を代理サーバで行う。

・外部フィルタの有効利用: これまでDeleGateの外部フィルタ機能を利用して、英日翻訳のためのフィルタ[9]や、テキストフォーマット、HTML形式の変換、画像形式変換、衛星回線や電話回線を利用した応用プロトコルの中継、などの応用が成されている。より広い応用に対応するために、インタフェース形式の拡張と標準化が必要である。

・Unicode/日本語コード変換: 従来、日本語文字コードとして3種類のものが混在して使われて来たが、これに加えてUnicodeが普及しつつある。Unicodeベースのクライアントやサーバと、従来の文字コードの間の変換中継を行うことで、情報の流通を支援する。

・多国語文字表示: 多国語の文字コードの混在するテキストをHTMLのインラインイメージの列に変換してWWWブラウザに表示させる機能、中国語の各種文字コードによるテキストを日本語の漢字コードにマップして表示する機能が、外部の協力者により提供され、DeleGateに組み込まれている。これらの機能は外部フィルタとして独立させたほうが有効利用でき、発展させられるだろう。

一方、応用プロトコルの中継システムの基礎技術としては、以下のようなものが重要であると考えている。

・プロトコルの形式記述と検証(プロセス代数)

・文字列のパターンマッチングと書き換え

・データ暗号化

・データ圧縮

・(中継システムのための)生成系とインタプリタ

現在DeleGateは、約6万行の手作りのCプログラ

ムで力任せに実現されているが、最終的には各プロトコルのコンパクトで形式的な記述から、利用者に必要十分な中継サーバをエレガントに生成するシステムに結びつけられればと考えている。

これまでDeleGateの開発では、現実に次々に直面する様々な要求や可能性を目前にして、ともかくもそれらを解決し実現して来た。「実現したら確実にありがたい」ことを「あたりまえの技術」で実現し続けて来たと言える。その結果、多数の利用者からの活発なフィードバックが得られ、それが改良を促進し、さらなる応用可能性を生んで来たと思う。[†1]

インターネットは現在もWWWを中心として飛躍的な発展を続けている。インターネット社会全体としては大枠を公開標準で抑えながら、その中でボトムアップ的な競争的発展が効果的に行われている。このような発展の過程では、仕様や実装の食い違いがあちらこちらに生じる。DeleGateのような中継サーバは、それらの隙間を埋め、歪みを緩衝することにより、全体として「疾走しながら改良し続け発展する」過程を支援する役割を担えると期待する。

[†1] DeleGateは*ftp://ftp.etl.go.jp/pub/DeleGate/*で配布されている。DeleGateに関する情報は*http://www.etl.go.jp/People/ysato/*を参照。

参考文献

[1] D.B.Chapman and E.D.Zwicky: Building Internet Firewalls, O'Reilly & Associates, Inc. (1995).

[2] W.R.Cheswick and S.M.Bellovin: Firewalls and Internet Security, AT & T Bell Laboratories, Inc. (1994).

[3] K.Siyan and C.Hare: Internet Firewalls and Network Security, New Riders Publishing (1995).

[4] A.Luotonen, H.F.Nielsen and T.Berners-Lee: Status of the W3C httpd, *http://www.w3.org/pub/WWW/Daemon/*

[5] Internet Cache Protocol Specification 1.4, *http://excalibur.usc.edu/icpdoc/icp.html*

[6] S.E.Spero: Analysis of HTTP Performance problems, *http://sunsite.unc.edu/mdma-release/http-prob.html*

[7] 佐藤: 多目的プロトコル中継システム DeleGate の開発, 電子技術総合研究所彙報 Vol.59, No.6 (1995) pp.381-396.

[8] 佐藤: 多目的プロキシサーバ DeleGate の機能詳細, インターフェース Vol.21, No.9 (1995) pp.130-146.

[9] 鵜川, 秋元, 藤田, 佐藤: DeleGate Proxy を使った Web 情報の英和逐語訳サーバ -EtoJ_Proxy-, 日本 WWW カンファレンス 95 神戸 (1994).

[10] NIC (Network Information Center): RFC (Request For Comments). *ftp://ds.internic.net/rfc/*

表 1: DeleGate が中継するプロトコル

(a) 内容を解釈して中継するプロトコル

名称	機能概要	規格名
HTTP	ハイパーテキスト転送	RFC1945
FTP	ファイル転送	RFC959
WAIS	広域情報検索	RFC1625
Gopher	広域情報提供	RFC1436
POP	メールボックス管理	RFC1460
SMTP	電子メール配送	RFC821
NNTP	電子ニュース配送	RFC977
X	Ｘウィンドウシステム	RFC1013
Telnet	遠隔ログイン	RFC854
Socks	回路レベル代理サーバ	RFC1928
CU-SeeMe	簡易遠隔会議	

(b) 解釈・使用するデータ形式とプロトコル

HTML	ハイパーテキスト記述言語	RFC1866
URL	一元的情報資源識別名	RFC1738
MIME	RFC822 の多目的拡張形式	RFC1521
CGI	共通ゲートウェイ仕様	
Ident	利用者認証プロトコル	RFC1413

図 1: クライアント / 代理サーバ / サーバ

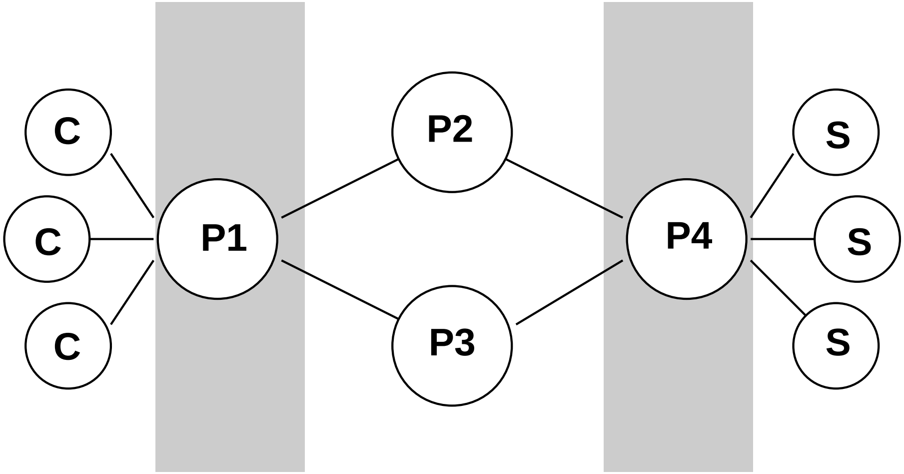

図 2: POP におけるパスワードの代理暗号化

図 3: HTTP からの自動的 X 代理サーバ生成