平成１４年度　研究開発成果報告書

「モバイル環境やセキュリティを考慮した名前解決方式と
その検証環境の研究開発」

目　　次

1　研究開発課題の背景

現在インターネットを利用するほぼすべてのアプリケーションはDNSによる名前解決を利用している。通信相手の名前を解決してIPアドレスを取得する名前解決システムはインターネット上で通信相手を特定するために必要不可欠な機構であり、インターネット上でアプリケーションを利用するためにはDNSによる名前解決システムがプラットフォームに依存しない形で安定して稼動している必要がある。

通常のOSではDNSによる名前解決を行うための共通のライブラリを提供している。またRFC2553ではこのライブラリの基本的なAPIを定めており、OSに依存しない汎化されたAPIを利用できるようになっている。これらのAPIは通常近傍にあるDNS cacheサーバと通信し、名前とIPアドレスの相互変換を行う形になっている。このライブラリ群は通常リゾルバライブラリと呼ばれる。またDNS cacheサーバとの通信はport 53を利用してUDPあるいはTCPで行われる。これらのプロトコルはRFC1035、RFC2671で定められている。

一方近年になり、特定のネットワーク上に固定的に接続された計算機環境だけではなく、異なるネットワーク間をノードが移動するモバイル環境も一般的になりつつある。特定のDNS cacheサーバを固定的に設定しておけば十分であった固定環境と違い、モバイル環境においては移動先ネットワーク上で利用可能なDNSサーバを探索する必要があったり、DNSが利用できない環境で近隣ノードの名前解決を行ったりする必要が生じる。

あるいはセキュリティやプライバシーを考慮したネットワーク環境において、ファイアウォールで分断されたいずれのネットワークにおいても適切な名前解決を行えるような機能も必要となることがある。

本研究開発においては、一般のネットワークアプリケーションが名前解決を行う際に標準的に使用するDNS cacheサーバとの通信機能を利用することで、様々なモバイル環境やセキュリティを考慮したネットワーク環境で利用可能な名前解決メカニズムをそれぞれの環境ごとに用意し、それらのメカニズムを統合的に処理することで環境に応じた適切な名前の解決を行うための汎用名前解決機構を試作開発して、各名前解決メカニズムの有用性を評価、検証することを目的とする。

2　研究開発分野の現状

2-1　DNS を利用できない場合における名前解決

DNS を利用できない場合において、近隣のノードの名前の解決に利用できる方法として現在 IETF で議論されているプロトコルとしては以下の 2 つがある。

- Linklocal Multicast Name Resolution (LLMNR)[1]
- IPv6 Node Information Queries[2]

LLMNR では、名前を解決したい対象となるノードで LLMNR を理解し、応答するためのプログラムが必要となる。

2-2　DNS サーバの自動発見

現在の DNS では、クライアントは通常近傍の DNS サーバを IP アドレスで指定することによって名前解決が行われる。この方法では、ユーザが何らかの形で近傍の DNS サーバを発見し、その IP アドレスを設定しなければならない。これはユーザが手設定することも可能であるが、現在の IPv4 における主流は DHCP クライアントが DHCP サーバから近傍の DNS サーバを通知してもらい、その値を設定するという方法である。この方法はユーザへの負担が少ないが、管理者は最終的に DNS サーバの設定だけでなく、DHCP サーバにその設定した DNS サーバのアドレスを静的に設定しなければならない。これは管理者の設定および保守コストを引き上げる結果となっている。加えて、IPv6 においては、各ノードがルータからの router advertisement message を受信することによって、自立的に自分のアドレスを決定し、インタフェースに割り当てる機能がある。この機能は Stateless Address Autoconfiguration と呼ばれ、ルータを設定するだけでそのルータに接続する IPv6 ノードが通信可能となることから、IPv6 の利用の容易さを高める特徴でもある。よって、アドレス情報など様々な管理オブジェクトを持つ DHCP というプロトコルは IPv6 では広く利用されないことが考えられ、DHCP によって DNS サーバのアドレスをクライアントに通知するという方法は好ましくないと思われる。

これを解決するために、Well-known なアドレスを DNS サーバに割り当て、そのアドレスに対して問い合わせを行うことにより、自動的な DNS サーバの発見を行う。また、Well-known なアドレスについては anycast を利用することも考慮する。

Well-known なアドレスとしては、Well known site local unicast addresses for DNS resolver[3]で定義されているアドレスを利用することができる。

---

[1] draft-ietf-dnsext-mdns-12.txt
[2] draft-ietf-ipngwg-icmp-name-lookups-09.txt
[3] draft-ietf-ipv6-dns-discovery-06.txt

2-3　ファイアウォールで分断されたネットワーク環境における選択的名前解決

DNS では、名前空間の構造は木構造となっており、インターネットにおいてはこの構成木は一つだけであって、いかなる DNS サーバに名前を問い合わせても必ず同じ答えが得られるようになっている。例えば www.toshiba.co.jp という名前を持つ IP アドレスの値を、日本にある DNS サーバとアメリカにある DNS サーバのどちらに問い合わせても同じ値が返される。そのため、クライアントは任意の DNS サーバに名前を問い合わせることができる。

近年は網の安全性を確保するためにしばしばファイアウォールが導入される。このような環境においては、セキュリティ向上のためファイアウォール内部のノードの IP アドレスを公開しない場合が多い。これはすなわち前述の「構成木は一つ」という前提が失われていることを意味する。すなわち、ファイアウォール内部の DNS サーバに問い合わせた場合と、ファイアウォール外部で問い合わせた場合、応答の結果が違う場合が起こりうる。これは応答の内容が異なるというだけではなく、どちらかの応答は害となる可能性が大きい。例えばファイアウォール内部でしか公開されていない名前の IP アドレスをファイアウォール外部の DNS サーバに問い合わせた場合、内部の DNS サーバに訊けば IP アドレスが得られるにも関わらず、そのような名前を持つノードは存在しないという応答が返され通信できないという状態に陥る。ファイアウォールは大企業だけの利用にとどまらず、SOHO、あるいは簡易なものであれば家庭網でも利用されることが考えられるので、この問題は深刻となる場合が大きい。

このため、移動するノードは、複数の DNS サーバに問い合わせることが可能になっている場合においては、ある問い合わせに対して適切な応答を選択しなければならない。

3　研究開発の全体計画

3-1　研究開発課題の概要

IPv6 システムを利用するために必須となる名前解決システムを実現するため、様々な名前解決プロトコルを組み合わせてもユーザが意識することなくそれらを利用可能な汎用名前解決エンジンの研究開発を行う。またこのエンジン上のモジュールとして、ユーザが様々な場所へ移動して IPv6 システムを利用可能な名前解決システムの研究開発、およびプライバシーやセキュリティを考慮した名前解決システムの研究開発を行う。さらに、これらモジュールを汎用名前解決エンジン上で選択的に動作させることにより、その有効性を確認する。
具体的には、以下の研究開発を行う。

（ア）汎用名前解決エンジン
　様々な名前解決メカニズムを統一的に扱い、既存の IP アプリケーションに対してそれらが解決した名前情報を透過的に利用可能な形で通知できるインタフェースを持つ汎用名前解決エンジンを、特定のプラットフォームに依存しない形で実装し、機能検証する。
（イ）モバイルサポートのための名前解決システム
　移動ノードが通信を行う場合に必要な名前解決への要件として、以下の項目について検討する。さらに、汎用名前解決エンジン上のモジュールとして実装し、機能検証する。
　（イ―１）DNS が利用できない環境での(近隣ノードの)名前解決方式
　（イ―２）移動場所に依存しない DNS サーバの自動発見方式
　（イ―３）障害発生時等を考慮した DNS サーバの適応的選択方式
（ウ）セキュリティやプライバシーを考慮した名前解決システム
　名前解決におけるセキュリティおよびプライバシーからの観点の課題への要件として、以下の項目について検討する。さらに、汎用名前解決エンジン上のモジュールとして実装し、機能検証する。
　（ウ―１）ファイアウォールで分断されたネットワーク環境における選択的名前解決方式
　（ウ―２）名前を一般に公開しない機器に対する名前解決方式

そして、（ア）で開発した汎用名前解決エンジン上で、（イ）および（ウ）で開発した名前解決モジュール群を統合して動作させ、その有効性を確認する。

3-2　研究開発目標

3-2-1　最終目標（平成１６年３月末）

(1) 複数の名前解決モジュールを組み込み、それらを選択的に利用可能な汎用名前解決エンジンの実現（3-1　（ア）に対応）
(2) DNS が利用できない環境でも近隣のノードの名前を解決可能な名前解決モジュールの作成（3-1　（イ―１）に対応）
(3) 近隣の DNS サーバを動的に発見可能な名前解決モジュールの作成（3-1　（イ―２）に対応）
(4) 近隣のDNSサーバ障害発生時に適応的にDNSサーバを選択することによって、効率的な名前解決を可能とするモジュールの作成（3-1（イ―３）に対応）
(5) ファイアウォール等で分断されたネットワーク環境において、適切な応答を選択可能な名前解決モジュールの作成（3-1　（ウ―１）に対応）
(6) 複数のローカルデータベースを利用した名前解決がグローバルなDNS システムと透過的に利用可能な名前解決モジュールの作成（3-1（ウ―２）に対応）
(7) 各名前解決モジュールを組み込んだ汎用名前解決エンジンの統合動作検証の完了

3-2-2　中間目標（平成１５年３月末）

(1) 名前解決モジュールを組み込み可能な名前解決エンジンの試作完了（3-1　（ア）に対応）
(2) DNS が利用できない環境でも近隣のノードの名前を解決可能な名前解決モジュールの作成（3-1　（イ―１）に対応）
(3) 近隣の DNS サーバを動的に発見可能な名前解決モジュールの作成（3-1　（イ―２）に対応）
(4) ファイアウォール環境等で名前空間の構造が異なる場合において、複数のサーバから異なる応答を受けた場合においても適切な応答を選択可能な名前解決モジュールの作成（3-1　（ウ―１）に対応）
(5) 個別の名前解決モジュールと汎用名前解決エンジンの組み合わせによる単体動作検証の完了

3-3　研究開発の年度別計画

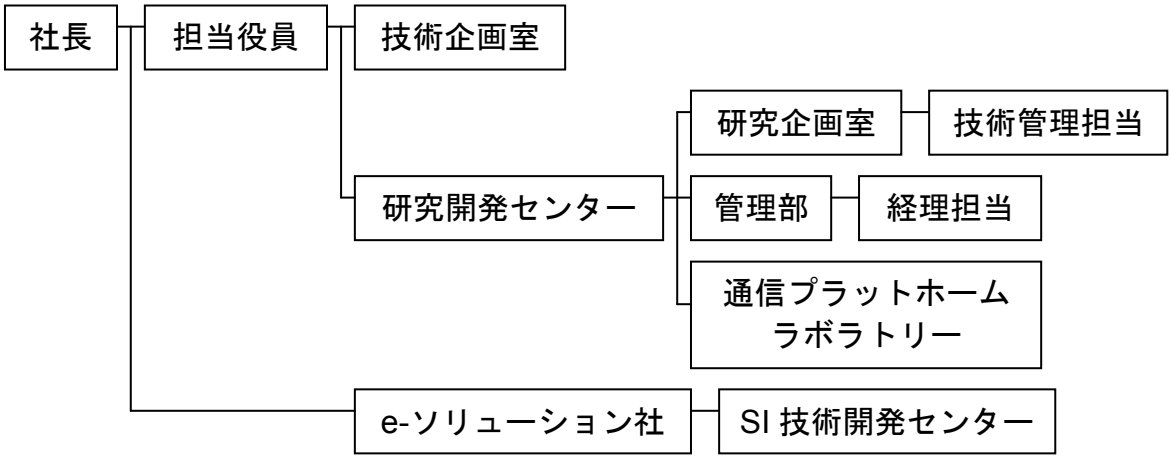
（金額は非公表）

研究開発項目	１３年度	１４年度	１５年度
【研究開発課題名】 モバイル環境セキュリティを考慮した名前解決方式とその検証環境の研究開発 【サブテーマ】 （ア）汎用名前解決エンジン	調査・仕様検討 →	機能試作・評価 →	機能試作・評価・統合動作検証 →

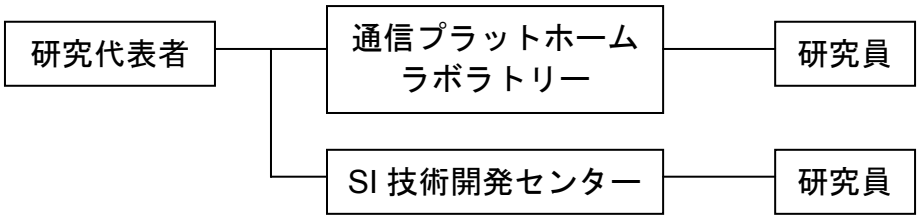
（イ）モバイルサポートのための名前解決システム （ウ）セキュリティやプライバシーを考慮した名前解決システム	→	→	→
間接経費			
合　　計			

3-4　研究開発体制

（1）　研究開発管理体制



（2）　研究開発実施体制



4　研究開発の概要（平成１４年度まで）

4-1　研究開発実施計画

4-1-1　研究開発の計画内容

IPv6 システムを利用するために必須となる名前解決システムを実現するため、さまざまな名前解決プロトコルを組み合わせてもユーザが意識すること無くそれらを利用可能な汎用名前解決エンジンの研究開発を行う。またこのエンジン上のモジュールとして、ユーザが様々な場所へ移動して IPv6 システムを利用可能な名前解決システムの研究開発、及び、プライバシーやセキュリティを考慮した名前解決システムの研究開発を行う。さらに、これらモジュールを汎用名前解決エンジン上で選択的に動作させることにより、その有効性を確認する。平成１３年度は、既存の名前解決の機能と関連標準化（IETF）の動向を調査し、“モバイルサポートのための名前解決システム”と“セキュリティやプライバシーを考慮した名前解決システム”の要求仕様をまとめる。また、現在使用されている名前解決システムについての調査を行ない、前述の要求仕様を実現する上での課題を明らかにし、“汎用名前解決エンジン”への要求仕様をまとめる。

平成１４年度は、“モバイルサポートのための名前解決システム”と“セキュリティやプライバシーを考慮した名前解決システム”のために必要な共通基盤要素技術の試作、および様々なモバイルサポート・セキュリティ要件に対応するための名前解決モジュールの試作を行ない、両者を組み合わせた単体動作検証を完了する。

具体的には、以下の５項目の研究開発を行う。

(1)　様々な名前解決システムのための共通基盤要素技術である、名前解決モジュールを組み込み可能な汎用名前解決エンジンの試作完了
(2)　DNS が利用できない環境でも近隣のノードの名前を解決可能な名前解決モジュールの試作完了
(3)　近隣の DNS サーバを動的に発見可能な名前解決モジュールの試作完了
(4)　ファイアウォール環境等で名前空間の構造が異なる場合において、複数のサーバから異なる応答を受けた場合においても適切な応答を選択可能な名前解決モジュールの試作完了
(5)　個別の名前解決モジュールと汎用名前解決エンジンの組み合わせによる単体動作検証の完了

4-1-2　研究開発課題実施計画

（金額は非公表）

研究開発項目	第１四半期	第２四半期	第３四半期	第４四半期	計
【研究開発課題名】モバイル環境セキュリティを考慮し					

た名前解決方式とその検証環境の研究開発 【サブテーマ】 （ア）汎用名前解決エンジン （イ）モバイルサポートのための名前解決システム （ウ）セキュリティやプライバシーを考慮した名前解決システム	基本設計	コーディング 基本設計 基本設計	単体検査 コーディング コーディング	総合試験 総合試験	
間接経費					
合　　計					

4-2　研究開発の実施内容

本研究開発では､次にあげる各ソフトウェアの試作および単体動作検証を行うものとする。

□　試作：
- 名前解決モジュールを組み込み可能な汎用名前解決エンジン
- DNS が利用できない環境でも近隣のノードの名前を解決可能な名前解決モジュール
- 近隣の DNS サーバを動的に発見可能な名前解決モジュール
- 複数のサーバから異なる応答を受けた場合に適切な応答を選択可能な名前解決モジュール

□　単体動作検証：
- 名前解決モジュールと汎用名前解決エンジンの組み合わせ

5　研究開発実施状況（平成１４年度）

本研究開発では、次にあげる各ソフトウェアの試作および単体動作検証を行った。

□　試作：
- 名前解決モジュールを組み込み可能な汎用名前解決エンジン
- DNS が利用できない環境でも近隣のノードの名前を解決可能な名前解決モジュール
- 近隣の DNS サーバを動的に発見可能な名前解決モジュール
- 複数のサーバから異なる応答を受けた場合に適切な応答を選択可能な名前解決モジュール

□　単体動作検証：
- 名前解決モジュールと汎用名前解決エンジンの組み合わせ

5-1　ソフトウェアの試作

様々な名前解決メカニズムを統一的に扱い、既存の IP アプリケーションに対してそれらが解決した名前情報を透過的に利用可能な形で通知できるインタフェースを持つ汎用名前解決エンジン（5-1-1　）を、特定のプラットフォームに依存しない形で実装した。

移動ノードが通信を行う場合に必要な名前解決の機能として、DNS が利用できない環境での名前解決を行う機能（5-1-2　）、および移動場所に依存しない DNS サーバの自動発見を行う機能（5-1-3　）を、それぞれ前述の汎用名前解決エンジン上のモジュールとして実装した。

名前解決におけるセキュリティおよびプライバシーの保護を行うために必要な名前解決の機能として、ファイアウォールで分断されたネットワーク環境における選択的名前解決を行う機能（5-1-4　）を、前述の汎用名前解決エンジン上のモジュールとして実装した。

5-1-1　名前解決モジュールを組み込み可能な汎用名前解決エンジン

内容　　リゾルバライブラリからの要求を受け､各名前解決モジュールに要求を伝え､その複数の結果から適切に応答を選び出しリゾルバライブラリに返す汎用名前解決エンジンを実装する。

状況　　汎用名前解決エンジンの試作開発を完了した。

成果　　以下の 5 つのサブユニットからなる汎用名前解決エンジンを実装した。

- Resolver-Engine Interface
  リゾルバライブラリからの要求パケットを受け付け、内部形式に展開する。また、モジュールからの応答をリゾルバライブラリに伝える。

- Policy Controller
  要求された名前に対して複数の名前解決モジュールに問い合わせた場合に得られる複数個の応答から、どのような応答を優先するかを管理する。

- Query Manager
  非同期に行われる複数の名前解決に関し、それぞれの状態を管理する。

- Module Manager
  現在どのような名前解決モジュールが接続されているかを管理する。

- Module Interface
  汎用名前解決エンジンと各名前解決モジュールとのインタフェースを抽象化する。

詳細　　汎用名前解決システムの概要を図 1 に示す。

ある名前解決を行うにあたり､その応答を引き出す概念上のエンティティをネームソースと呼ぶ｡ネームソースは名前解決を行う各プロトコルに対して定義される。例えばある一つの DNS cache server は DNS 上の一つのネームソースであり､ある一つのリンクに対してある特定のプロトコルでブロードキャストを行って名前解決を行うような場合では､そのリンクは利用した特定のプロトコル上において一つのネームソースとなる｡また､ノードが独自に管理するデータベースもネームソースにあたる。

汎用名前解決システムは､現在インターネットを利用するほぼすべてのアプリケーションが名前解決のために利用する DNS の仕組みを利用するものとする。具体的には、アプリケーションが使用する基本 API ライブラ

リ（リゾルバライブラリ）が DNS cache server に対して行う DNS 問い合わせのための通信を利用し、アプリケーションに対して DNS cache server として振舞うことで名前解決機能を提供することとなる。

各ノード上では DNS への問い合わせパケットを理解するデーモンプログラムを起動しておく。このデーモンプログラムは port 53 のパケットを受信し、名前要求の内容を理解し、適切な名前解決を行い応答するプログラムであり、これが汎用名前解決システムとなる。ノードでは、近傍の DNS cache server として自己のノード、すなわち IPv6 アドレスでは::1、IPv4 アドレスでは 127.0.0.1 を指定する。

各アプリケーションは従来どおりの動作を行う。すなわち、名前の解決のためにリゾルバライブラリを呼ぶ。リゾルバライブラリは指定された DNS cache server、すなわち自己に対して DNS の問い合わせパケットを送信する。このパケットは汎用名前解決システムによって受信される。汎用名前解決システムは、様々な名前解決機構に準じたモジュールを内包している。これら複数のモジュールに問い合わせることにより、適切なネームソースからの応答を選択し、リゾルバライブラリに対して応答する。リゾルバライブラリはこの応答をアプリケーションに伝える。

この結果として、汎用名前解決システムではアプリケーションや OS が提供するリゾルバライブラリにまったく手を加える必要がなく、複数の名前解決機構を利用できる。また、モジュールを追加することによって、新しい名前解決のためのメカニズムが提案された場合においてもモジュールの追加だけで各アプリケーションは新しい名前解決メカニズムを利用でき、効率的に実環境において各名前解決メカニズムの検証を行うことができる。

汎用名前解決システムは、主に汎用名前解決エンジンとモジュールに分けられる。汎用名前解決エンジンは、リゾルバライブラリからの要求を受け、モジュールに各要求を伝え、その複数の結果から適切に応答を選び出しリゾルバライブラリに返すのが主な役割である。図 2 に汎用名前解決システムの概略を示す。

汎用名前解決エンジンは、次の方針に基づいて設計を行った。汎用名前解決エンジンの概略を、図 3 に示す。

- 容易に実装可能となるようにモジュラリティの高い構成を持つこと。
- 既存のアプリケーションなどにインパクトを与えずに導入することが可能であること。
- 多くのプラットフォーム上で稼動させられるように移植性が高いこと。
- 複数の名前解決方式を同時に利用可能であること。
- 名前解決に対して得られた複数の結果に対し、適応的に応答を選択する機能を持つこと。

以下、各サブユニットについて説明する。

Resolver-Engine Interface

リゾルバライブラリからの要求パケットを受け付け、内部形式に展開する。また、モジュールからの応答をリゾルバライブラリへと伝える。リゾルバライブラリに対する反応も Resolver-Engine Interface の役割である。これにはまず各モジュールの処理とは独立に、最大のタイムアウト時間を決め、各モジュールの反応を待たずとも自立的に名前解決の失敗をリゾルバに通知するという役割がある。
汎用名前解決機構検証システム内の内部表現形式は、圧縮を利用しない DNS パケットフォーマットを利用する。

Policy Controller

要求された名前に対して、複数のモジュールに問い合わせた場合、当然複数個の応答が返ってくることが考えられる。どのモジュールを利用するか、どのような応答を優先するか、どのモジュールからの応答を優先するかを管理するのが Policy Controller の役割である。Policy は基本的に以下の要素によって管理される。以後この要素を Policy Selector と呼ぶ。

- Protocol
  名前の解決に使用するプロトコル。例えば DNS。
- Type
  問い合わせが解決を要求している型。RFC1035、RFC1886 等の QTYPE に準ずる。
- 名前
  問い合わせが解決を要求している名前。

各モジュールはそれぞれが解決可能な Policy Selector の要素を Policy Controller に伝える。Policy Controller は、Resolver-Engine Interface から受けた要求を元に問い合わせ可能なモジュールを選択し、それらのモジュールに対して重み付けを行ったうえで各モジュールに問い合わせ要求を渡す。各モジュールに問い合わせ要求を渡す作業は実際には Query Manager を通して行われる。

Query Manager

名前の解決は非同期に行われる。例えば 3 つのアプリケーション A・B・C が同時に問い合わせを発行した場合には、これらは見かけ上同時に処理される。そのため、現在どのような応答をモジュールに対して待っているのかといった管理が必要となる。この処理を行うのが Query Manager である。現在どのような問い合わせが解決中であるか、また各問い合わせに対して現在どのモジュールから返ってきているかを管理する。

Module Manager

現在どのようなモジュールが接続されているかを識別し管理する。モジュールは接続がモジュール情報をエンジンに対して宣言する。Module Manager はこの情報を管理し、また接続情報も保持する。

Module Interface
各モジュールは汎用名前解決エンジンとは別に動作するプログラムであるので、エンジンとモジュールを結ぶインタフェースが必要となる。Module Interface はこのインタフェースの抽象化を行う層である。
今回試作したプロトタイプでは、各モジュールはローカルアドレスに対する TCP 接続によってエンジンとリンクする。Module Interface はモジュールからの Connection・Disconnection を扱う。また、Query Manager から、あるいはモジュールからのデータの Serialize・Packetize を行う。

名前解決モジュールは基本的にプロトコルごとに用意される。プロトコル単位でモジュールの内容は大きく異なる。基本的なモジュールフレームワークは図 4 に示すような構造を持ち、以下の要素から成り立つ。

Module Interface
エンジン側の Module Interface の stub となる層。また、プロトコル依存である Module Core で利用されるデータ構造に変換する。

Module Core
プロトコルごとの処理が記述される。

Per-Name-Source Connector
汎用名前解決機構検証システムにおいては、一つのプロトコル上で概念上複数のネームソースを持つことができる。各モジュールはどのネームソースからの応答かを名前解決エンジンに伝える必要がある。すなわち、応答はモジュール単位ではなくネームソース単位で管理される。

汎用名前解決エンジンとモジュールの間は、ローカルアドレスを使用した TCP コネクションで接続される。汎用名前解決エンジンは事前に定められた port を passive open し、各モジュールはその port に対して接続をする。

モジュールは接続後、以下の情報をモジュール情報として宣言する。

- プロトコル識別番号
  そのモジュールがどのようなプロトコルを利用するかを示す番号。プロトコル識別番号は別途定義される。
- モジュール名
  そのモジュール自身の名前。7 ビット ASCII で表記される。
- モジュールバージョン番号
  そのモジュールのバージョン番号。

・ タイプセレクタ
そのモジュールが解決できる Query Type。512 ビットのビットマスクで表現され、ビットが立っている部分が解決可能な Query Type である。RFC1035 的には 256 ビットしか使用しない。上位 256 ビットはローカルな拡張のために利用される。

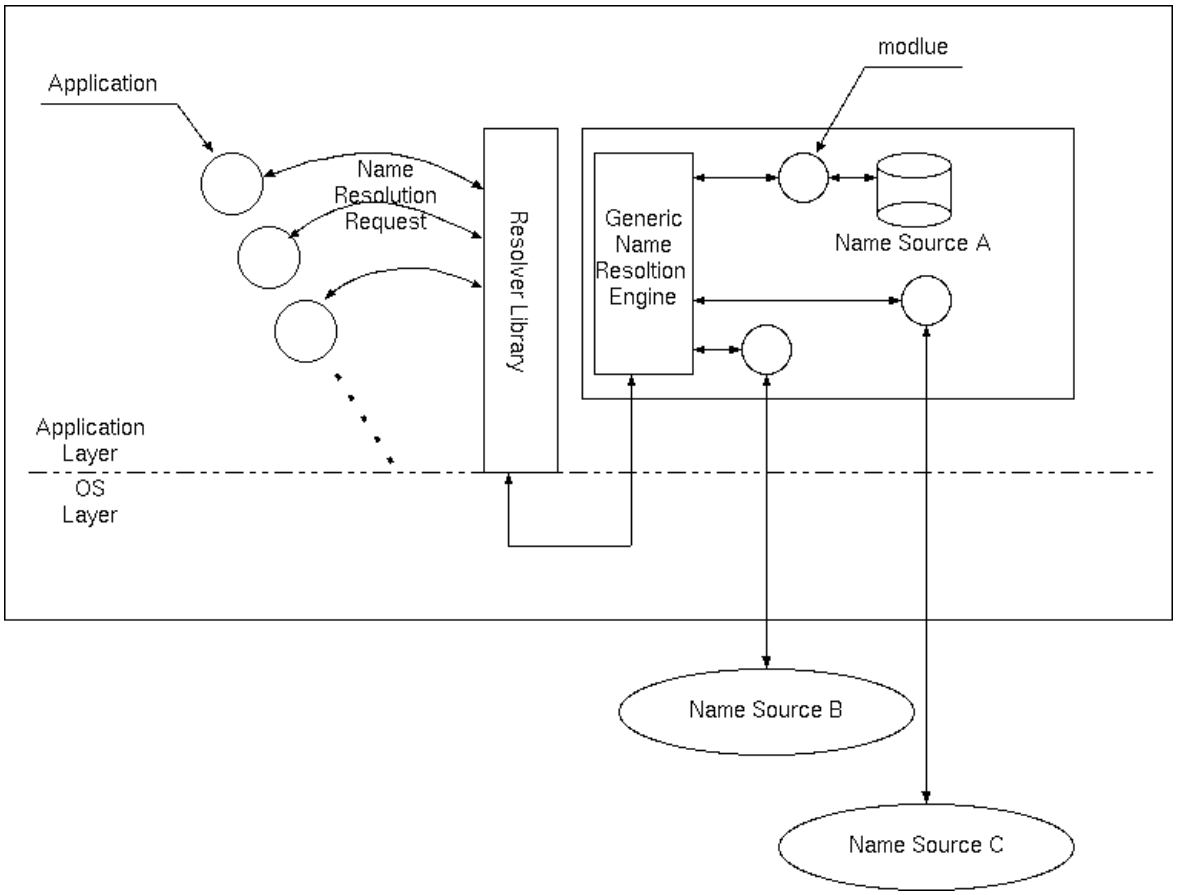


図 1　汎用名前解決システムの概要

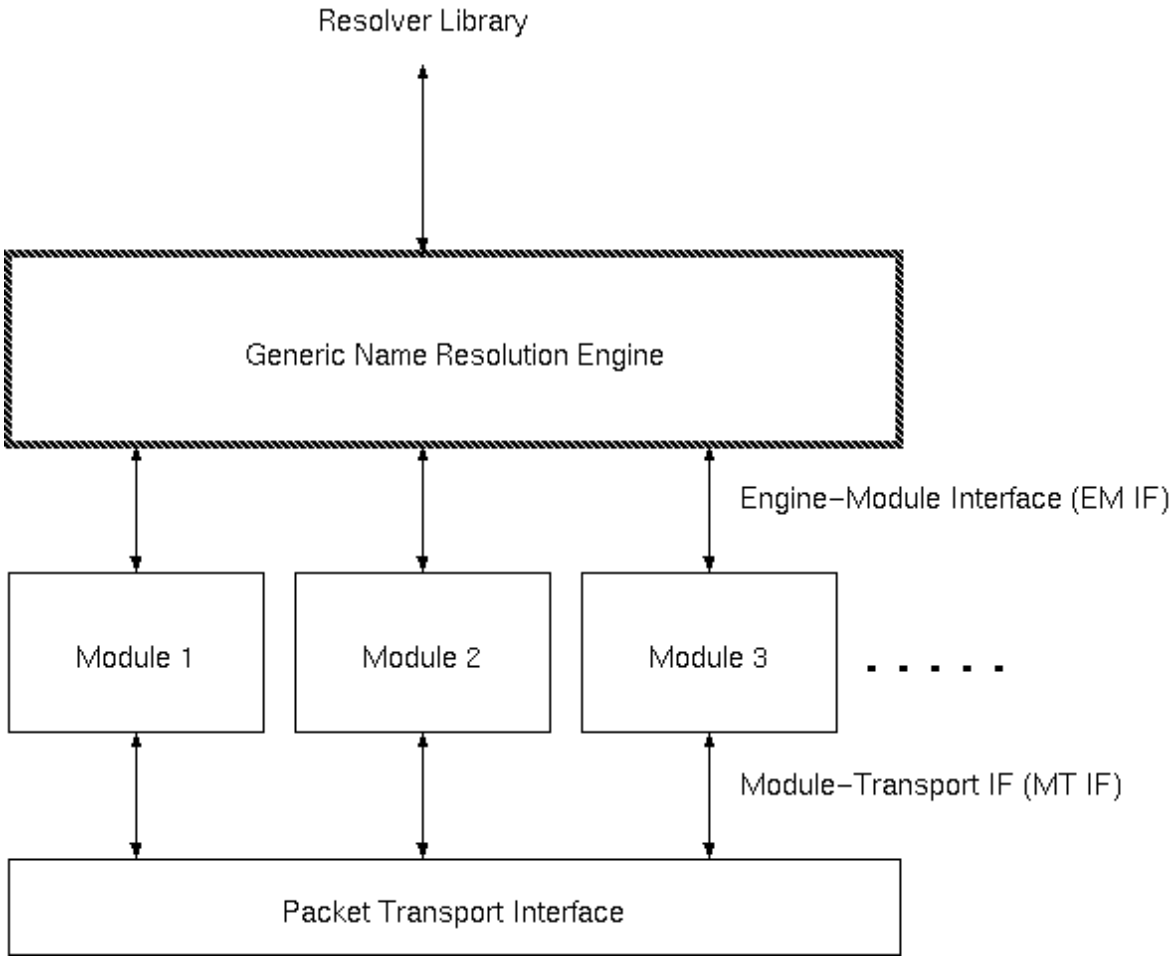


図 2　汎用名前解決システムの概略

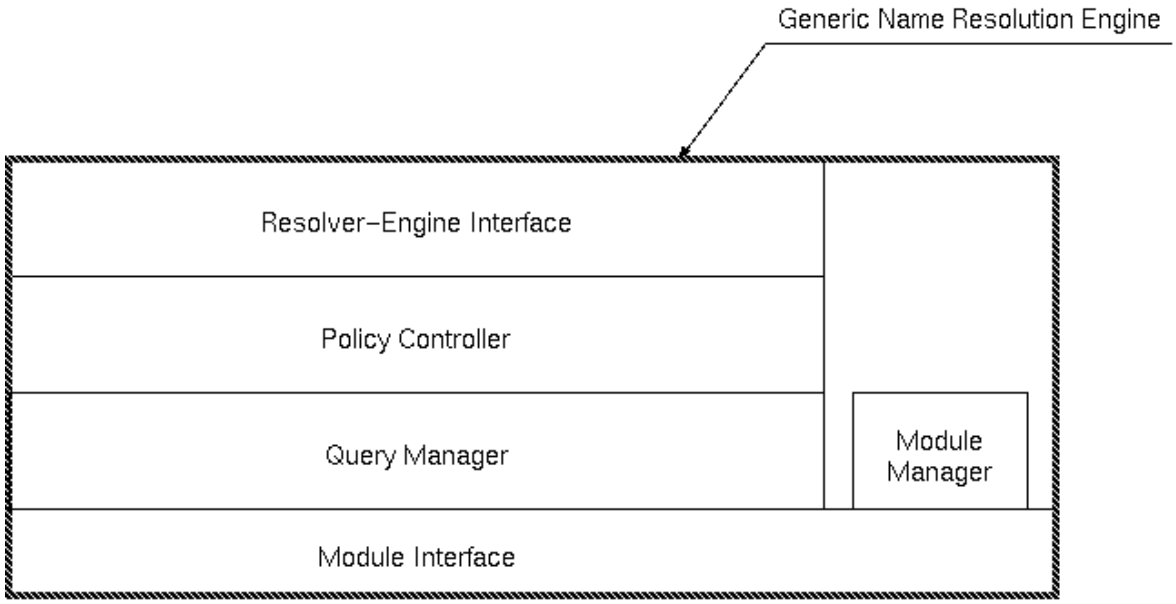


図 3　汎用名前解決エンジン概要

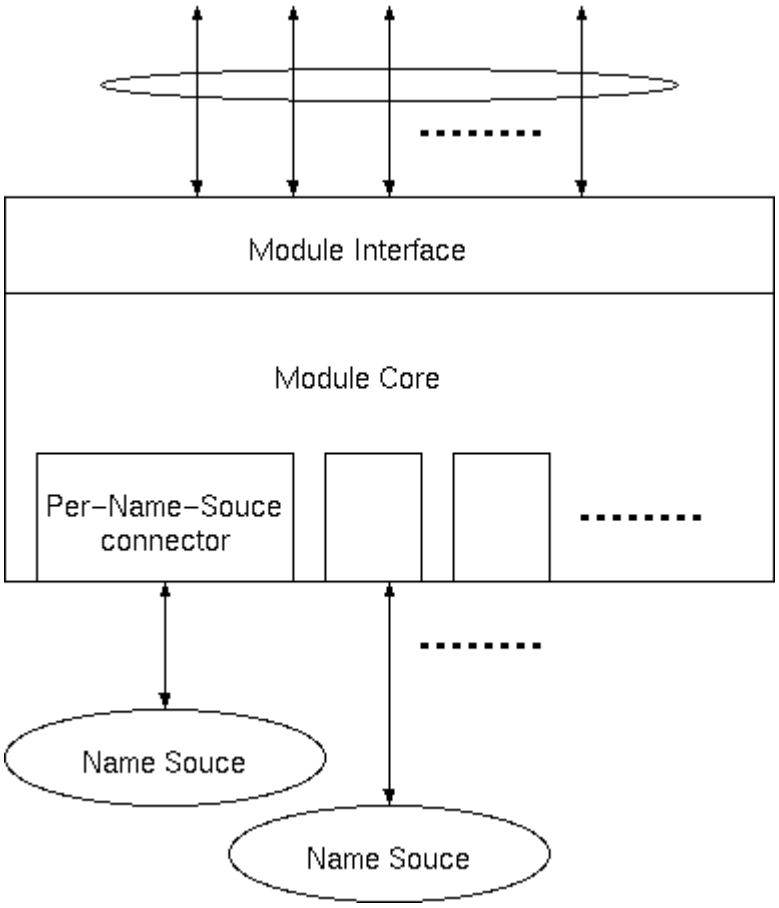


図 4　モジュールフレームワーク

5-1-2　DNS が利用できない環境でも近隣のノードの名前を解決可能な名前解決モジュール

内容　DNS を利用できない場合において、近隣のノードの名前の解決に利用できる方法として現在 IETF で議論されているプロトコルのうち、「IPv6 Node Information Queries[4]」を利用して近隣のノードの名前を解決する名前解決モジュールを実装する。

状況　名前解決モジュールの試作開発を完了した。

成果　以下の解決方式を使用する名前解決モジュールを実装した。

- AAAA の解決には、ICMP Node Information Query の Query Type を 3(Node Address)としたパケットを All-Nodes Multicast Addresses(FF02::1)を宛先アドレスとして送信する。これから得られた名前を、AAAA によって問い合わせられた名前と比較を行い、マッチした名前を持つアドレスを AAAA の応答として返す。

- PTR の解決には、ICMP Node Information Query の Query Type を 2(Node Name)としたパケットを PTR で指定された名前から生成できる IPv6 アドレスに対して送信する。これから得られた名前を、PTR の応答として返す。

詳細　本名前解決モジュールで使用する ICMP Node Information Query による名前解決機構の動作について、概略を図 5 に示す。

[4] draft-ietf-ipngwg-icmp-name-lookups-09.txt

Network Environment

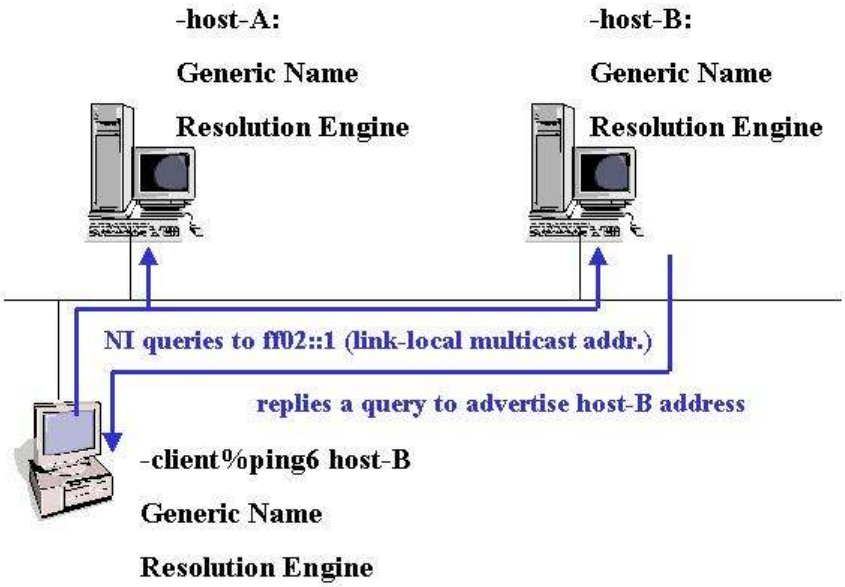


図 5　ICMP Node Information Queries による名前解決

5-1-3　近隣の DNS サーバを動的に発見可能な名前解決モジュール

内容　現在 IETF で議論されている「Well known site local unicast addresses for DNS resolver[5]」に基づき、Well-known なアドレスを DNS サーバに割り当て、そのアドレスに対して問い合わせを行うことによって自動的な DNS サーバの発見を行う名前解決モジュールを実装する。

状況　名前解決モジュールの試作開発を完了した。

成果　以下の解決方式を使用する名前解決モジュールを実装した。

- 個々の Well-known アドレスに対して Per-Name-Source Connector を用意する。
- Well-known なアドレスに対して要求を forward する。アドレスは順に試行し、原則的に応答のあったものを使う。
- anycast においてはアドレスの不一致が起きるため、anycast を許す条件下では、anycast アドレスと発見された実サーバのアドレス、および要求した問い合わせの ID によって管理することによりアドレスの不一致を処理する。

詳細　本名前解決モジュールで使用する Well known site local address による DNS server discovery の動作について、概略を図 6 に示す。

[5] draft-ietf-ipv6-dns-discovery-06.txt

server1: fec0::X (anycast), fec0::1 (unicast)

server2: fec0::X (anycast), fec0::2 (unicast)

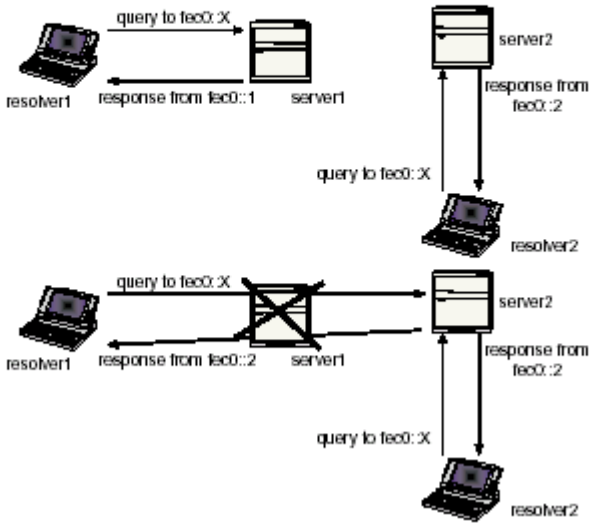


図 6　Well-known site local address による DNS server discovery の動作例

5-1-4　複数のサーバから異なる応答を受けた場合に適切な応答を選択可能な名前解決モジュール

内容　近年、網の安全性を確保するためにしばしば導入されるファイアウォールを前提とした環境において、ファイアウォール内部の DNS サーバに問い合わせた場合とファイアウォール外部で問い合わせた場合に応答の結果が異なるケースを想定し、ある問い合わせに対して適切な応答を選択する名前解決モジュールを実装する。

状況　名前解決モジュールの試作開発を完了した。

成果　以下の解決方式を使用する名前解決モジュールを実装した。

- 応答の内容がNXDomainでなく、かつAnswerRRが0でもなく、かつエラーでもないものを「良い応答」とする。

- 得られた応答のうち、良い応答であり、かつその良い応答を答えたサーバよりもより高い優先度を持つサーバがないか、あるいはそのより高い優先度のサーバの応答が良い仮応答でないのであれば、そのサーバからの仮応答を最適とする。

- 得られた仮応答がすべて「良い応答」ではなく、サーバ全員の応答をすべて持っているのであれば、得られた応答の中にNXDomainがあれば最適応答はその応答とし、そうではなく、得られた応答の中に AnswerRR が 0 であるものがあれば、最適応答はその応答とし、得られた応答がすべてエラーであればエラーが最適応答とする。

詳細　ファイアウォール内でファイアウォール外のホストに関する名前解決を行った場合に、OS 付属のリゾルバを使用した場合の挙動を図 7 に、本名前解決モジュールを汎用名前解決モジュールとあわせて使用した場合の挙動を図 8 に示す。

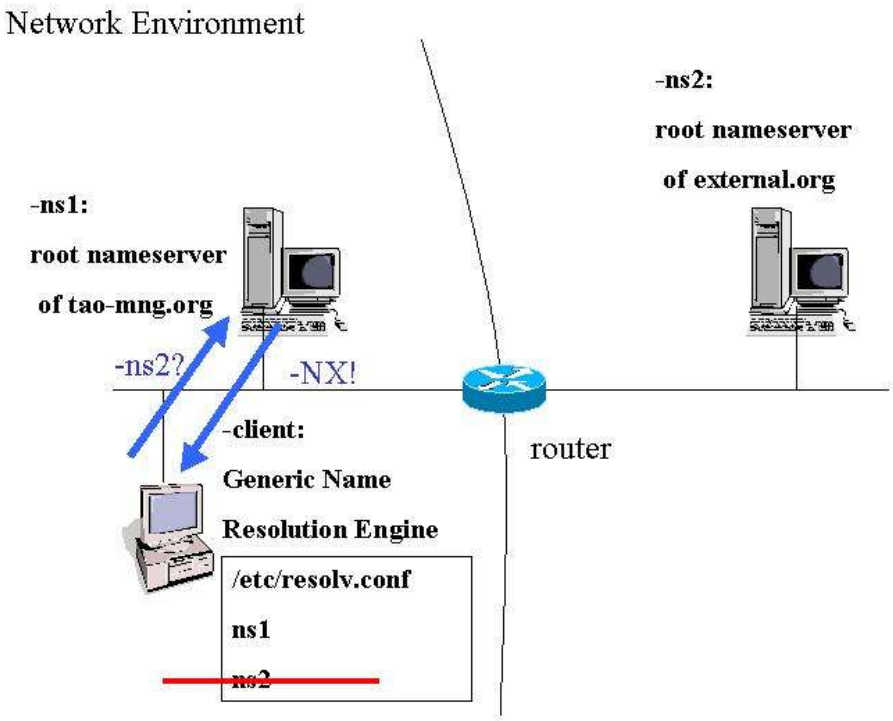


図 7　OS 付属のリゾルバの場合

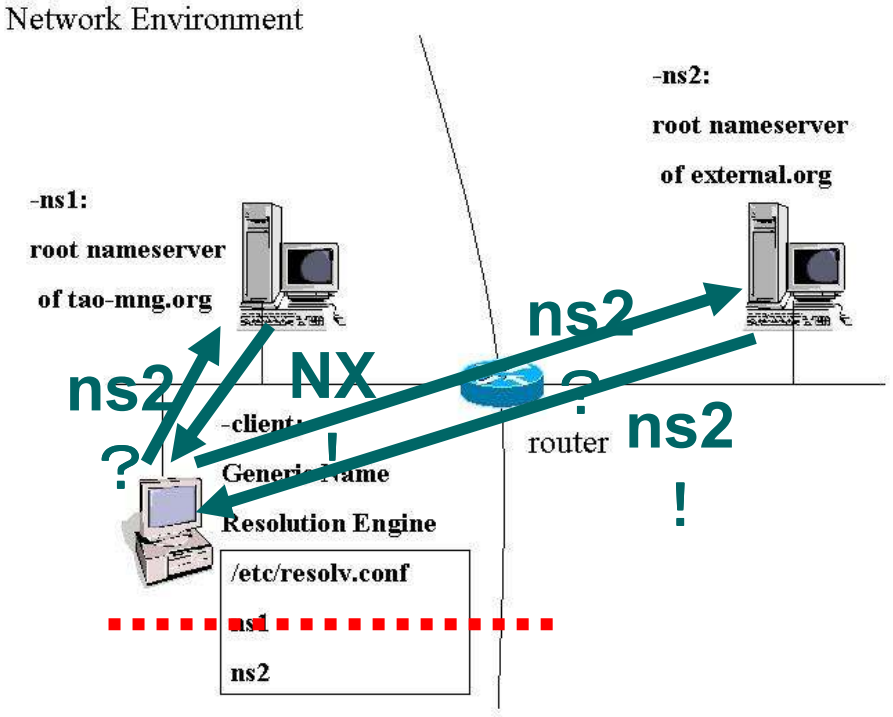


図 8　汎用名前解決エンジンの場合

5-2　単体動作検証

5-1　で試作した汎用名前解決エンジンおよび名前解決モジュール群に対し、汎用名前解決エンジン上で各名前解決モジュールを統合して動作させ、その有効性を確認した。

5-2-1　名前解決モジュールと汎用名前解決エンジンの組み合わせ

内容　5-1-1　で実装した汎用名前解決エンジン上に、5-1-2　で実装した名前解決モジュール、5-1-3　で実装した名前解決モジュール、5-1-4　で実装した名前解決モジュールをそれぞれ統合し、各名前解決モジュールを利用した名前解決機能の動作検証を行う。

状況　試作開発を行った各名前解決モジュールごとに汎用名前解決エンジンと接続した単体動作検証を行い、正常に動作していること、および機能の有効性を確認した。

成果　各名前解決モジュールを汎用名前解決エンジンと組み合わせずに使用した場合と、組み合わせて使用した場合における名前解決試験を実施し、以下の結果を得た。

- DNS が利用できない環境でも近隣のノードの名前を解決可能な名前解決モジュール
  - ■ 汎用名前解決エンジンを使用しない場合
    単体動作試験環境では名前解決に失敗した。
  - ■ 汎用名前解決エンジンを使用した場合
    単体動作試験環境では名前解決に成功した。

- 近隣の DNS サーバを動的に発見可能な名前解決モジュール
  - ■ 汎用名前解決エンジンを使用しない場合
    単体動作試験環境では名前解決に成功した。
  - ■ 汎用名前解決エンジンを使用した場合
    単体動作試験環境では名前解決に成功した。

- 複数のサーバから異なる応答を受けた場合に適切な応答を選択可能な名前解決モジュール
  - ■ 汎用名前解決エンジンを使用しない場合
    単体動作試験環境では名前解決には成功したが解決するまでに 12 秒あまりかかっていた。
  - ■ 汎用名前解決エンジンを使用した場合
    単体動作試験環境では名前解決に成功した。

詳細　本動作検証は、図 9 に示す試験用ネットワーク環境上で行った。ここで、cl2 は tao-mng.org から external.org へ移動するものとする。

試験用ネットワークの機器構成を次に示す。

- cl1・cl2・cl3・ns1・ns2
  FreeBSD-4.7R＋KAME-SNAP030123
- ns3

FreeBSD-4.6.2R

・ rt
CISCO1605R

・ ns1・ns2
BIND-9.1.2

ここで、汎用名前解決エンジンを cl1・cl2 にインストールして、汎用名前解決エンジンを使用しない場合と使用した場合とで､それぞれ以下の手順に従い動作検証を行った。

・ DNS が利用できない環境でも近隣のノードの名前を解決可能な名前解決モジュールの動作検証

cl1 から、cl2.tao-mng.org に対する名前解決を行った。

・ 近隣の DNS サーバを動的に発見可能な名前解決モジュールの動作検証

tao-mng.org 側に接続した cl2 から、cl1.tao-mng.org に対する名前解決を行った。

external.org 側に接続した cl2 から、ns3.external.org に対する名前解決を行った。

・ 複数のサーバから異なる応答を受けた場合に適切な応答を選択可能な名前解決モジュールの動作検証

cl1 から、cl2.tao-mng.org に対する名前解決を行った。

cl1 から、ns3.external.org に対する名前解決を行った。

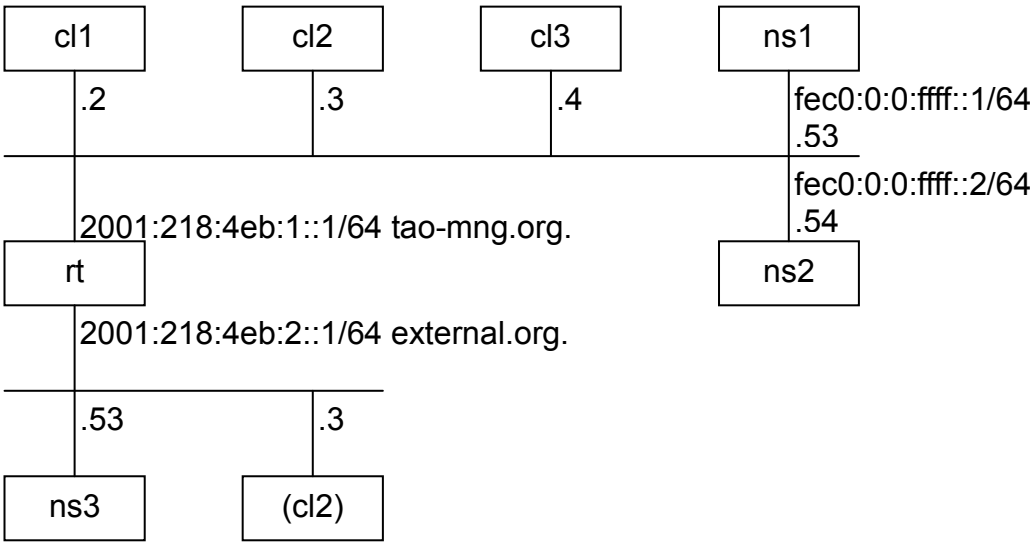


図 9　試験用ネットワーク環境

5-3　総括

5-3-1　本研究開発の成果について

本委託業務で行った研究開発は、当初予定されていた平成１４年度の研究開発内容をすべて実施することができ、成果物の有効性が確認できたという意味で成功したものと考えられる。

本研究開発は、3　に記した研究開発全体計画の中の平成１４年度の研究開発に相当する。本研究開発が成功裏に終了したことにより、引き続き平成１５年度の研究開発を実施する準備が整ったといえる。

5-3-2　平成１５年度の研究開発について

3　に記した研究開発全体計画において、平成１５年度の研究開発内容としては、平成１４年度までの研究開発の成果を踏まえた上で、最終的に以下の目標を実現するものとする。

(1) 複数の名前解決モジュールを組み込み、それらを選択的に利用可能な汎用名前解決エンジンの実現（3-1　（ア）に対応）
(2) DNS が利用できない環境でも近隣のノードの名前を解決可能な名前解決モジュールの作成（3-1　（イ―１）に対応）
(3) 近隣の DNS サーバを動的に発見可能な名前解決モジュールの作成（3-1　（イ―２）に対応）
(4) 近隣のDNSサーバ障害発生時に適応的にDNSサーバを選択することによって、効率的な名前解決を可能とするモジュールの作成（3-1（イ―３）に対応）
(5) ファイアウォール等で分断されたネットワーク環境において、適切な応答を選択可能な名前解決モジュールの作成（3-1　（ウ―１）に対応）
(6) 複数のローカルデータベースを利用した名前解決がグローバルなDNS システムと透過的に利用可能な名前解決モジュールの作成（3-1（ウ―２）に対応）
(7) 各名前解決モジュールを組み込んだ汎用名前解決エンジンの統合動作検証の完了

5-3-3　本研究開発の成果がもたらす効果について

近年になって名前解決のための新しい技術やプロトコルが提案されているにもかかわらず、あまり実際に使用されていない。それは実装がしにくいことに一つ原因がある。これらの状況を鑑み、本研究開発ではまず様々な名前解決方式を試作、検証できるための汎用名前解決エンジンを開発する。これは容易に従来のアプリケーションと組み合わせて利用可能であり、プラットフォームに依存しない。さらに、この名前解決エンジンの有用性を検証するために上記問題を解決する名前解決方

式の実装をエンジンに組み込んで評価、検証する。

本研究開発の成果をベースとして実施される予定の研究開発全体計画の成果物を利用することにより、アプリケーションや OS が提供するリゾルバライブラリにまったく手を加える必要なしに、複数の名前解決機構を利用できる。また、モジュールを追加することによって、新しい名前解決のためのメカニズムが提案された場合においてもモジュールの追加だけで各アプリケーションは新しい名前解決メカニズムを利用でき、効率的に実環境において各名前解決メカニズムの検証を行うことができる。

5-3-4　まとめ

本研究開発では、様々な名前解決メカニズムを統一的に扱う仕組みである汎用名前解決エンジンと、そのエンジン上で利用可能なモバイルサポートのための名前解決モジュールおよびセキュリティやプライバシーを考慮した名前解決モジュールの試作を行い、各名前解決モジュールと汎用名前解決エンジンとを組み合わせた単体動作検証を行った。

本研究開発の成果は、引き続き実施する予定である平成１５年度の研究開発のベースとなるものであり、本研究開発が成功裏に終了したことで、研究開発全体計画を予定通りに遂行させることが可能となった。

参考資料、参考文献

P. Mockapetris,
"DOMAIN NAMES - CONCEPTS AND FACILITIES",
Request for Comments,
rfc1034.txt,
November 1987.

R. Gilligan, S. Thomson, J. Bound, W. Stevens,
"Basic Socket Interface Extensions for IPv6",
Request for Comments,
rfc2553.txt,
March 1999.

Levon Esibov, Dave Thaler,
"Linklocal Multicast Name Resolution (LLMNR)",
Internet draft (work in progress),
draft-ietf-dnsext-mdns-12.txt,
August 2002.

Matt Crawford,
"IPv6 Node Information Queries",
Internet draft (work in progress),
draft-ietf-ipngwg-icmp-name-lookups-09.txt,
May 2002.

Alain Durand, Jun-ichiro itojun Hagino, Dave Thaler,
"Well known site local unicast addresses for DNS resolver",
Internet draft (work in progress),
draft-ietf-ipv6-dns-discovery-06.txt
August 2002.

(添付資料)

1. 研究発表、講演、文献一覧

研究発表は以下の 1 件を実施

情報処理学会　第 3 回分散システム／インターネット運用技術研究会
開催日：2002 年 10 月 18 日(金)：岡山大学)
題名：現在の名前解決システムの課題と汎用名前解決エンジンの提案
著者：山田 竜也、 嶋田 雄二郎、島津 伸行、倉富 修、岡 光秋、岡本 利夫、栄 光宏
予稿：情報処理学会　分散システム／インターネット運用技術研究報告
No.27-5(2002.10.18), pp.25-30 (査読なし)