平成１３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

参考資料、参考文献

（添付資料）

1　研究発表、講演、文献等一覧

１　研究開発課題の背景

IPv6 の普及に伴い，その広大なアドレス空間を利用して今後多種多様なデバイスが数多くネットワークに接続されると考えられる．また，その接続形態も有線，無線など多岐にわたり，デバイスも移動し，また動的にアドホックなネットワークを形成して通信するなど，ネットワークを柔軟かつ動的に構成しながら運用する形態が普通に行われるようになっていくと考えられる．さらに IPv6 特有な事象としてアドレスのリナンバリング(renumbering)があり，デバイスに割り付けられるアドレスが変化することがある．これらの背景を考慮すると，IPv6 では名前解決の技術がこれまでに増して非常に重要になる．しかし，現在の名前解決のシステムには数多くの課題がある．

まず，現在はインターネットにおける名前解決は DNS(Domain Name System)に頼っているが，外部接続のないアドホックなネットワークにおいても通信相手を名前で指定することができないと通信相手の指定が繁雑で，かつ機器の設定に負担がかかるなど，ユーザに大きな負荷を与えてしまう．一般に，グローバルなリソースにアクセスする場合には DNS は必須であるが，現在のインターネット環境では DNS サーバを静的にどこかに設定するのが一般的であり，これはネットワークの管理コストを引き上げる要因になる．また，DNS サーバに障害があったり，クライアントの接続環境がかわったりすると名前の解決ができなくなることがあり，この接続環境の変化をクライアントが自分で検出できないと，具体的にはタイムアウトまでの間長く待たされてしまうといったように，クライアントの性能と利便性に大きな悪影響が生じる．

そして，企業網などでは，今後もファイアウォール環境が続くと考えられる．そのような場合，セキュリティの観点から内部と外部で名前空間の構成木が異なる場合があり，社内網の名前は外部では解決できないといったような事象が発生する．よってネットワーク上を移動する機器を操作するユーザは，そのような違いを意識すること強いられるために，現在は利用上の負担が大きい．

また，名前の公開にはプライバシー問題も考えられる．例えば情報家電などへの応用においては，電源の制御といったインターネットからのアクセスを簡便にするために家庭の家電機器に名前を付与することが考えられるが，これを一般に公開してしまうと，各家庭にどのような家電機器があるかというプライバシーを漏洩してしまう危惧がある．しかしアドレスを直接入力してのアクセスはリナンバリング等でアドレスが変更された場合のことを考慮するとユーザへの負担が大きい．このため，プライバシーを保護できる名前解決方式が必要となるが，現在の DNS の枠組では，特定の人間にだけ名前情報を公開するということは難しい．加えて家庭での使用といった利用形態を考えると名前解決のために複雑な手続きをとることは現実的ではない．

２　研究開発分野の現状

DNS 関連のセキュリティの問題として，IETF で標準化が薦められている DNSSEC（DNS 問い合わせに関するセキュリティ機能）とトランスレータの相性が悪いということが指摘されている．現在，IPv6-IPv4 トランスレータは DNS の応答を書き換えることによって機能を実現しているが，この仕様を継続すると， DNSSEC との併用ができず将来，確実にセキュリティ上の問題が発生すると思われる．またこれに関連し，通常 DNS を使用する場合には，近隣のキャッシュサーバに問い合わせを行うが，DNSSEC が利用できない場合にはそこで悪意を持つ第三者が虚偽の応答を利用して攻撃を行う可能性もあるため，これを保護するシステムも必要となる．

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いくつか提案されているにもかかわらず，実際にはほとんど使用されていない．それはそのような新しい実装を，実環境で実際に使用することが難しいという現状が大きな原因の一つである．

これらの状況を鑑み，我々はまずさまざま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．

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、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。

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）及び（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

（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（４－１（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（４－１（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（４－１（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（４－１（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（４－１（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（４－１（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汎用名前解決エンジンの試作完了（４－１（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試作完了（４－１（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試作完了（４－１（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試作完了（４－１（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

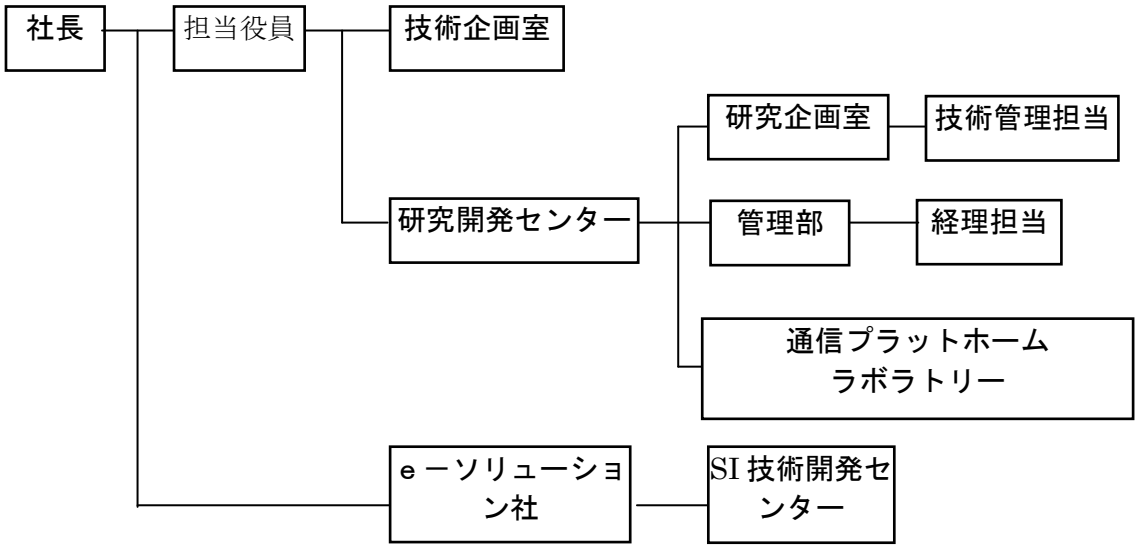
（金額は非公表）

研究開発項目	13 年度	14 年度	15 年度	年度	年度	計	備　考
【研究開発課題名】 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 【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	調査 ・仕様検討 →	機能試作 ・評価 → → 13 →	機能試作 ・評価 統合動作検証 → → →				
間接経費							
合　　計							

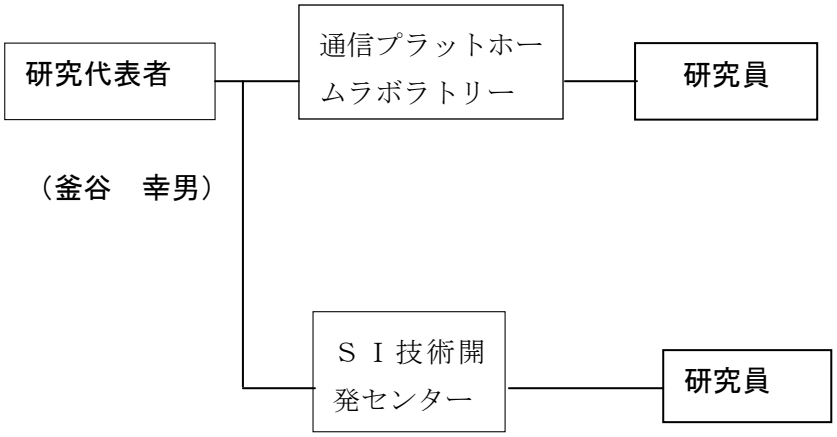
注）１　経費は研究開発項目毎に消費税を含めた額で計上。また、間接経費は直接経費の３０％で計上（消費税を含む。）。
　　２　備考欄に再委託先機関名を記載

3-4 研究開発体制

研究開発管理体制



研究開発実施体制



（永見　健一）
（石山　政浩）
（神明　達哉）
（尾崎　哲）
（神田　充）
（勝部　泰弘）
（松澤　茂雄）
（市江　晃）
（上原　清彦）
（村井　俊雄）

（岡本　利夫）
（栄　光宏）
（岡　光秋）
（倉富　修）
（嶋田　雄二郎）
（島津　伸行）
（山田　竜也）

４　研究開発の概要（平成 13 年度）

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。

4-1-2　研究開発課題実施計画

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）及び（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。

4-2　研究開発の実施内容

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た。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た.

５　研究開発実施状況（平成 13 年度）

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た。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た。

5-1　汎用名前解決エンジン(現状の DNS 実装と IPv6 との関係)

5-1-1　DNS の IPv6 対応状況

DNS の「IPv6 対応」とは、リソースレコード(RR)でのレベルの対応と，トランスポートの対応がある．Internet Software Consortium が開発している Berkeley Internet NameDomain(BIND)の version 9 は IPv6 に対応している．機能は以下の通り．

（１）IPv6 アドレスのためのリソースレコード(RR)

RFC 1886 に定義してある正引き(AAAA RR)および逆引き(PTR RR)に対応している．この結果次のような変換が可能となる．

- 正引き

  www.kame.net. IN AAAA => 3ffe:501:4819:2000:280:adff:fe71:81fc

- 逆引き

  上位ドメイン: ip6.int.
  4 ビットずつを 1 ラベルとし、下位バイトから逆順に 32 階層で表現
  2001:0200:0000:4819:0280:adff:fe71:81fc の逆引きは次のようになる
  $ORIGIN 9.1.8.4.0.0.0.0.0.0.2.0.1.0.0.2.ip6.int.
  c.f.1.8.1.7.e.f.f.f.d.a.0.8.2.0 IN PTR www.kame.net.

（２）トランスポートでの対応

トランスポートでの対応は，問い合わせ、応答を IPv6 パケットで行うことをさす．この結果三つのケースが考えられる．

1. IPv6 用の RR, IPv4 トランスポート
2. IPv6 用の RR, IPv6 トランスポート
3. IPv4 用の RR, IPv6 トランスポート

IPv4 用の RR および IPv4 トランスポートは IPv6 とは無関係となる．現在稼働している実装の多くは 1 のみである． 1.のみで実運用上は問題ないが、IPv4 に依存しない環境構築には 2.が重要となる.また,2.をできる実装は通常 3.でも運用可能となる． IPv6 トランスポートの利用の視点から見た場合,現在運用されているネームサーバの現状は次のようになっている.

Top Level Domain(TLD): IPv6 アドレスを持つネームサーバの数
ルート, gTLD: 13 ネームサーバ中ゼロ
改善の試みが進展中
ccTLD: 243 ゾーン、584 ネームサーバ中
2 ゾーン、3 サーバ
ルートゾーンからのグルーRR なし

また，現在の IPv6 対応 DNS 関連プログラムの状況は以下の通り．

- サーバ

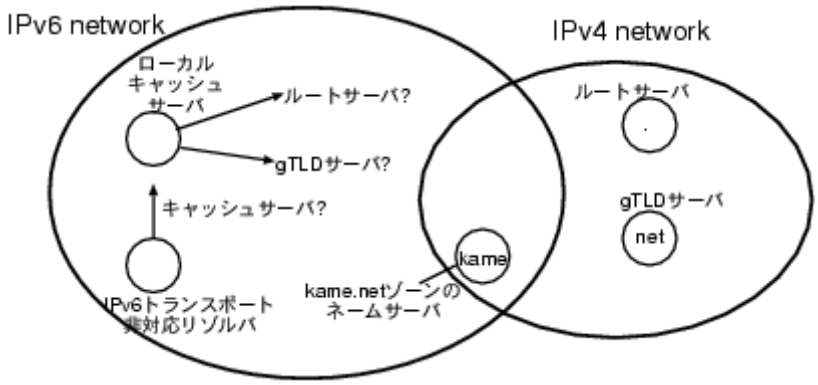
  BIND version 9 (BIND9)

- リゾルバ

  対応: FreeBSD/NetBSD/OpenBSD/Linux

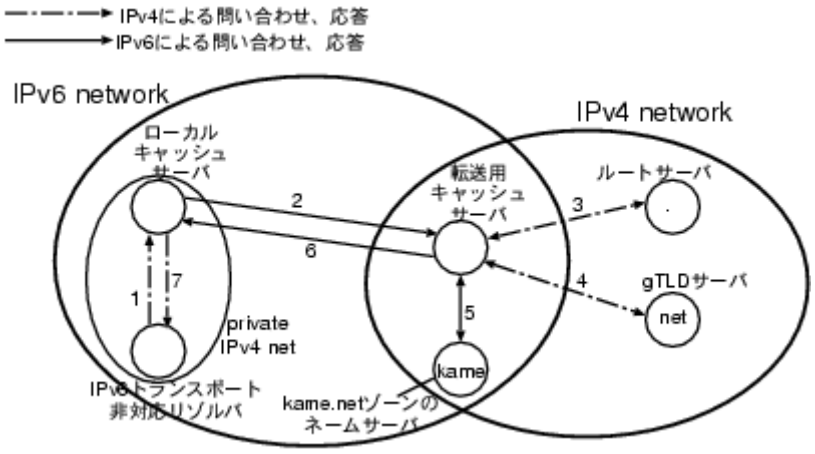
BIND9 lwres: アプリケーションの再コンパイルが必要
BIND8, 9 libbind
未対応: BSD/OS
Solaris
Windows

IPv6 ネットワークから見た場合の DNS の現状の問題点は, IPv6 トランスポートに対応しないスタブリゾルバはキャッシュサーバへの問い合わせができないという点と, IPv6 のみのネットワークにいるキャッシュサーバは名前解決できないという点である.



［図１］

この問題に対する運用での対応方法としては, 名前解決用の IPv4 ローカルネットワークを組むか, デュアルスタックの転送用キャッシュサーバを用意するという手が考えられる.



［図２］

次に, 現在の BIND9 の IPv6 トランスポートを述べる.

以下の機能は IPv4 と同等に行なえる

- 問い合わせ
- 応答
- ゾーン転送
- ACL
- 制御コマンド

ただし，IPv6 トランスポートの有効化を行なった場合には，wildcard bind されたソケットですべての UDP 応答を処理するために 2 つの named を同時に起動するのは不可能となっている．

また，IPv4-mapped IPv6 アドレスが使われることもある． IPv4-mapped IPv6 アドレスとは，::ffff:x.y.z.w (x.y.z.w は IPv4 アドレス)の形式をさす．この場合 IPv6 ソケットで IPv4 パケットを受信する実装がある．すなわち， IPv4 アドレスを"mapped" IPv6 で表現するためにアクセス制御が複雑になる．

5-1-2 データ長の制限と EDNS0

現在の DNS データ長(UDP)の上限は 512 バイトであるため,大きなデータが扱えない．よって， IPv6, DNSSEC への対応には不十分と言える． 例えばルートサーバに www.kame.net.の A RR を問い合わせることを考える．この場合の応答は gTLD サーバに対する NS RR とグルーRR になるが，現状は 13 サーバであるため，459 バイトとなる．しかし，AAAA グルーRR も付けるなら 5 サーバが限界 (471 バイト)となってしまう．仮に 13 サーバのままだとすると 823 バイト必要となる．

この解決方法のひとつとして Extension Mechanisms for DNS(EDNS0, RFC 2671)がある．これは， OPT RR とよばれるリソースレコードを追加し，問い合わせ時に、応答用の受信バッファ長を通知するようにする．応答には、通知された長さまでデータを詰めてよい．

現在の EDNS0 の実装例は BIND 9(サーバ、libbind) である． BIND version 8 は、EDNS0 付きの問い合わせにはエラーを返す．

☆ BIND 9 における EDNS0 の現状について

BIND 9 では新規のサーバへの問い合わせには常に OPT RR を付ける．エラーが返ってきた場合は OPT RR なしでやり直す．エラーを返したサーバは記憶する．次回からは OPT RR なしで問い合わせる． この記憶は 24 時間でリフレッシュされる．

5-1-3 IPv6 に新しく追加された RR について

マルチホーム、リナンバリング(アドレスつけかえ)対応のために 1 つのアドレスを複数の RR に分割して登録する RR が追加された．

- 正引き: A6 RR (RFC 2874)
- 逆引き: ビットラベル (RFC 2673)
  DNAME RR (RFC 2672)
  上位ドメインは ip6.arpa. に変更される

しかし実際には採用しない方向と思われる．理由は,まず実装を鑑みた場合複雑すぎ、また導入コストに比べて効果が見えない． このため A6、ビットラベル、DNAME は"experimental"になる． ただし ip6.arpa.は導入の方向 (RFC 3152)となる．

BIND9 では A6、ビットラベル、DNAME を実装済みである．

5-1-4 ip6.arpa.への移行

前章で述べたように， IPv6 での逆引きの上位ドメインは in6.arpa に変更されるこ

とになる．このため，リゾルバ，サーバ両方とも移行措置が必要となった．具体的には以下の通り：

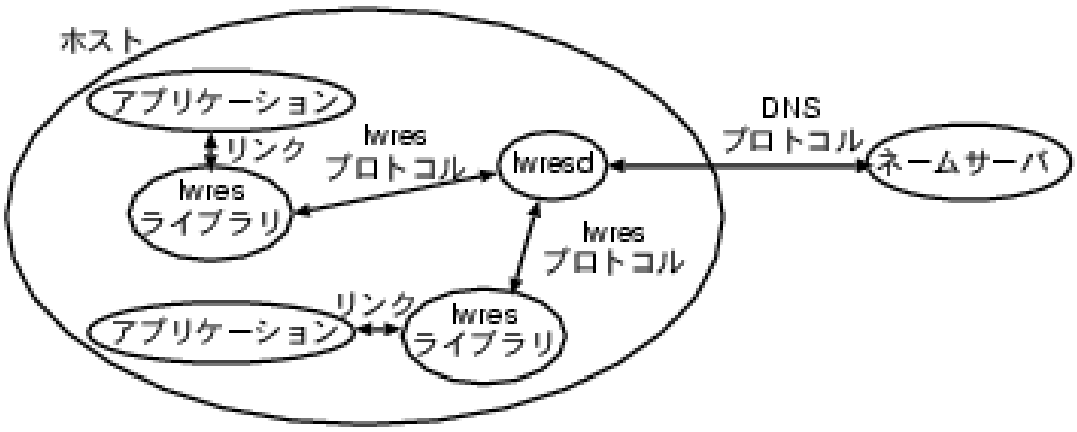
- リゾルバ側：ip6.arpa.と ip6.int.の両方を試すようにする．ただし実装はまだおこなわれていない．
- サーバ側：ip6.arpa.と ip6.int.の両方を管理する．

5-1-5 lwres：Light Weight Resolver

DNSSEC や A6 に代表される，DNS の新機能を基本ライブラリとして実装することには限界がある．BIND9 では，これに対応する方法として lwres ライブラリと lwres デーモン(lwresd)を用意している．

- lwres ライブラリ
  - アプリケーションへのインタフェース
  - インタフェースは既存の DNS 用ライブラリ関数に合わせる
  - ヘッダファイルだけ変えればソースレベルで互換
- lwres デーモン：
  - DNS による名前解決を担当
  - 実体は named：デーモンは"light weight"ではない
  - DNS の機能拡張をデーモンで吸収し、ライブラリを軽くする
  - ライブラリとデーモンは独自のプロトコル(UDP)で通信

lwres 環境のアーキテクチャを[図３]に示す．



［図３］

lwres の利用法は次のようになる．

1. lwresd の起動
   特別な設定は不要
2. /etc/resolv.conf を見る
   サーバの指定があればそのサーバに問い合わせ
   サーバの指定がなければ自力でルートサーバに問い合わせる
   IPv6 トランスポートもサポート

アプリケーション側

lwres ライブラリのヘッダファイルを include する

コードは変える必要なし
liblwres をリンクする

しかし lwres の機能を利用するためにはヘッダファイルのインクルードを追加し，liblwres を link してすべてのアプリケーションを再コンパイルしなければならないという問題がある.

5-2　モバイルサポートのための名前解決システム

5-2-1 モバイル向け名前解決システムの現状

名前解決システムのサーバ側では新機能の追加が進んでいるが、クライアント側では旧来の stub resolver を使用しているため、事実上、新機能が利用できない。しかし、stub resolver を新しいものに置き換えようとすると既存のアプリケーションソフトウェアの再コンパイルが必要になるため非常にコストが高くなってしまう。これを解決するために、アプリケーション・ソフトウェアには変更を加えずに、クライアント側にインテリジェントな汎用名前解決エンジンと名前解決の各機能を実装するモジュール群を用意することによってこの問題を解決する。

汎用名前解決エンジンは、クライアントホストにおいてローカルのネットワークデーモンとして動作し、クライアントに搭載されている stub resolver に対するネームサーバの設定をこの汎用名前解決エンジンに指定することで動作する。これは、一般の UNIX システムでは/etc/resolv.conf に書くネームサーバの名前を汎用名前解決エンジンに指定することで実現する。以下、具体的な名前解決方法として、(1) Link-Local Multicast によって名前解決を行う LLMNR 実装、(2) ICMPv6 を使用して名前解決を行う NI 実装を調査した。

5-2-2 mDNS(LLMNR)による名前解決

LLMNR 実装は、リンクローカルマルチキャストで名前解決を行う。

Sender は、host.example.com を解決したいときに LLMNR query を LINKLOCAL に port = 5353 で multicast し、responder は、FQDN を権威とするホストのみが UDP/IP ユニキャストを使用して応答する。

sender の受信時には、

- HopLimit/TTL = 255 でなかったら捨てる
- アドレスの正当性を検証
  - IPv4 -> IPV4LINK
  - IPv6 -> RFC2373

といった検証を行う。検証成功したら受理し、キャッシュする、失敗したら無視し、応答を待つ。

名前の衝突の解決は、ユニーク性を検証できたときのみ UNIQUE RR を使う。ここで、ホスト名のユニーク性の検証の実施時期としては、

1. 起動時
2. LLMNR 問合せに複数の I/F で応答すると設定されたとき
3. UNIQUE RR を追加するとき

などが考えられる。

解決手順としては、

- 登録ホスト
  - DNS dynamic update の手順によって要求を送信

- UNIQUE レコードを持つホスト
  - UNIQUE RR がないリクエストを受信
    - ✧ RFC2136.Section 3 に従って、YXRRSET をユニキャストで応答
- 登録ホスト
  - 送信 I/F から YXRRSET 応答を受け取ったときはその名前は使えない
  - それ以外は、LLMNR クエリと動的更新要求にその名前を UNIQUE RR として使える

その他、詳細は以下の通り。

◎制限事項

✧ 事前に分割されたセグメントがブリッジによって接続されたときは利用不可

✧ 送信者はこのクエリに応答する最初の応答者を除いた全ての応答者に受信した最初の応答を送信する

✧ そのホストが所有する UNIQUE RR を含む要求応答を受信するホストは、その要求を送らなかったとしても、LLMNR スコープにいる他のホストが同じ名前について権威が無いことを動的 LLMNR 更新要求機構を使用して検証しなければならない

◎マルチインタフェースへの対応条件

✧ 各 I/F でユニーク性を検証する

✧ 有効なインタフェースのみで LLMNR を使用

✧ 異なるリンク、異なる名前空間での名前の衝突の解決はしない -> out of scope

◎API

RFC2535 に従う

◎使用の制限

DNS が利用できないときの最後の手段

◎アクセス制御・認証

- リンクローカルの送信者からしか受け付けない
- TTL, HopLimit = 255 のときのみ
- 最終的には Layer2 でのアクセス制御が必要
- LLMNR と DNS のキャッシュを分割する
- 認証は既知共有鍵

5-2-3 ICMP Node Information による名前解決方式

Node Information Messages には、NI Query、NI Reply の二種類があり、いずれもフォーマットは同じで、ICMPv6 で運ばれる。フィールドの内容は以下の通り。

✧ Type

139 - NI Query

140 - NI Reply

✧ Qtype

16bit フィールド

情報の型を示す

要求から応答へコピーされる

✧ Flags

Qtype に依存

定義されていないときは 0

✧ Nonce

64bit の偽造防止のための数字

要求から応答へコピーされる

✧ Data

クエリ：Subject Address または Name

応答：Qtype に依存した値

Subject of Query が name のとき、

name は DNS wire format が必須

name は以下のどちらかであるかもしれない

長さ 0 で終了するラベルを含む FQDN

二個の長さ 0 のラベルに続く単一の DNS ラベル

query は最大一個の名前しかないので、DNS name compression は使用不可

送信

Node Information Query を ICMPv6 で送る

宛先は Queried Address（これは Subject Address, Subject Name とは異なる）

返信

Node Information Reply をユニキャストで送る

メッセージ処理は、以下のシーケンスで行われる。

(1) 質問者が NI Query を投げる

クエリの Subject が

(1-1) IPv6 アドレスなら

そのまま宛先アドレス

しかし、target node に関する情報を持っていれば宛先アドレスは不要

LINKLOCAL multicast アドレスにも NI Query を送信可能

RFC2373 に従う

(1-2) name

target address の情報を持っていなかったら link-scope multicast

正規化

RFC2535,sec8.1 に従う

英小文字かつ未圧縮

RR の最初の名前を MD5(RFC1321)でハッシュ

128bit ハッシュの先頭 32bit を使用

ff02:0:0:0:0:2::/96 の最後につける

-> NI Group Address と呼ぶ

(1-3) nonce

random / pseudo random

応答を受け取ったら必ず検査

必要なら ipsec を使う

(2) Responder は、宛先アドレスが以下なら受理

unicast

anycast

参加している link-local multicast

NI Group address

proxy service を提供している名前の NI Group address

デフォルトは proxy を有効

✧ Query の Subject Address または name がそのノードに属さないなら破棄

✧ 単一ホスト名はFQDNの最初の語に一致
名前はDNSSEC正規化によるクラス非依存
✧ Qtypeを知らないとき
ICMPv6 Code = 2, データなし, を返す
返答にはrate-limitを設定
どこのクエリを捨てるかをローカルポリシによって決定
✧ 応答を拒絶するとき
Refuse Code = 1, データなし
返答にはrate-limitを設定
✧ 受理するとき
flagを埋め、応答
address != anycast, multicast
source address = queried address
address == anycast, multicast
source address = queryを受け取ったアドレス
multicast/anycast addressに対するqueryへのreplyは
0 から MAX_ANYCAST_DELAY_TIME (RFC2461)の random interval 待つ

✧ Qtypeは
0 - NOOP
1 - サポートされたQtypes
2 - ノード名
3 - ノードアドレス
4 - IPv4アドレス
がある
✧ セキュリティ
デフォルト：global addressからの要求は拒否
NI応答量を制御できるようにする
リナンバリングが頻繁に起きるネットワークには不向き
オリジナルのDNSも同様

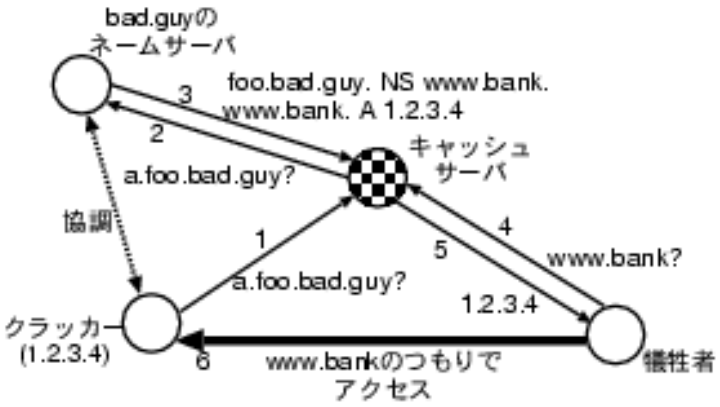
5-3 セキュリティやプライバシーを考慮した名前解決システム(DNS におけるセキュリティ)

DNSにおける主なセキュリティ上の課題には次のようなものが考えられる.
- サーバ・リゾルバ間、サーバ・サーバ間の通信(transaction)を認証する
  - 動的更新
  - ゾーン転送
- データベース検索結果の正当性
  - 偽者サーバへの誘導を防ぐ
    - キャッシュ汚染
    - 乗っ取られたネームサーバからの応答
    - 応答メッセージの改竄、なりすまし

5-3-1 キャッシュ汚染

キャッシュサーバに誤ったキャッシュを作らせることによって,犠牲者を誤ったアド

レスへ誘導する．古いBINDの実装に対しては簡単に実現できる．



［図４］

5-3-2 TSIG: transaction signatures

TSIG(transaction signatures)はRFC2845に定義されており，サーバ同士、またはサーバ・リゾルバ間の通信を認証に利用され，動的更新やゾーン転送などに用いられる．共通秘密鍵を用いてDNSメッセージ全体に署名する．

TSIG RR は メタRRであり，ゾーンファイルには現れないが， メッセージの最後に自動的に追加される． DNS メッセージ全体の一方向ハッシュ値を格納することにより署名を行なう．また， 繰り返し攻撃防止のために時刻を利用するため，利用者間での時刻同期が必要となる．

TSIGはBIND9で利用可能である．

5-3-3 SIG(0): Request and Transaction Signatures

SIG(0)：Request and Transaction Signatures は RFC 2931 に定義されており，動機・用途は基本的に TSIG と同じである．ただし， SIG RR を使った公開鍵方式であるところがTSIGとことなる点である．

SIG(0)はフォーマット・タイプなどはSIG RRと同じであり，やはりメタRRである．TSIGと同様，利用者間での時刻同期が必要である．

BIND9ではSIG(0)は実装されているが，まだ不完全である．BIND9では，SIG(0)を受信して検証はできるが、送信する手段はまだ実装されていない．

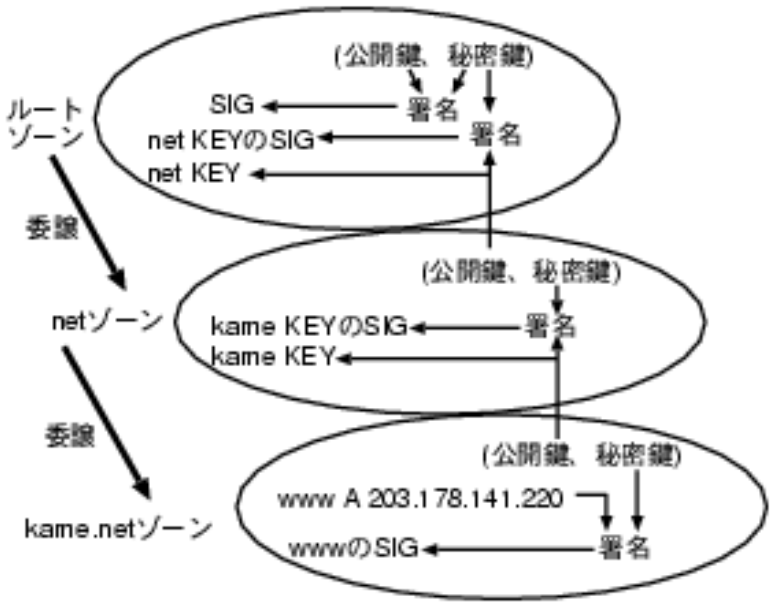
5-3-4 DNSSEC

DNSSECはRFC 2535で定義されており， DNSのデータベース全体の正当性を保証することを目的とする．これは 検索時のなりすまし、キャッシュ汚染からの防御に利用できる．また，"NXDOMAIN(名前が存在しない)"という応答の正当性も保証する．これはNXDOMAINを利用したDoSからの防御に使用できる．

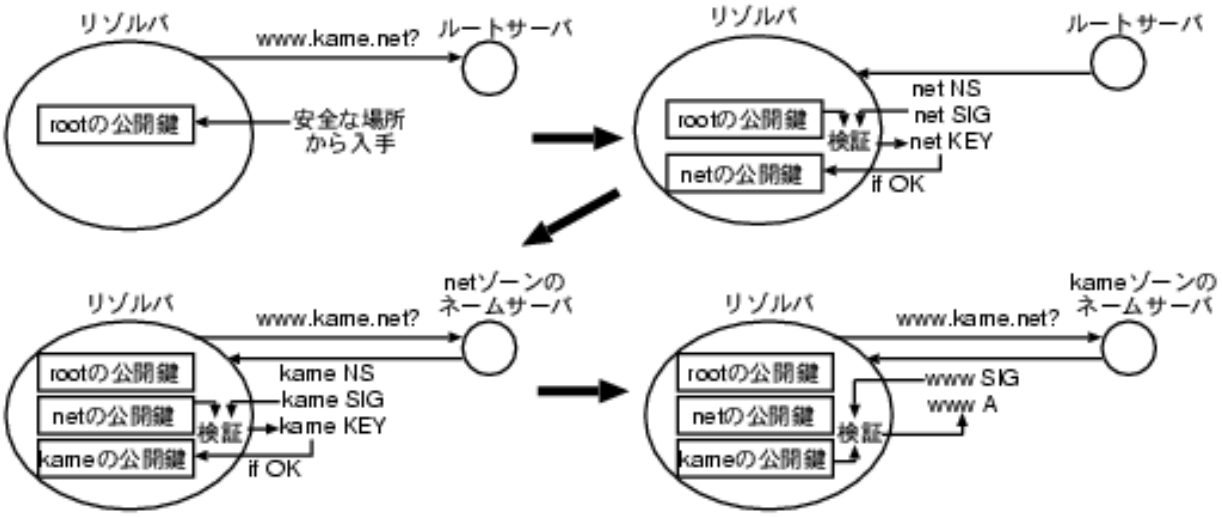
DNSSECは公開鍵方式による署名によって実現される.鍵はゾーンごとに管理され，公開鍵をKEY RRとして作成する．そして，前述のSIG RRを使用してゾーン内の各RRを秘密鍵で署名する．

署名の検証は， 応答のRRに対応するSIG RRを入手し，そのゾーンのKEY RRによって検証する．KEY RR自体の正当性は上位ゾーンの署名によるSIG RRで保証する．最終的にはルートゾーンによる署名に帰着する．ルートゾーンの公開鍵は既知であると仮定している．

DNSSEC の署名を[図５]に示す.　また，DNSSEC の署名検証を［図６]に示す.



[図５]



[図６]

5-3-5 KEY RR

DNSSEC で利用される KEY RR のデータは以下の情報から構成される.

- フラグ(2 バイト): DNSSEC ではふつう 256 = 認証と秘匿
- プロトコル(1 バイト), 3=DNSSEC, 4=IPSEC, etc.
- アルゴリズム(1 バイト): 1=RSA/MD5, 3=DSA, etc.
- 公開鍵: base 64 でエンコーディング

5-3-6 SIG RR

DNSSEC で利用される SIG RR のデータは主に以下のような情報から構成される.

- type covered: 署名対象の RR タイプ
- アルゴリズム
- オリジナル TTL: ゾーンファイルに登録された RR の TTL

- 署名の有効期間(はじまりと終り)
- 鍵のタグ: 複数の鍵を同時に使用している場合の識別子
- 署名を付けたゾーン名（ふつうは RR の属するゾーン、KEY RR に対する SIG RR の場合は上位ゾーン）
- 署名: base 64 でエンコーディング

5-3-7 NXT RR

NXT RR は, "NXDOMAIN"の正当性を保証するために利用される. 単純に"NX"を署名した場合は再利用される可能性があるため,これではセキュリティを保てない.そのため,「前後」の名前を示して、その中間がないことを保証するという方式をとる. ただし,これには RR の正規化と順序付けが必要である. NXT RR のデータは,「次の名前」と「その RR のタイプのリスト」から構成される. 応答データの構成は, コード"NXDOMAIN"と一緒に NXT RR が返ることになる. このとき, NXT RR の key は「一つ前」の RR であり, next name は「一つ後」の RR を示している. NXT RR は、その中間には何もデータがないことを主張しており, NXT RR 自体の正当性は SIG RR で保証する. ただしこの方法の問題点はあるゾーン内の全 RR が芋蔓式に漏れることになる.

5-3-8 署名の手順

以下に署名の手順を示す.

○初回の手順

1. ゾーン用の鍵を生成する
2. 公開鍵(KEY RR)を上位ゾーンに署名してもらう
   KEY RR に対する SIG RR を得る
3. 秘密鍵でゾーンの各 RR を署名する
   それぞれ対応する SIG RR が生成される
   各 RR に対応する NXT RR と、それに対する SIG RR も生成される

○更新手順

4. 新しい鍵を生成する

5-1. 新旧両方の KEY RR を上位ゾーンに署名してもらう

5-2. 新しい KEY RR だけを上位ゾーンに署名してもらう

6. 新しい秘密鍵でゾーンの各 RR を署名する。古い KEY RR は残す
   (古い鍵で署名した SIG RR の有効期限が切れる)
7. 古い KEY RR を削除する

5-4 総括

以上のように、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た。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た。

参考資料、参考文献

[1] S. Thomson, C. Huitema., DNS Extensions to support IP version 6 (RFC 1886), December 1995.
[2] D. Eastlake., Domain Name System Security Extensions (RFC 2535), March 1999.
[3] P. Vixie., Extension Mechanisms for DNS (EDNS0) (RFC 2671), August 1999.
[4] M. Crawford., Non-Terminal DNS Name Redirection (RFC 2672), August 1999.
[5] M. Crawford., Binary Labels in the Domain Name System (RFC 2673), August 1999.
[6] P. Vixie, O. Gudmundsson, D. Eastlake, B. Wellington., Secret Key Transaction Authentication for DNS (TSIG) (RFC 2845), May 2000.
[7] M. Crawford, C. Huitema., DNS Extensions to Support IPv6 Address Aggregation and Renumbering (RFC 2874), July 2000.
[8] D. Eastlake., DNS Request and Transaction Signatures (SIG(0)s) (RFC 2931), September 2000.
[9] R. Bush., Delegation of IP6.ARPA (RFC 3152), August 2001.
[10] R. Arends , M. Larson , D. Massey, S. Rose , DNS Security Introduction and Requirements draft-ietf-dnsext-dnssec-intro-01.txt, Work in progress
[11] R. Arends , M. Kosters, D. Blacka, DNSSEC Opt-In, draft-ietf-dnsext-dnssec-opt-in-01.txt, Work in progress
[12] R. Arends, M. Larson, D. Massey, S. Rose, Resource Records for DNS Security Extensions, draft-ietf-dnsext-dnssec-records-00.txt, Work in progress
[13] S. Rose , DNS Security Document Roadmap draft-ietf-dnsext-dnssec-roadmap-05.txt, Work in progress
[14] Olafur Gudmundsson, Delegation Signer Resource Record , draft-ietf-dnsext-delegation-signer-07.txt, Work in progress
[15] P. Vixie, S. Thomson, Y. Rekhter, J. Bound, Dynamic Updates in the Domain Name System (DNS UPDATE) (RFC2136), April 1997.

（添付資料）

1　研究発表、講演、文献等一覧　　（←研究開発実績報告書を参考に記載）

今年度は、動向調査が主業務のため、該当なし。