マルチメディア通信と分散処理　97－38
コンピュータセキュリティ　8－38
（2000．3．22）

複数ネットワークへの同時アクセスのためのアドレス解決方式

田中　俊介　松田　栄之　箱守　聰

株式会社 ＮＴＴデータ　情報科学研究所

〒104-0033 東京都中央区新川 1-21-2 茅場町タワー
Tel: 03(3523)8080　Fax: 03(3523)8090
Email： shun@rd.nttdata.co.jp

あらまし:
企業などのネットワークでは，FireWall を介してインターネットに接続し，メンバーが移動端末などで組織外からアクセスするためのリモートアクセスサーバを設けている場合が多い．その場合，ネットワークの内側のユーザとネットワークの外側のユーザには異なるサービスを提供している．しかしながら，従来の端末のネットワーク機能では，1 つのネットワークに接続することを前提に作られているため，ユーザが同時に複数のネットワークへ接続することが難しい．本研究では，ユーザが複数のネットワークに同時に接続し，複数ネットワークのサービスを同時利用可能な環境の実現を目指す．本報告では，問題点の抽出と解決アプローチの提案を行う．また，問題解決の具体例として，複数ネットワーク内の全ホストに対するアドレス解決機能について述べる．

A Design of Host Address Resolution Method for Multiple Networks

Shunsuke Tanaka　Shigeyuki Matsuda　Satoshi Hakomori

NTT DATA CORPORATION
Laboratory for Information Technology

Kayabacho Tower,1-21-2,Shinkawa,Chuou-ku,Tokyo,104-0033,Japan
Tel: 03(3523)8080　Fax: 03(3523)8090
Email: shun@rd.nttdata.co.jp

Abstract:
Recently, enterprise private networks have come to join the Internet through firewalls. Also, users have come to take along mobile terminals and access to their network from outside networks through the Internet by Virtual Private Network(VPN). In such case, users may want to access the resources not only on their private network, but also on the network which they are temporary connecting. There are some difficulties to realize such environments, since conventional network services support only one network configuration.
In this paper, we discuss the problems and requirements for network service functions to support multiple network configurations. We also describe the design and implementation of host address resolution method.

1. はじめに

近年のインターネットの普及は，学術団体だけではなく，企業などの商業団体が利用しはじめるなど，利用者の幅が広がったことが影響している．企業が自組織のネットワーク(イントラネット)をインターネットに接続する場合には，セキュリティの面から FireWall を導入し，インターネットから内部ネットワークへアクセスできないようにする場合が多い[1,2].

最近，企業間の合併，提携が相次いでいる．2 つの企業が 1 つになる合併も多いが，間接業務のアウトソーシングや競合関係にある企業がある事業部門に限って提携するなど，弱い繋がりの連携も多い．提携した企業間では，お互いに相手のネットワーク，サービスを利用可能にするためにネットワーク構成を変更する必要がある．ネットワークの変更方法としては以下の 3 つがある．

(1) ネットワーク同士を VPN(拠点間 VPN)で結合する方法 [3,4]

この方式では，全ユーザが 2 つのネットワーク全域にアクセスできるフラットなネットワークが作られる．ユーザの端末には設定の変更やモジュールの追加は不要であるが，2 つのネットワーク間でネットワークの基盤的役割を果たすサーバ(認証サーバ，DNS サーバなど)を連携させる必要があり，接続先が増える度に管理者の作業が多くなる．

企業の合併などには適するが，弱い繋がりの企業間連携などには不向きである．

(2) ISP(Internet Service Provider)などが提供するローミング[15]を利用して VPN を構築する方法

ネットワーク同士は同一の ISP に VPN で接続し，外出先にいるユーザは，PHS や携帯電話を利用して最寄りのアクセスポイントに接続し，ISP 経由でネットワークに接続する方式である．

この方式では，任意のユーザが任意のネットワークにアクセスする環境を ISP などが用意するため，ユーザの端末での設定変更，モジュール追加は不要であり，管理者もサーバの設定を変更する必要はない．

しかし，LAN が敷設された環境でも，帯域幅が狭く，コストが高い公衆網へアクセスしなければならないという問題点がある．

(3) リモートアクセスを利用する方式

図 1 のように，各端末が，所在地ネットワークへの LAN 接続と，遠隔地のネットワークへのリモートアクセスによって，複数のネットワークに接続する方式である．

この方式では，各ネットワークは独立しているため，提携先が増えた場合にも管理者の作業負担は小さい．そのため，弱い繋がりの企業間連携など，特定少数のユーザだけが両方のネットワークにアクセスできる環境を構築したい場合に向いている．

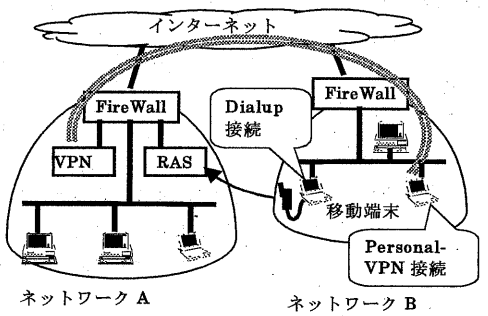

図 1　2 つのネットワークへの接続

しかし，現在の移動端末では，ユーザが複数のネットワークが提供している複数のサービスを同時に利用することができない(いずれか 1 つのネットワークのサービスしか利用できない)という問題点がある．現状では，ネットワーク A のサービスを利用していた状態から，アプリケーションを終了し端末の設定を変更して端末を再起動することで，ネットワーク B のサービスを利用している場合が多い．端末の再起動には数分かかるため，ユーザの作業効率が低下している．

本研究は，図 1 のようなネットワーク構成の場合に，複数のネットワークへ同時に接続し，複数ネットワークの複数サービスを同時に利用可能な環境の実現を目的とする．2 章で，従来技術とその問題点について記述する．3 章では，目的とする環境を実現するための課題について記述する．4 章では，具体的な課題の 1 つであるアドレス解決に関して，課題を解決する新しいアドレス解決方式について述べる．

2. 従来技術と問題点

アプリケーションから見たネットワークの機能は，データ転送を実現する TCP/IP 層と，その上位の通信サービス層に大別できる．ユーザが複数ネットワークに接続し，それらのリソースを利用するためには，各々の層において複数ネットワークを意識した処理が必要となる．本章では，各層における従来技術とその問題点について述べる．

2.1. TCP/IP 層

この層では，TCP/IP に関する設定を自動的に行ない，各ネットワーク内に接続された全てのホストと TCP/IP での通信が行なえる状態を動的に実現することが望まれる．

複数ネットワークに接続する場合には，図 1 のように以下の 2 つの接続形態がある．

1) LAN 接続と Dialup 接続の併用

Dialup 接続では，LAN インタフェースとモデムインタフェースの 2 つのネットワークインタフェースを利用する．2 つのインタフェースにはそれぞれ別の IP アドレスが割り当てられる．このとき，2 つのインタフェースには，DHCP[16]などを利用することにより，IP アドレスを自動的に設定することができる．他のホストと通信するときには，ネットワーク毎にインタフェースとソース IP アドレスを使い分けるため，どちらのネットワークに接続しているホストとも通信が行なえる．

2) Personal-VPN 接続

Personal-VPN 接続とは，Mobile IP[5,6]のリバーストンネリング[7]もしくはそれに相当する技術を利用して別のネットワークと VPN 接続する技術である(通常の Mobile IP では不十分である[8])．ネットワーク A およびネットワーク B の FireWall には Personal-VPN の IP パケットを通す設定を行なう[2]．Personal-VPN では端末内に LAN インタフェースと仮想ネットワークインタフェースがある．LAN インタフェースには DHCP 等によってネットワーク B のアドレスが自動的に割り当てられる．一方，仮想ネットワークインタフェースにはネットワーク A の IP アドレスを静的に割り当てる．ネットワーク A 内のホストとの通信は全て仮想ネットワークインタフェースを利用して行なう．よって，どちらのネットワークに接続しているホストとも TCP/IP での通信は行なえる．

以上のように,TCP/IP 層においては,複数ネットワークに接続した場合にも,全てのネットワークのホストと通信を行なうことが可能となっている．

2. 2. 通信サービス層

この層は､ユーザ名から暗号化パスワードを得るユーザ認証，ホスト名から IP アドレスを得るアドレス解決(リゾルバ)など，何らかの「名前」に対して応答となる「値」を返す API 群からなり，ネームサービスと呼ばれる[9]．ネームサービスの API は通常 OS に付随したライブラリとして実装されており，OS によって提供するＡＰＩの種類が異なっている．代表例として，UNIX 系 OS および Windows 系 OS が提供している API を表 1 に示す[10,11]．

同一の OS においても，同一の API を提供する複数の異なる実装がある．例えば，UNIX 系 OS のリゾルバは，hosts ファイル，DNS(BIND)[12]，NIS[13]，ディレクトリサービス[9]の 4 種類がある．また，nsswitch という複数の実装を使い分ける技術もある[10]．

表1 ネームサービス一覧

ネームサービス		OS	利用 AP	実装形態
名前	値			
ユーザ名	ユーザ ID	UNIX 系，Windows 系	多	1),2)
グループ名	グループ ID			
ユーザ名	パスワード			
ホスト名	IP アドレス			
プリンタ名	ホスト名			1)
サービス名	ポート番号	UNIX 系	並	1),2)
プロトコル名	プロトコル ID	UNIX 系	少	1),2)
エイリアス	メールアドレス			
ネットワークアドレス	ネットワークマスク			
RPC 名	RPC 番号			
デフォルトメッセージサーバ	ホスト名	Windows 系	少	2)
サービス名	ハンドラ			

これらネームサービスの実現形態は，以下の 2 つに大別できる．

形態 1) 名前と値のテーブルをホスト上のファイルとして保持し，ファイルから値を得る方式．

形態 2) ネットワーク上に名前と対応する値を保持する名前解決サーバがあり，ホストは名前解決サーバのアドレスを保持しており，ホストが名前解決サーバに問い合わせを行うことで値を得る方式．

複数ネットワークに接続する場合，形態 1 では全ネットワークの全ソースに対して，名前と値の組を記述することで，全リソースを同時利用可能になるが，ユーザの負担が大きい．形態 2 の場合，複数ネットワークに接続する場合には，各ネットワーク上の名前解決サーバにアクセスできる必要がある．しかし，現在実装されれいるいずれの API においても，名前解決サーバを 1 つしか指定することができない．このため，接続するネットワーク毎に設定を切替える機能を有する必要がある．

以上のように，通信サービス層においては，複数ネットワークへの接続をサポートしているとは言い難い．このような環境でユーザへの負担を少なくするためには，この層における機能の充実が望まれる．

3. 複数ネットワークへの同時アクセス実現に向けた課題

複数ネットワークへのアクセスにおけるネームサービスの課題は，①複数ネットワークアクセスのための

複数名前解決サーバの登録に関する課題、②複数ネットワークにアクセスすることによる名前解決の課題に分けることができる．

(A) 複数の名前解決サーバの登録に関する課題

1)複数名前解決サーバの登録

例えば図１において，ネットワークＡの名前解決サーバはネットワークＡの情報しか持たず，ネットワークＢの名前解決サーバはネットワークＢの情報しか持っていない．名前解決サーバを１つしか登録できないと，他のネットワークの名前解決ができないため、アクセスするネットワークに対する名前解決サーバを登録する必要がある．

2)名前解決サーバの登録順変更

同じ名前のエントリが複数の名前解決サーバに登録されている場合，問い合わせる名前解決サーバによって異なる応答を受け取ることになる．また，同一名称が存在すると参照する名前解決サーバの順番によって期待していない応答が帰ってくることになる．そこで，問い合わせを発行する時点で，何らかの条件に基づいて，問い合わせるサーバを切替える仕組みが必要になる．

3)名前解決サーバ自動登録との連携

通常，名前解決サーバを自動登録する DHCP クライアントでは１つの名前解決サーバしか登録できないため，名前解決サーバ情報(サーバの IP アドレスなど)を上書きする．前述の通り，名前解決サーバを複数登録する必要があることから，名前解決サーバ情報の更新は上書きでなく追加の形式にする必要がある．名前解決サーバ登録順の動的変更を実現するためには，設定ファイルが複数になる場合があるため，全ての設定ファイルに名前解決サーバ情報を追加する必要がある．

(B) 名前解決の課題

1)他ホストへのアクセスした場合のユーザ認証

図 1 の環境を例として考える．ユーザはネットワーク A のアカウントとネットワーク B のアカウントという 2 つのアカウントを持つ．ネットワーク A 内のホストにアクセスする場合，ネットワーク A のアカウントを提示してアクセスする必要がある．ネットワーク B のアカウントを提示すると，アクセスを拒否される．

2)自端末自体のユーザ ID，ユーザ認証

図１の環境では，ネットワークＡのユーザアカウント，ネットワーク B のユーザアカウント，自端末端末内のユーザアカウントという 3 つのユーザアカウントがある．3 つのアカウントではユーザ名は同じ場合もあるし，違う場合もある．アクセス対象とするホストにおけるユーザアカウントの整合性を取る必要がある．

3)他ホストからのアクセスに対するユーザの認証

図１で，ネットワークＡ内のホストが移動端末にアクセスしてくる場合を考える．ネットワークＡ,ネットワークＢには，それぞれユーザ名とユーザ ID のテーブルがあり，これらのユーザＩＤの整合性を取る必要がある．

4. アドレス解決機能の実現

ホスト名を IP アドレスに変換するアドレス解決機能であるリゾルバを対象に，複数ネットワーク上に配置されるホストのアドレスを解決する機能を実現した．本章ではその実現方式について述べる．

4.1. 現状のアドレス解決方式

２つのネットワークからなる環境において，従来のリゾルバを用いてホストのアドレス解決を行うときの例を，図２に示す．図２において，移動端末は xyz.or.jp ドメインのネットワークに接続しており，abc.co.jp ドメインへも Personal-VPN により接続されている．各ドメインには，ドメイン内のホスト名のアドレスを解決する内向け DNS サーバと，ドメイン外のアドレスを解決する外向け DNS サーバとが配置されている．ここで，移動端末には abc.co.jp ドメインの内向け DNS サーバがアドレス解決のためのサーバとして登録されているとする．

ここで，移動端末が www.in.abc.co.jp というホストのアドレス解決を行うときには，abc.co.jp ドメイン内の内向け DNS サーバが呼び出され，このサーバが対応する IP アド

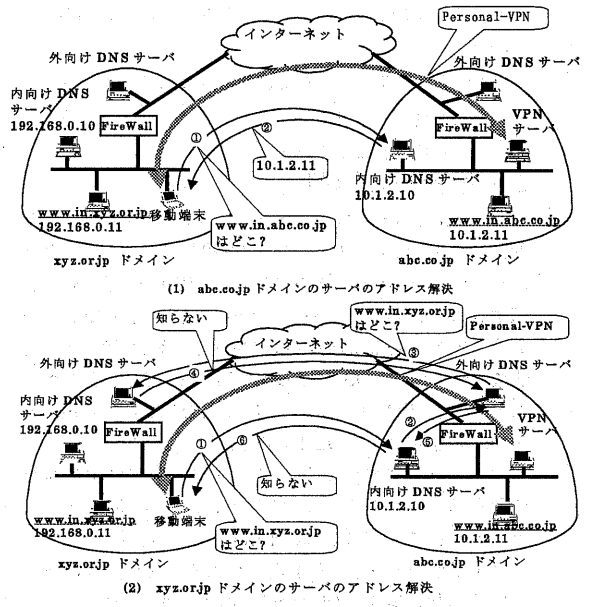

図 2　従来のアドレス解決方式での問題点

レスを応答する（図2(a)）．一方，www.in.xyz.or.jp というホストのアドレスを解決しようとすると，abc.co.jp ドメイン内の内向け DNS サーバも，xyz.or.jp ドメインの外向けサーバもこのホストに対応するアドレスを持たず，abc.co.jp ドメイン内の内向けサーバは外部からアクセスすることができないため，解決することができない．

4. 2. 実現方式の提案

リゾルバを複数ネットワークからなる環境に対応させるため，以下の機能を実現する．

(1) 複数の DNS サーバへの問い合わせ機能

DNS サーバを登録する設定ファイルにおいて，サーバの IP アドレスを記述するエントリを拡張し，問い合わせるホスト名に応じて複数の DNS サーバへアドレス解決を依頼することができるようにする．3章で述べた課題に対処するため，各エントリには，以下の3つの形式が記述できるようにする．

(a) 特定のホスト名と，そのホスト名のアドレス解決に用いる DNS サーバのアドレスの組
(b) 特定のドメイン名と，そのホスト名のアドレス解決に用いる DNS サーバのアドレスの組
(c) 任意のホスト名に対して，そのホスト名のアドレス解決に用いるサーバのアドレス

ここで，これらの3種のエントリは，(a)(b)(c)の順に優先度を持つ．例えば，www.in.abc.co.jp というホスト名の場合，リゾルバはまず(a)に該当するエントリの中からホスト名が一致するものを探す．次に(b)の該当する中から in.abc.co.jp に該当するエントリを探す．それでも一致するものがない場合には，(c)のエントリを対象として記述順に DNS サーバへアドレス解決を依頼する．こうすることによって，例えばユーザが通常利用しているファイルサーバのホスト名を(a)の形式で記述しておけば，移動先のネットワークに同じ名前のホストが存在していても，通常利用するホストのアドレスを得ることができる．

(2) ユーザおよびグループ毎に DNS サーバの設定ファイルを切り替える機能

3 章で述べたように，複数のネットワークを利用可能な環境では，実際に各アプリケーションプログラムがどのネットワーク上のリソースにアクセスすればよいかを決めることは難しい．ここでは，プログラム毎に DNS サーバへの問い合わせの順序を変更できるようにするために，アプリケーションプログラムが持つユーザ ID とグループ ID の組み合わせにより，参照する DNS サーバ問い合わせの設定ファイルを変更する機能を実現する．これにより，複数のネットワーク上に同じ名前を持つリソースがある場合にも，アプリケーションプログラムの走行環境を変えておくことで望みのリソースへアクセスすることができる．

4. 3. 実装

4.2 節で提案した機能を実装し，機能の確認を行った．

実験環境では，図3に示すように，4台の DNS サーバ，2台のファイアウォールからなるネットワークを構築し，そこにホスト名のアドレス解決を要求する端末を接続した．DNS サーバは BIND サーバプログラムを利用した．

端末は DOS/V 型計算機であり，OS は，Linux(Debian GNU Linux + Kernel 2.2.1)が走行する．提案するリゾルバは，この端末上の libc ライブラリ群の一つである libresolv を改造して実現した．このライブラリはダイナミックリンクライブラリとして作成しているため，端末上のプログラムは再コンパイルすることなしに提案する機能を利用することができる．

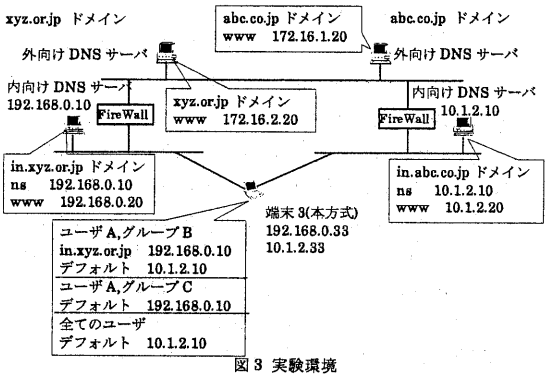

図 3 実験環境

5. おわりに

本報告では，ユーザが複数ネットワークに接続した場合に，全てのネットワークが提供するサービスを同時に利用できる環境を実現することを目的として，従来技術の問題点を調査し，ネットワーク機能に求められる課題を述べた．また，ネームサービスの1つであるアドレス解決に関して，解決指針にもとづいて，複数ネットワークに接続された全てのホストのアドレス解決が行なえる新しいリゾルバを提案，実装した．

今後の課題としては，3 章で述べた課題を解決していくことである．また，リゾルバの実装からは以下の課題が挙げられる．

1)実装したリゾルバの性能評価

実装したリゾルバと従来のリゾルバで，アドレス解決の処理時間を測定し比較を行なう．

2)サーバのリストを切替える実装方式の検討

今回は，検索対象の名前，ユーザ ID ・グループ ID を

用いた．これらの方式ではユーザの設定項目が多く不便である．より自動的かつ適切に切替えられる実装方法が望まれる．

[参考文献]

[1] 日本インターネット協会，インターネット白書'98, Jun.1998

[2] D. B. Chapman and E. D. Zwicky, Building Internet FireWall, O'REILLY, Jun. 1996

[3] 河合,インターネット VPN-急浮上するイントラネット広域化手法,日経コミュニケーション No.224,Jun.1996

[4] 井上 訳,エクストラネット-その設計と導入-,Dec. 1997

[5] W.Simposon, IP in IP Tunneling,RFC1853,Oct.1995

[6] C.Perkins, IP Mobility Support,RFC2002,Oct.1996

[7] G.Montenegro, Reverse Tunneling for Mobile IP, RFC2344,May 1998

[8] 石山 他, Mobile IP の現状と問題点に関する一考察,IPSJ-MBL-98-7-10, Dec. 1998

[9] L. Howard, An Approach for Using LDAP as a Network Information Service, RFC2307, Mar. 1998

[10] Free Software Foundation, The GNU C Library Reference Manual for Version 2.00 Beta, Oct. 1996

[11] 山根ドキュメンテーション 訳, Microsoft WindowsNT 4.0 Server ネットワーキングガイド,Mar.1997

[12] 浅羽,上水流 訳,DNS and BIND,ASCII, Aug.1995

[13] 下山,城谷, SUN システム管理, ASCII,Mar. 1991

[14] 山本, Mew マニュアル, May 1998

[15] B. Aboba, etc, Review of Roaming Implementations, RFC2194, Sep. 1997

[16] R. Droms, etc, Dynamic Host Configuration Protocol, RFC2131, Mar. 1997

[17] S. Alexander and R. Droms, DHCP Options and BOOTP Vendor Extensions, RFC2131, Mar. 1997