特別研究報告

題目

Unified Multiplex 通信アーキテクチャにおける アドレス管理機能の設計

指導教員
村田 正幸 教授

報告者
西田 和生

平成 21 年 2 月 16 日

大阪大学 基礎工学部 情報科学科

平成 20 年度 特別研究報告

Unified Multiplex 通信アーキテクチャにおける アドレス管理機能の設計

西田 和生

内容梗概

我々の研究グループでは IPv6 の広いアドレス空間を活用し，既存のアプリケーションを変更せず，クライアントへの匿名性の提供やサーバの待ち受けアドレスの保護などの高い安全性を提供できる Unified Multiplex 通信アーキテクチャを提唱している．本アーキテクチャはアドレスの利用方法に大きな特徴がある．クライアント側では，セッションを確立する度にそのセッション固有のアドレス “Ephemeral Address” を新たに割り当てて使用する．またサーバ側では，接続を要求するクライアントごとに異なる専用のアドレス “Specific Service Address” によってサービスを提供する．これらの新しいタイプのアドレスはセッションが確立されている期間にのみ有効である．また，サービスごとにアドレスが動的に変化するなど， IP アドレスをインターフェースごとに静的に割り当てる既存の通信方式にはない独自の性質を有する．

本報告ではこれらのアドレス特性について，その生成から，割り当て，削除までの流れを厳密に整理・分析する．また，サービス専用アドレスをアプリケーションの通信にあわせてオンデマンドで提供するためには，アドレス管理機能が必要となる．そこで，アドレスの生成タイミング，生成方法，選択方法などアドレス管理に必要となる機能について詳細設計を行い，その実装方式を示す．さらに，Unified Multiplex 通信アーキテクチャの安全性を示すため，Unified Multiplex 通信ノードが Legacy 通信ノードに悪影響を与えないことを，オペレーティングシステムごとに実験し検証する．

主な用語

IPv6, Unified Multiplex, ephemeral address, specific service address, アドレス管理

目 次

図 目 次

表 目 次

1 はじめに

我々の研究グループでは IPv6 の広いアドレス空間を活用し，既存のアプリケーションの変更を要することなく，高い安全性などを提供できる Unified Multiplex 通信アーキテクチャ (以下， Unified Multiplex 通信) を提唱している [1]．Unified Multiplex 通信は，従来の通信アーキテクチャ (以下， Legacy 通信) においてアドレスとポート番号によって行われていたセッション多重化の考えを刷新し，セッション開始時に異なるアドレスを動的に割り当て，セッションの終了時にアドレスの削除を行うことにより， IP アドレスのみを用いてセッションの多重化を実現する．これにより，Unified Multiplex 通信はサービス専用アドレスを実現する．

サービス専用アドレスとして，クライアントが送信元アドレスとして割り当てる Ephemeral Address (以下，EA) [2]，サーバが待ち受けアドレスとして割り当てる Specific Service Address (以下，SSA) を新しく導入する．このようなアドレスはアプリケーションの通信要求に応じた動的な生成，割り当て，通信終了後のアドレスの削除など， Legacy 通信 にはなかった新しい特性がある．

これらの特性により，クライアントで用いる送信元アドレスが常に一定ではなくなるため，他者に IP アドレスからノードを対応づけることが不可能であるという，ロケーションプライバシの保護が可能となる．またサーバでは，待ち受けのためのサービス専用アドレスが分からない第三者からのアクセスは，現実時間内にアドレスを特定し接続することが不可能になるため，特別なセキュリティ処理を施すことなく高い安全性を持った通信が可能となる．

このように Unified Multiplex 通信では，従来にないサービス専用アドレスという概念を導入しているが，サービス専用アドレスを実現するためには，アドレスをいつ生成し，どのアドレスを選択するか，またいつ割り当て，削除を行うかを管理する必要がある．

本報告ではこれらのアドレス特性について，その生成から，割り当て，削除までの流れを厳密に整理，分析する．また，サービス専用アドレスをアプリケーションの通信にあわせてオンデマンドで提供するためには，アドレス管理機能が必要となる．そこで，アドレスの生成タイミング，生成方法，選択方法などアドレス管理に必要となる機能について詳細設計を行い，その実装方式を示す．さらに，Unified Multiplex 通信アーキテクチャの安全性を示すため，Unified Multiplex 通信ノードが Legacy 通信ノードに悪影響を与えないことを，オペレーティングシステムごとに実験し検証する．

以下， 2 章で Unified Multiplex 通信の概要を示す．次に 3 章でサービス専用アドレスの特性を分析し，その分析を元に 4 章でサービス専用アドレスを実現するための必要な機能について述べ， 5 章でアドレス管理機能の未実装機能の設計を示す． 6 章でアドレス管理機能の元になる機能の Uncertain 状態の実験を示し，最後に 7 章でまとめと今後の課題につい

て述べる.

2 Unified Multiplex 通信の概要

従来の IPv4 を中心とする Legacy 通信アーキテクチャでは， IP アドレスは一つのノードに対して一つの固定アドレスを設定するのが一般的である．また，IPv4 はアドレス空間が限定されるため，1 ノードに対して多数のアドレスを設定する方法を採用することは事実上不可能となっている．しかし，このような従来の IP アドレスでは，以下に述べる問題がある．すなわち，ノードのアドレス情報が第三者に伝われば，プライバシ情報が露呈する危険性があり，また，その情報を元に第三者からの攻撃を受ける要因になる．インターネットの普及や利用方法の進化により，このようなセキュリティに関する問題意識は急速に高まってきている．

我々はこの問題に対して， IPv6 のアドレス空間の広さを通信セキュリティの向上に活用するために，1 ノードに対して複数の動的に変化するアドレスを実現する Unified Multiplex 通信アーキテクチャを提案している．

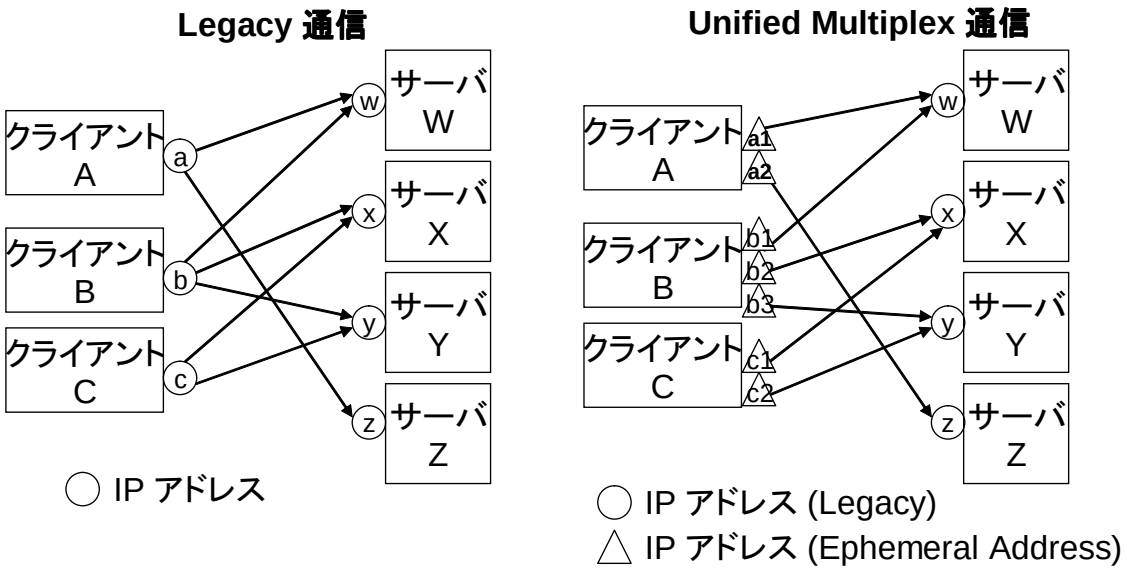
128 bit のアドレス空間を有する IPv6 の仕様では，ユーザの利用するエンドノードにおいて，下位 64 bit 分はユーザが自由に設定，利用することが可能である．2^{64} ($\fallingdotseq 1.8 \times 10^{19}$) という空間は，一秒に一つのアドレスを探索した場合，述べ一千億年かけても全数の探索できない程度の広さであり，現実的には探索不可能な広さであると考えることができる．その結果，単にランダムな IP アドレスを動的に割り当てるだけで，非常に高いセキュリティを提供できる可能性がある．

Legacy 通信と Unified Multiplex 通信では以下のように大きな違いが存在する．Legacy 通信では， IP アドレスはノードのインタフェースに固定で一つ割り当てられており，セッションの多重化を行う際には動的に変化するポート番号をあわせて用いる．これに対して Unified Multiplex 通信では，インターフェースに対して多数のアドレスを割り当て，それぞれをセッションに対応づけすることによって，セッションの多重化を IP アドレスのみで行う．

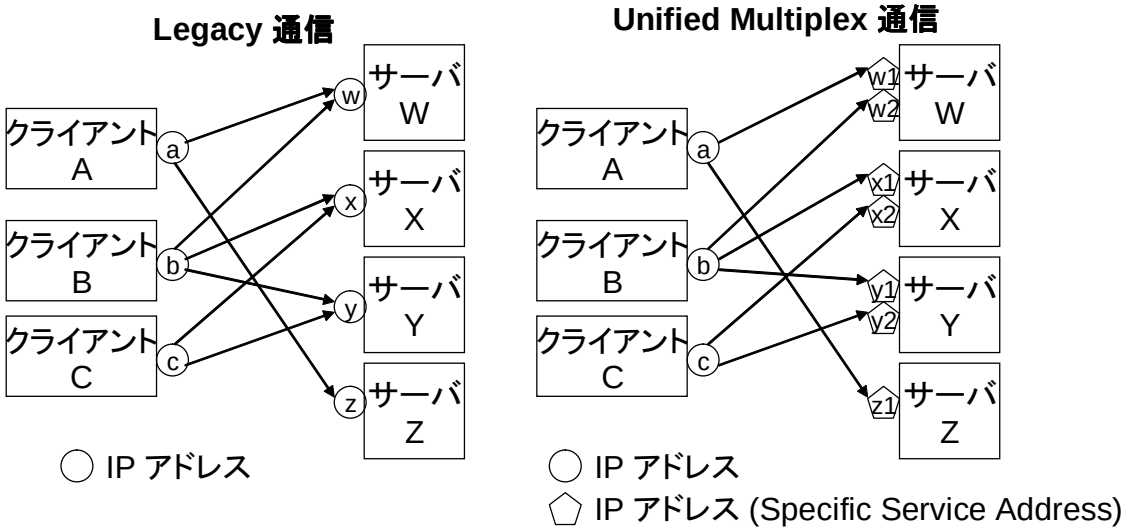
Unified Multiplex 通信において，多重化を行うアドレスは クライアントで送信元アドレスとして使用される EA, サーバで待ち受けアドレスとして使用される SSA の二つのサービス専用アドレスに分類される．Ephemeral Address (EA) は，Legacy 通信において送信元ポート番号として通信開始時に動的に割り当てを行うポート番号である Ephemeral Port の考え方を，IP アドレスに適用したものである．EA はセッション開始時に動的に割り当てられ，セッション終了時に削除される．一方， Specific Service Address (SSA) はサーバがある特定のクライアントからの接続を待ち受けるために割り当てられるアドレスであり，サービスの利用ごとに一度きりの待ち受けを実現するものである．

図 1 に Unified Multiplex 通信におけるサービス専用アドレスを示す．Legacy 通信ではク

ライアント A がアドレス a を用い，サーバ W が アドレス w で待ち受けを行っているものに対して接続を行う．このとき，クライアント A がサーバ Z に対して接続を行う際にもサーバ W に対して用いたアドレスであるアドレス a を送信元アドレスとし，接続を行う．また，サーバ W においても， クライアント A からの接続に使用したアドレス w をクライアント B からの接続の待ち受けるに使用する．これに対し， Unified Multiplex 通信では Ephemeral Address を使用するため クライアント A はサーバ W に対して接続を行う際に， アドレス a1 を用いて接続を行い，サーバ Z に対しては アドレス a2 を用いる．さらに，クライアント A が使用したアドレスは毎回異なるため， サーバ W がアドレス a2 に対して，サーバ Z がアドレス a1 に対して，それぞれ接続できない．サーバ側においても Specific Service Address を使用し，サーバ W はクライアント A に対してサービスを提供する際には アドレス w1 を用い，クライアント B に対しては アドレス w2 を用いてサービスを提供する．このことにより，クライアント A はアドレス w2 に対して，クライアント B はアドレス w1 に対して接続を行うことは不可能になる．以上のように， Unified Multiplex 通信アーキテクチャでは，サーバおよびクライアントの両方においてサービス専用アドレスを提供できるが，このようなサービス専用アドレスは，従来のネットワークインタフェースごとに割り当てられていた IP アドレスの概念を根本的に考え直す必要がある．そこで次章では Unified Multiplex 通信におけるサービス専用アドレスが，どのような状態遷移を経るのかについて，詳細な分析を行う．



(a) Ephemeral Address



(b) Specific Service Address

図 1: Unified Multiplex 通信アーキテクチャにおけるサービス専用アドレス

3　サービス専用アドレスの特性分析

従来，IP アドレスはノードごとに割り当てられ，ノード識別子としてアドレスが使用される．一方，Unified Multiplex 通信アーキテクチャでは IP アドレスをサービスごとに割り当てることによってサービス専用アドレスを実現する．以下，サービス専用アドレスと従来のノード識別子としてのアドレス （以下，ノード識別子アドレス） を用いた通信手順を比較することによって，サービス専用アドレスがいつ有効化され，削除されるかなどの特性を分析する．また，サービス専用アドレスをアプリケーションの通信要求に対してオンデマンドで提供するためには，アドレス管理を自動的に行う必要がある．このため，アドレス管理機能にどのような機能が必要になるのかを述べる．

まず，図 2 にサービス専用アドレスの生成から消滅までの流れを示す．従来の Legacy 通信方式で用いられているノード識別子としての IP アドレスと比較すると，サービス専用アドレスの有効時間は非常に短い．また，図 3 にサービス専用アドレスを用いた通信の概要を示す．具体的な手順は以下の 15 ステップで列挙できる．

(1) サーバは，セッションごとに動的に変化するアドレスを生成する．

(2) サーバは，生成したアドレスがすでに他のノードに使用されていないかを確認する．

(3) サーバは待ち受けを行うサービスごとに割り当てるべきアドレスを決定する．アドレスは (2) までで生成されたアドレスの中から選択する．

(4) サーバはサービスの待ち受けを行う．

(5) クライアントは接続したいサーバのアドレスを取得する．

(6) サーバはクライアントに対して待ち受けアドレスを通知する．

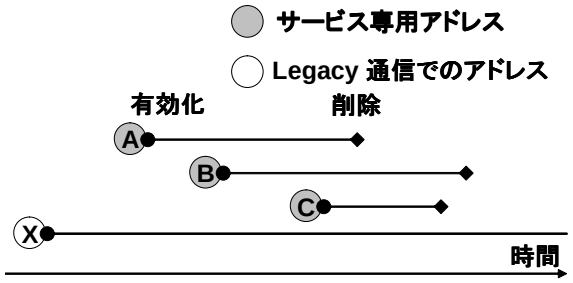


図 2: サービス専用アドレスと Legacy 通信のアドレスの有効時間

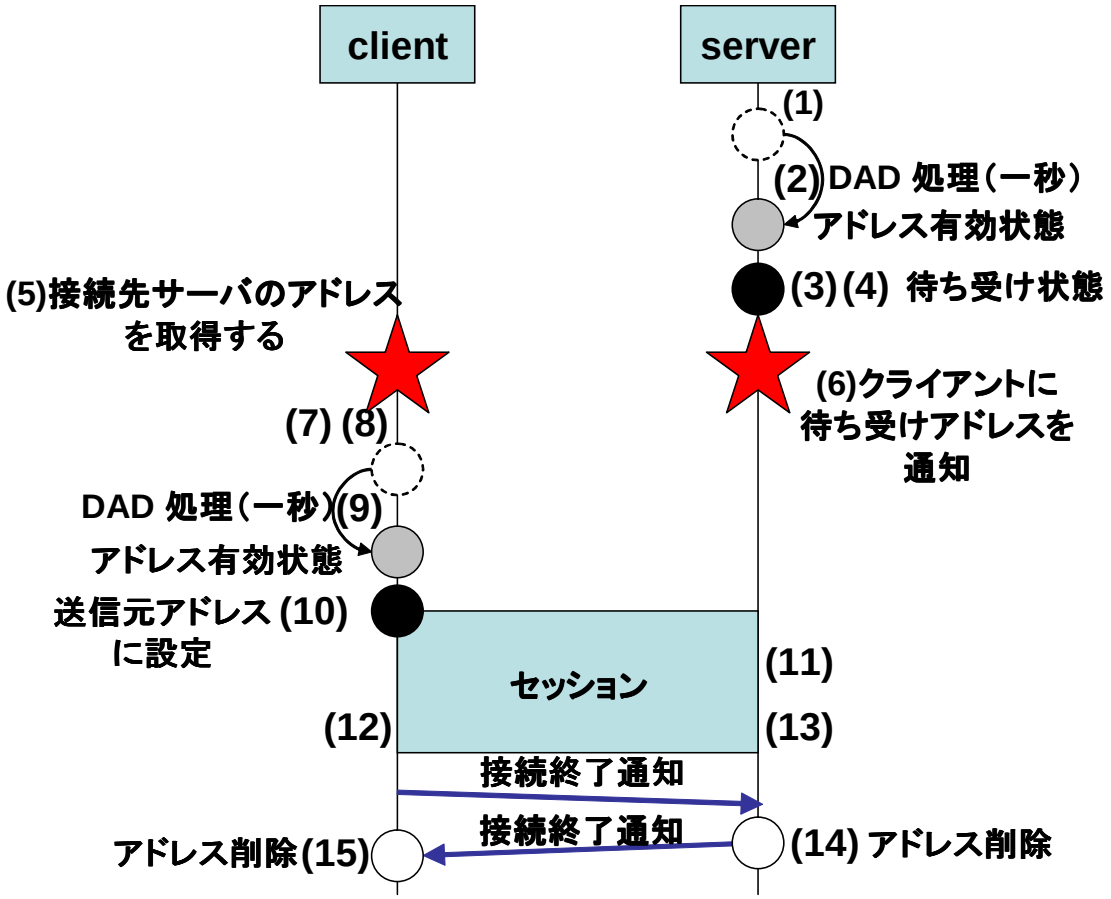


図 3: サービス専用アドレスを用いた通信の概要

(7) クライアントはセッションごとに動的に変化する送信元アドレスを生成する.

(8) クライアントはサーバに対してどのアドレスを使用して接続するかを決定する.

(9) クライアントはそのアドレスが他のノードに使用されていないかを確認する.

(10) クライアントはサーバに対して接続を行う前に，通信セッションに対してその通信の送信元アドレスとして設定する，その後通信されたサービス専用アドレスを用いて接続する.

(11) サーバはクライアントから接続を受け付けたアドレスでは他のクライアントからの接続を受けない.

(12) クライアントは通信を行っている間，送信元および宛先アドレスによって他の通信セッションとの識別を行う.

表 1: クライアントとサーバの共通項目に関する Legacy 通信と Unified Multiplex 通信の違い

	Legacy	Unified
(5)(6) サーバの待ち受けアドレス通知	すべてのクライアントに対して同じアドレスを通知	クライアントごとに別々のアドレスを通知
(2)(9) アドレスがすぐに利用可能か	アドレスはすでに有効状態なのですぐに利用可能	DAD が必要
(12)(13) セッション多重化に必要な情報	送受信アドレスおよび送受信ポート番号	送受信アドレス
(14)(15) アドレスを削除するか	アドレスの削除を行わない	アドレスの削除を行う

(13) サーバは通信を行っている間，送信元および宛先アドレスによって他の通信セッションとの識別を行う．

(14) サーバはセッションを終了後，使用していたアドレスの削除を行う．

(15) クライアントはセッションを終了する際に使用していたアドレスの削除を行う．

表 1 にクライアントおよびサーバの共通項目，また表 2 および表 3 で，クライアントおよびサーバにおいて Legacy 通信と Unified Multiplex 通信の各手順における違いをそれぞれ示す．これらの表より，サービス専用アドレスを実現する Unified Multiplex 通信と Legacy 通信の違いを分析し，サービス専用アドレスの特性を示す．

表 2: クライアント側における Legacy 通信のアドレスと Unified Multiplex 通信の違い

	Legacy	Unified
(7) 送信元アドレスの変化	常に同じアドレス	セッションごとに動的に変化
(8) 送信元アドレス選択時の比較範囲	クライアントに設定されているすべてのアドレスを比較	クライアントに設定されている prefix を比較
(10) 送信元アドレスの割り当て	新たにクライアントに割り当てない	割り当てるアドレスは適宜生成する

表 3: サーバ側における Legacy 通信のアドレスと Unified Multiplex 通信の違い

	Legacy	Unified
(1) 待ち受けアドレスの変化	常に同じアドレス	接続を待ち受けるクライアントごとに動的に変化
(3) 待ち受けアドレスとプロセスの関係	すべてのプロセスは同じアドレスで待ち受け	プロセスごとに違うアドレスで待ち受け
(4) 待ち受けアドレスの割り当て	新たにサーバに割り当てない	割り当てるアドレスは適宜生成する
(11) 待ち受け可能クライアント数	アプリケーションによる	1 クライアントのみ

このとき，Legacy 通信と異なり，機能の設計が必要になる箇所を共通項目，クライアント，サーバに分類を行い記述する．

共通項目

(5)(6) サーバの待ち受けアドレス通知機能

サーバの待ち受けアドレスをクライアントにどのようにして通知を行うか．サーバの IP アドレスを取得する方法として DNS (Domain Name Service) が存在するが，Unified Multiplex 通信ではクライアントごとに別々の異なるアドレスを通知する．

(2)(9) アドレス即時利用可能機能

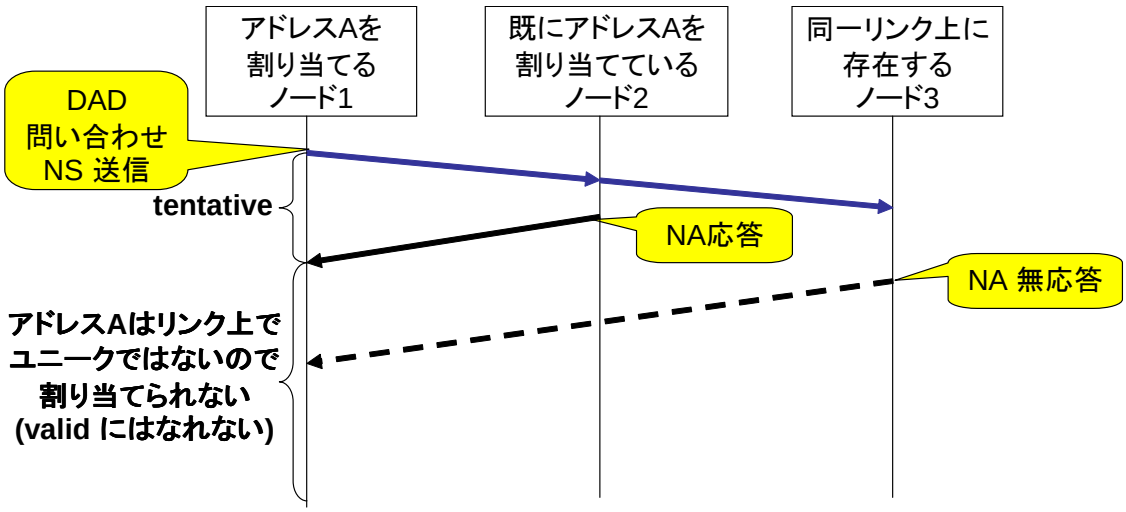
クライアント，サーバともにアドレスを割り当てる際にアドレスが他のノードで使用されていないことの確認が必要になる．この確認には [3] に定められている DAD (Duplication Address Detection) を用いる．図 4 に DAD の概要を示す．アドレス A を割り当てたいノード 1 が ICMP (Internet Control Message Protocol) による NS (Neighbor Solicitation) を用いて，アドレス A が他のノードに割り当てられていないかを確認する．この確認を行っている期間を tentative 状態という．確認が行われた際に，既にアドレス A の割り当てを行っているノード 2 は アドレス A を割り当てられていることをノード 1 に対して NA (Neighbor Advertisement) を用いて，応答する．NA を受けたノード 1 は重複していることを確認したのでアドレス A を割り当てることはできない．また，アドレス重複を通知する NA の応答がない，アドレスが重複していないを意味し，アドレスを割り当てることができる．このとき，アドレス A と通信を試みるノード 3 が通信を行うため， L2 アドレス（つまり，MAC アドレス）をノード 1 に対し， NS を用いて問い合わせる．ノード 1 には L2 アドレスを NA を用いて通知を行う．DAD の応答確認待ち時間に関しては，ネットワーク上には性能の低いノードや様々な方法で DAD のプロトコルを実装したノードが存在することが想定され， 一秒を要するとされている．この DAD の応答確認待ち時間のため，アドレスは即時利用が不可能になるので，アプリケーションの要求に対してオンデマンドに対応するためには工夫が必要になる．

(12)(13) セッション多重化機能

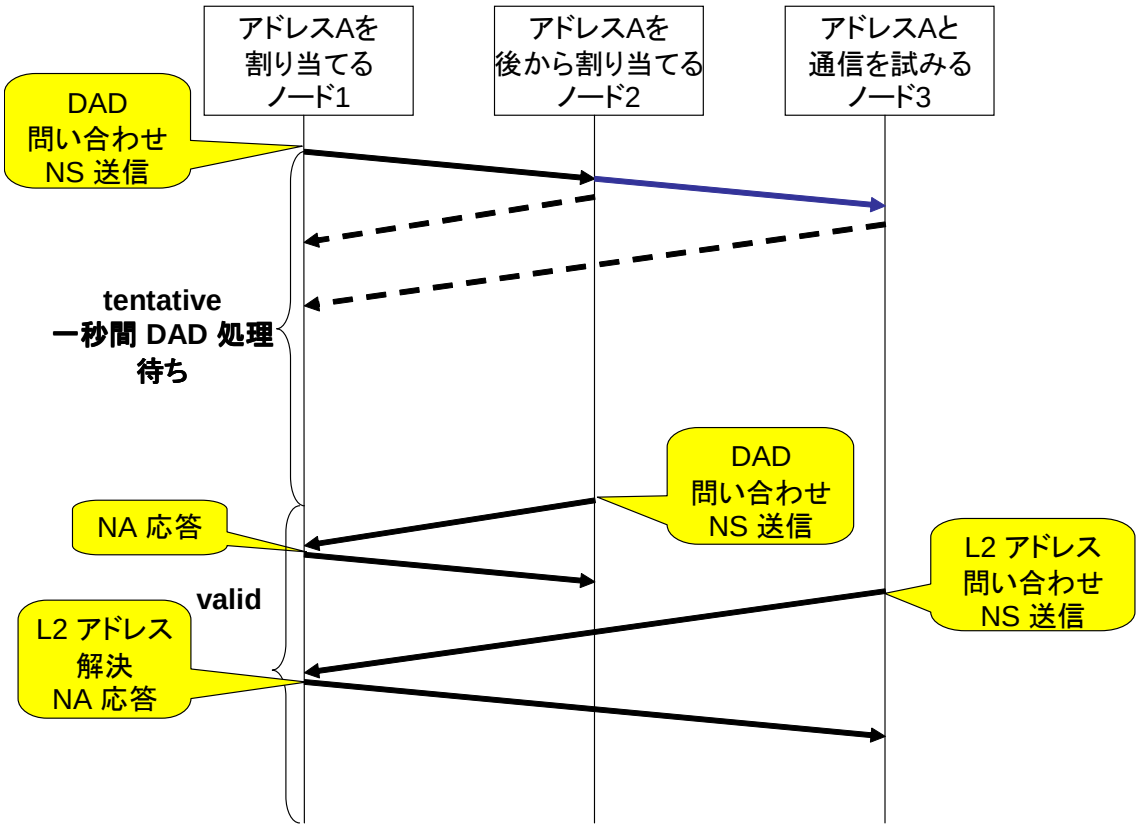
クライアント，サーバが通信を行っている間，クライアント，サーバそれぞれがセッションの識別をどのように行うか．

(14)(15) アドレス削除機能

クライアント，サーバはどのタイミングで使用していたアドレスの削除を行うか．



(a) DAD を行い重複を検知しアドレスの割り当てができない場合



(b) DAD を行い重複がないことを一秒確認しアドレス割り当てができる場合

図 4: DAD (Duplication Address Detection) 処理の概要

クライアント

(7) 変化する送信元アドレスの生成

Unified Multiplex 通信では，送信元アドレスが接続を行うたびに変化する．

(8) 送信元アドレス選択機能

Legacy 通信では，送信元アドレスの選択はマルチホーム環境などごく限られた場合のみ必要であり，またその選択候補も非常に少量である．これに対して，Unified Multiplex 通信ではサービスへの割り当てアドレスは，あらかじめプールされたアドレスの集合から選択することから，多数のアドレスから最適なアドレスを取捨選択するための方式が必要となる．

(10) 送信元アドレス割り当て機能

Legacy 通信ではアドレスは常に有効状態で，送信元アドレスに割り当てを行う際にアドレスに対して別途処理を行う必要はない．これに対し， Unified Multiplex 通信ではノードの稼働中にアドレスの生成から削除までをオンデマンドで処理する必要があることから，アドレスを割り当てる際に別途有効化処理が必要である．

サーバ

(1) 変化する待ち受けアドレスの生成

Unified Multiplex 通信では，送信元アドレスが接続を行うたびに変化する．

(3) 待ち受けプロセス選択機能

サーバがどのプロセスのソケットで待ち受けを行うかを選択する機能．サーバがどのサービスを提供するかによる．

(4) 待ち受けアドレス割り当て機能

(11) 単独クライアント待ち受け機能

サーバはクライアントから接続を受けたアドレスで他のクライアントからの接続を待ち受けないためにはどうするか．

4　サービス専用アドレスを実現するアドレス管理機能

本章では 3 章で挙げたサービス専用アドレスを実現するために必要な諸機能の設計方針について述べる．これらの機能は，共通機能，サーバ機能，およびクライアント機能にグループ化できる．以下では各グループごとに必要な機能の詳細について述べる．

4.1　アドレス管理に必要となる機能

共通項目

(5)(6) サーバの待ち受けアドレス通知機能

Legacy 通信のようにサーバのアドレスが固定である場合は，DNS [4] を使用し，アドレスの取得が可能になる．しかし，サービス専用アドレスを用いる通信では，クライアントのセッションごとに異なるアドレスを提供しなければならない．このような使用方法では DNS での対応は困難である．このため DNSO (DNS Name Space Override) によってこれを解決している． DNSO では，クライアントがサーバのアドレスを取得のための DNS クエリをフックし，その DNS クエリの情報を元に NOCA (Name Override Client Agent)， ブローカー，NOSA (Name Override Server Agent) を連携させ，サーバに対して待ち受けアドレスを要求する機構である．この機構を利用することによって，クライアントは宛先アドレスの取得を行うことが可能になる．

(2)(9) アドレス即時利用可能機能

Unified Multiplex 通信を実装する環境である IPv6 では ノードがアドレスを割り当て，使用する際には DAD 処理が必要になる．その DAD 処理には一秒要することになり，ユーザがセッションを開始する際に新たなアドレスをすぐに使用するにはセッションを開始する一秒前にアドレスを生成する必要がある．しかし，一秒間に複数のセッションを開始する場合の対応はできず，しかもセッションの開始タイミングを予測することは難しい．そこで，あらかじめ複数のアドレスを割り当てておかなければならない．しかし，そのアドレスは有効状態ではなく，セッション開始時に初めて有効状態になるアドレスである必要がある．以上の要求を満たすアドレスの割り当て方法としてすでに DAD 処理済みであるが，有効状態ではないアドレスである Uncertain 状態 [5] のアドレスを保持しておくアドレスプール機能が必要になる．また，アドレスプール機能には，アドレスプールに追加アドレスの値の生成機能，アドレスプールはどれだけのアドレスを保持しておく必要があるのかを管理するアドレス数に関する管理機能を追加で考える必要がある．

(12)(13) セッション多重化機能

セッションの識別のため，クライアントでは送信元アドレスおよび宛先アドレスの組み合わせで行い，サーバでは，待ち受けアドレス，クライアントの送信元アドレスの組み合わせによって多重化を行う．

(14)(15) アドレス削除機能

セッション終了時に使用していたアドレスの削除を行う．正常にセッションが終了しなかった際にも確実にアドレスの削除を行うために PCB (Protocol Control Block) [6] が削除されるタイミングでその PCB に関連付けを行っているアドレスの削除もあわせて行う必要がある．

クライアント

(7) 変化する送信元アドレスの生成

クライアントはセッションごとに違うアドレスを生成する必要がある．この機能はアドレスの即時利用可能機能の中で動的なアドレスの生成を行うことによって実現される．

(8) 送信元アドレス選択機能

クライアントの送信元アドレスの選択を行う場合， Unified Multiplex 通信ではアドレスプールの中に存在する多数のアドレスより選択を行う必要がある．このとき，全アドレスの全数検索を行うことによって，通信ごとに多くの計算コストがかかり，効率的ではない． Unified Multiplex 通信では送信元アドレスの選択に要する検索は必要な検索のみにとどめる必要がある．必要な検索とは，宛先アドレスの prefix [7] に一番近い prefix を持つ Uncertain 状態のアドレスであるということだけである．

(10) 送信元アドレス割り当て機能

アドレスプールから適切なアドレスを選択し，クライアントの送信元アドレスとして割り当てを行う．このとき，選択した Uncertain 状態のアドレスは通信に使用することができない状態なので有効化処理が必要になる．Uncertain 状態の実装はカーネルで行うのでこの有効化処理もカーネルで実装される．アドレスを送信元アドレスに設定を行うタイミングはクライアントが connect() のシステムコールを呼んだときである．

サーバ

(1) 変化する待ち受けアドレスの生成

クライアントの変化する送信元アドレスの生成と同じく，アドレスの即時利用可能機能を用いれば実現できる．

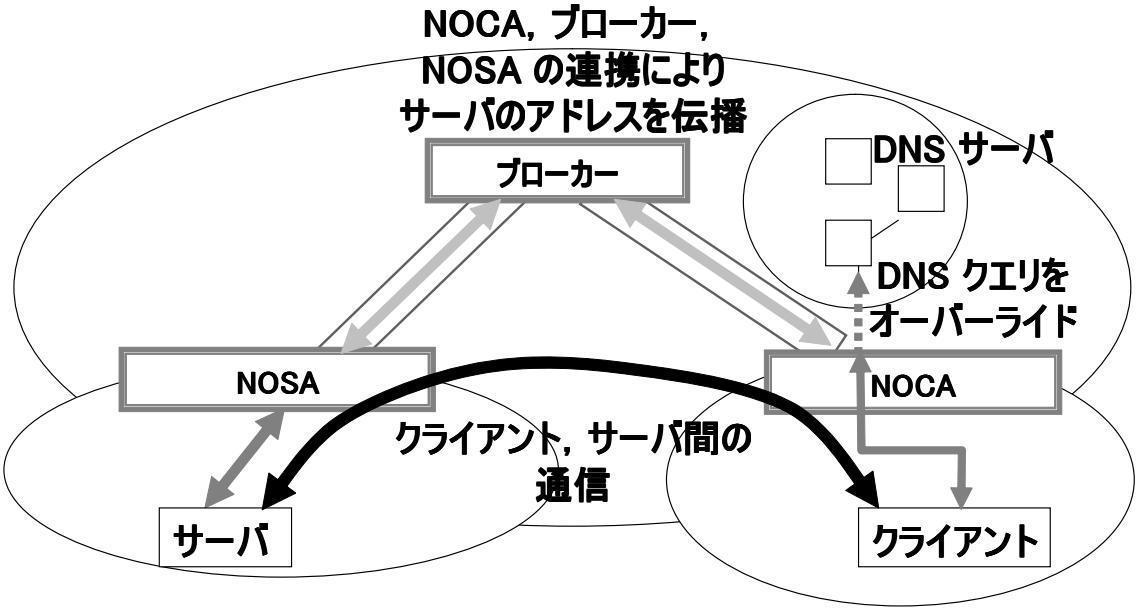


図 5: DNSO (DNS Name Space Override) の概要

(3)(4) 待ち受けアドレス割り当て機能

サーバの待ち受けアドレスは DNSO の仕組みを利用し，17 ページで述べた，NOSA から指定される．このとき，どのプロセスで待ち受けるかは FQDN (Fully Qualified Domain Name) によって指定される．図 5 に DNSO の動作を示す．サーバは NOSA からアドレスを受け取り，そのアドレスで待ち受けを行う．このとき，NOSA でそのアドレスの DAD 処理を行ってない場合は サーバで別途 DAD 処理を行い，そののち待ち受けアドレスとして割り当てる必要がある．DAD 処理が完了済みであればすぐにアドレスを割り当てることが可能になる．

(11) 単独クライアント待ち受け機能

サーバはクライアントからの接続を待ち受け，実際に通信をする際に使用するソケットは一度きりの接続のみを行う．接続を待ち受けるソケットが子ソケットを作ることなく，待ち受けを行ったアドレスの通信を行うことで実現可能である．

4.2 各機能の実装状況

以上の機能それぞれについて，現在どの範囲が実装され，それがどのように実装されているのかを確認するためにアドレス管理機能の実装対応表を整理，作成する．この表により，

実装がまだ不足している範囲の設計を行う．表 4 にどのように設計，実装がなされているかを示す．

表 4: アドレス管理機能設計と実装状況

機能	設計，実装状況
(5)(6) サーバーの待ち受けアドレス通知機能	DNSO により設計，実装 [8]
(2)(9) アドレスの即時利用可能機構	アドレスプールの設計は行われていない
(12)(13) セッション多重化機能	カーネル内の PCB の変更により設計，実装 [9]
(14)(15) アドレス削除機能	カーネル内のアドレス情報にフラグを設定することにより設計，実装 [9]
(7) 変化する送信元アドレスの生成	アドレス生成機能の設計は行われていない
(8) 送信元アドレス選択機能	まだ設計は行われていない
(10) 送信元アドレス割り当て機能	カーネルの機能として設計，実装 [9]
(1) 変化する待ち受けアドレスの生成	DNSO により設計，実装 [8]
(3)(4) 待ち受けアドレス選択機能	DNSO により設計，実装 [8]
(11) 単独クライアント待ち受け機能	LISTEN ソケット変更により設計，実装 [9]

DNSO 実装の概要

DNSO の概要を図 5 に示す．クライアントが存在するエンドネットワークに NOCA (Name Override Client Agent) を配置，サーバが存在するエンドネットワークに NOSA (Name Override Server Agent) を配置する．また， NOCA および NOSA の連携を行うため ブローカーを NOCA と NOSA の間に設置する．NOCA とブローカー，ブローカーと NOSA は事前に VPN 等を用いて接続されたものと仮定する．

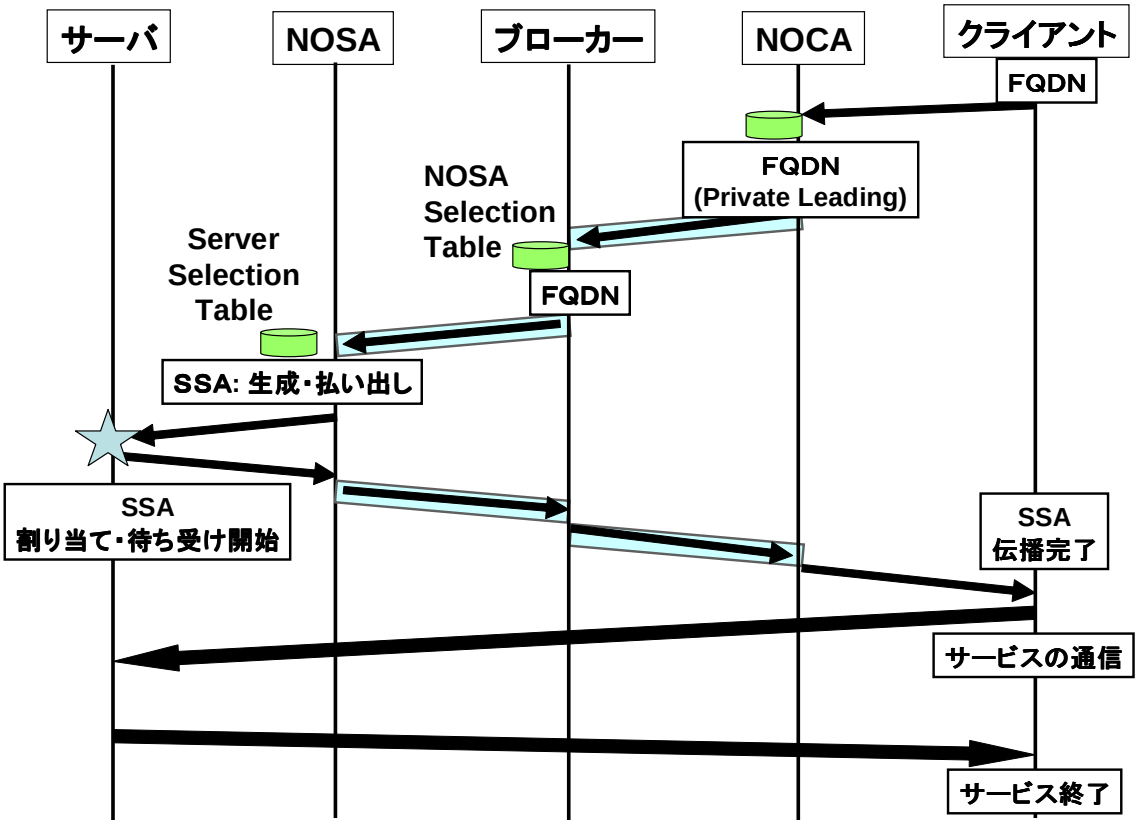


図 6: DNSO (DNS Name Space Override) の動作

図 6 に 実際の DNSO の動作を示す．DNSO の動作は以下の 9 個のステップより構成される．

1. クライアント： 名前解決
   クライアントはサーバの FQDN (Fully Qualified Domain Name) を名前解決するため， DNS クエリを DNS サーバに送信する．

2. NOCA： DNS クエリ のフックと判定
   NOCA は DNS クエリをフックし， PLR (Private Leading Resolving) の存在する名前一覧が登録された PLR Table に， FQDN が存在するか確認する． PLR Table に FQDN が見つかれば， FQDN 情報をブローカーに送信する

3. ブローカー： NOSA 選択
   ブローカーは， NOSA 情報と管理下の名前一覧が登録された NOSA Selection Table から， FQDN に対応する NOSA を選択し， その NOSA に FQDN 情報を送信する．

4. NOSA： FQSN， SSA 情報の生成・払い出し
   NOSA は FQDN 情報とランダムな文字列をハッシュし，そのハッシュ値の上位 64 bit を，IPv6 アドレスのインタフェース ID として使用してサーバのアドレスの prefix と合わせて SSA 情報を生成する． そして， サーバの名前一覧が登録された Server Selection Table から，FQDN 情報に対応するサーバを選択して，そのサーバに SSA 情報と FQDN 情報を送信する．

5. サーバ： SSA の付与
   サーバは FQDN 情報の表すサービスを検索し， そのサービスプロセスに対して SSA を付与する．ただし，DAD の結果， SSA 情報が衝突していれば SSA を付与しない．サーバは SSA を付与できたかどうかの結果を NOCA に返信する． このとき， NOSA とサーバにおいて， DAS (Delayed Address Setting) を用い， NOSA で SSA 情報を事前に生成して DAD 処理までを完了させておくと， サーバに SSA を付与するときのアドレス衝突を回避できる． また， DAD の処理に必要な一秒間を待つ必要がなくなる．

6. NOCA： SSA 情報の伝播
   NOCA はサーバでサービスプロセスに SSA を付与できたことを確認すると SSA 情報をブローカーに返す．サーバが SSA を付与できなかったときは， 再度 4 の処理から行う．

7. ブローカー： SSA 情報の伝播
   ブローカーは SSA 情報を NOSA に返す．

8. NOCA： SSA 情報の伝播
   NOCA は DNS クエリの Reply パケットを生成して， SSA 情報をクライアントに伝達する．

9. クライアント： サービス利用と終了
   クライアントは SSA 情報を用い，サーバにアクセスする．サーバはクライアントのサービス利用が終了すると，サービスを提供していたプロセスが終了し，同時に SSA を削除する．

Uncertain 状態の概要

従来の IPv6 におけるアドレスの状態遷移の基本的な仕様 [10] では以下のアドレスの状態がある．その状態を図 7 に示す．

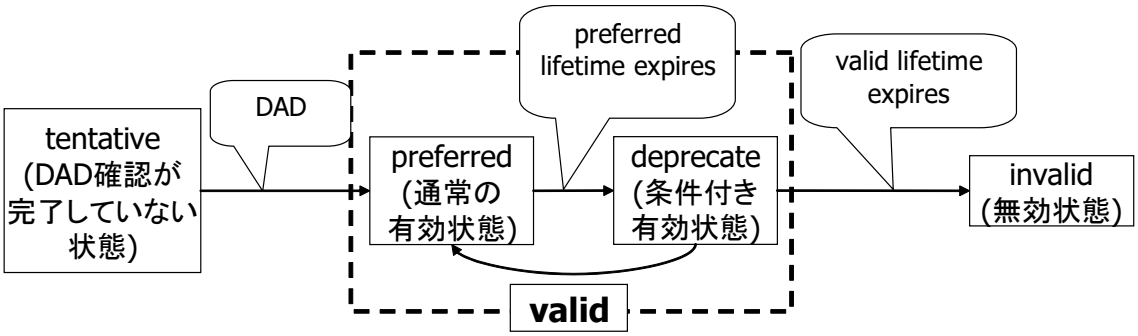


図 7: Legacy 通信のアドレス状態遷移

- Tentative 状態
  まだ有効ではなく，衝突確認 (DAD) を行っている状態
- Valid 状態
  - Preferred
    通常の有効状態
  - Deprecated
    条件付き有効状態，そのアドレスが宛先アドレスになっているパケットは受け取るが，送信元アドレスとしては設定しない．
- Invalid 状態
  無効になった状態

この Tentative 状態から Valid 状態に遷移するために一秒間の DAD 処理の時間を待たなければならない．その解決方法として Uncertain 状態を新しく導入する．

図 8 にあたらしいアドレス状態である Uncertain 状態を導入した際のアドレス状態遷移を示す．図に示すように，DAD 処理を終えるとアドレス状態は Tentative 状態から従来のような Valid 状態に遷移するのではなく，新しく導入した Uncertain 状態へ遷移させる．つまり， DAD 処理を Valid 状態になるよりこのような役割を負う Uncertain 状態は図 9 に示すような以下の二種類の NS メッセージに対する対応の仕方により定義することができる DAD 処理ための NS に対しては NA を返信し，重複していることを示し，他のノードが対象アドレスを設定できないようにする．リンク層アドレス解決のための NS に対しては全く返信せず．対象アドレスを設定しているノードは存在しないように他のノードには思わせる．この二種類の NS メッセージは発信元アドレスが明確に異なり（ DAD 処理の場合は未

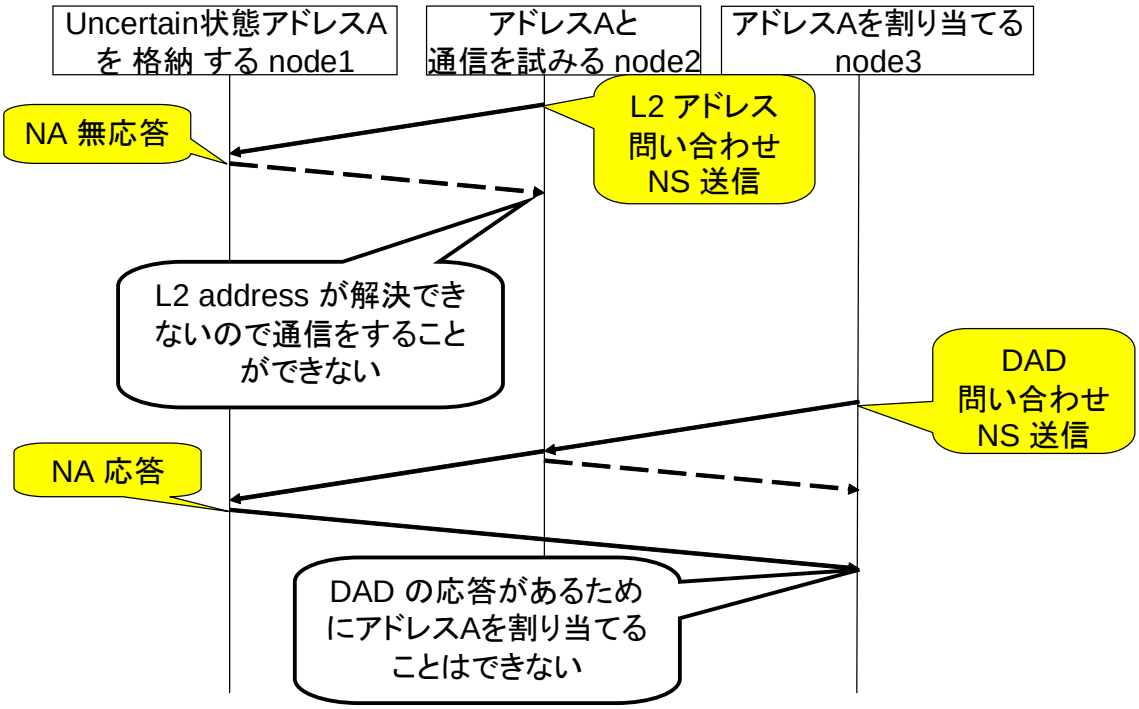


図 8: Uncertain 状態を導入したアドレス状態遷移

指定アドレス (::) が設定されている)，受信ノード側で容易に区別することが可能で，既に本機能の実装と動作検証を終えている．Uncertain 状態では対象のアドレスはいずれのノードにも設定されていないが，関係のない第三者ノードがそのアドレスを自身に設定できない状態であり，アドレスを予約している状態とみることができる．この状態を利用することによりアドレスをプールするという目的が果たせる．また DAD 処理を既に終えているため，必要時にアドレスの即時利用が可能となる．

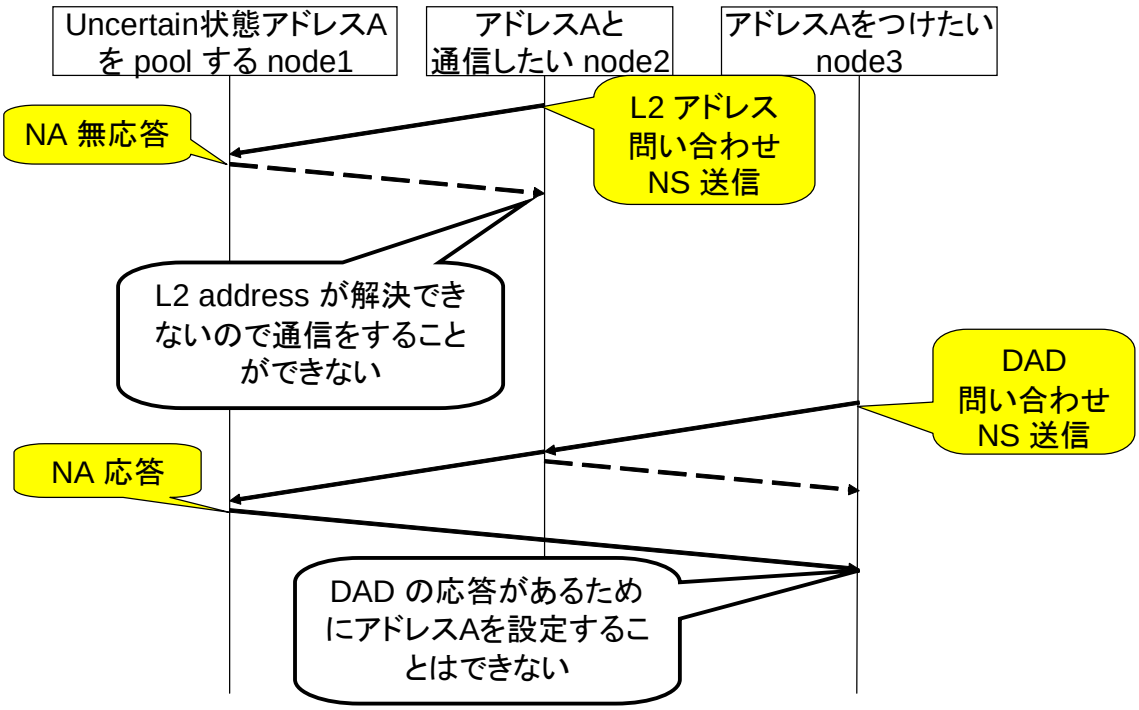


図 9: Uncertain 状態の概要

5　アドレス管理機能の実装方式

4 章で述べたアドレス管理機能の機能を実装することでアドレス管理機能がどのように動作するかを示す.

1. 最初に，ユーザがアドレス管理機能に対して生成するべきアドレスの prefix と生成するべきアドレス数を通知する．このとき， prefix は現在，ノードに割り当てられているものである必要がある.

2. アドレス管理機能はユーザからの通知を受け，指定 prefix のアドレスプールを作成，指定数のアドレスをアドレスプールに格納する.

3. 通信プロセスが宛先アドレスを確定させるタイミングで送信元アドレスの選択が行われる．このとき，宛先アドレスの prefix とアドレスプールに存在する prefix の比較を行い，近い prefix のアドレスを選択する.

4. アドレスプールからアドレスの選択が行われたことをアドレス管理機能がアドレスプールより通知を受ける.

5. アドレス管理機能はアドレスプールからの通知を受け，アドレスを再度生成し，適切な prefix のアドレスプールに対して格納を行う.

これらの動作を繰り返す.

5.1　アドレスプール機能の設計

5.1.1　アドレス生成機能

Uncertain 状態のアドレスの生成を行う際に，多様に対応可能であるシステムを構築するため，アドレス生成機能は管理機能とは独立したプロセスで実行する．アドレス管理機能はアドレスの生成を行う際にアドレス生成プロセスに対してどの prefix のアドレス値を生成するべきなのかを通知，アドレス生成プロセスはそれに応じたアドレス値を作成し，アドレス管理機能に対して通知を行う．図 10 にアドレスを生成する際に 生成プロセスに対してアドレス値生成の要求を出す様子を示す.

5.1.2　アドレス数管理機能

アドレスプール内に保持されている Uncertain アドレスの数を制御する．実装の容易さを考え，アドレスが消費 (Uncertain 状態から Valid 状態 [10] に遷移する) されるたびに新しい

Uncertain 状態のアドレスを プールに追加するという方法を採用する．これは，アドレスの状態を管理しているのはカーネルなので，カーネルにおいて，アドレスの状態が Uncertain から Valid に遷移したことを知らせるメッセージをルーティングソケットによって，カーネルからユーザランドに対して発信する．ユーザランドでは，メッセージを受け取るとアドレスの生成を行い，プールへ追加を行う．アドレス数管理機能は以下の動作の繰り返しになる．

1. アドレス管理機能はユーザから prefix ごとに Uncertain 状態のアドレスをあらかじめ生成しておく数の指定を受ける．
2. Uncertain 状態のアドレスを指定数生成する．
3. Uncertain 状態のアドレスが消費されるタイミングで カーネルが アドレス管理機能に対してルーティングソケットにより，どの prefix であるかのメッセージを通知する．
4. メッセージを受けたアドレス管理機能は Uncertain 状態のアドレスを再度生成する．

図 11 にアドレス数管理機能の例として prefix が一つの場合の例を示す．次のような手順でアドレス数を一定に保つことができる．

1. ユーザがアドレス管理機能に対して prefix の指定とアドレスプールに生成を行うアドレス数を指定する．
2. アドレス管理機能は指定数のアドレスを生成し，アドレスプールに対して格納を行う．
3. アドレスを取り出す際に，アドレスプールの情報を格納しているカーネルからアドレス管理機能に対して通知される．通知はルーティングソケットを使用する．
4. アドレス管理機能はアドレスの取り出しの通知を受け取ると，アドレスを生成し，そのアドレスをアドレスプールに対して格納を行う．

以上の動作により，初期値としてユーザが指定したアドレス数をアドレスプールが常に確保することになる．

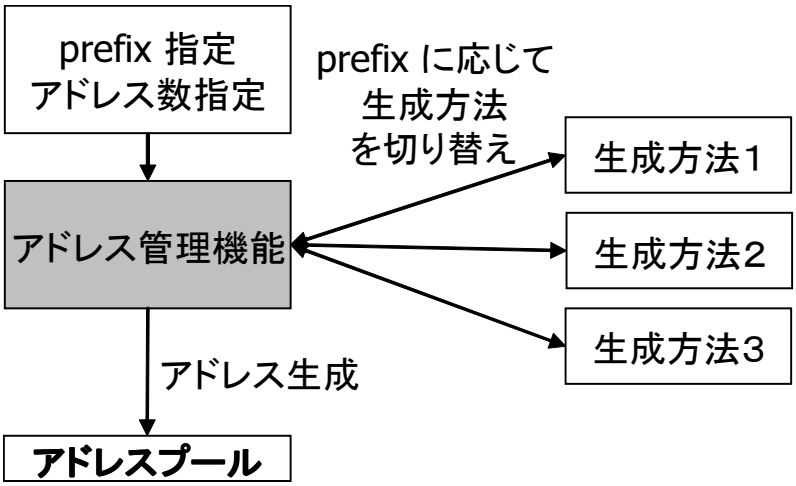


図 10: アドレス管理機能におけるアドレス生成処理

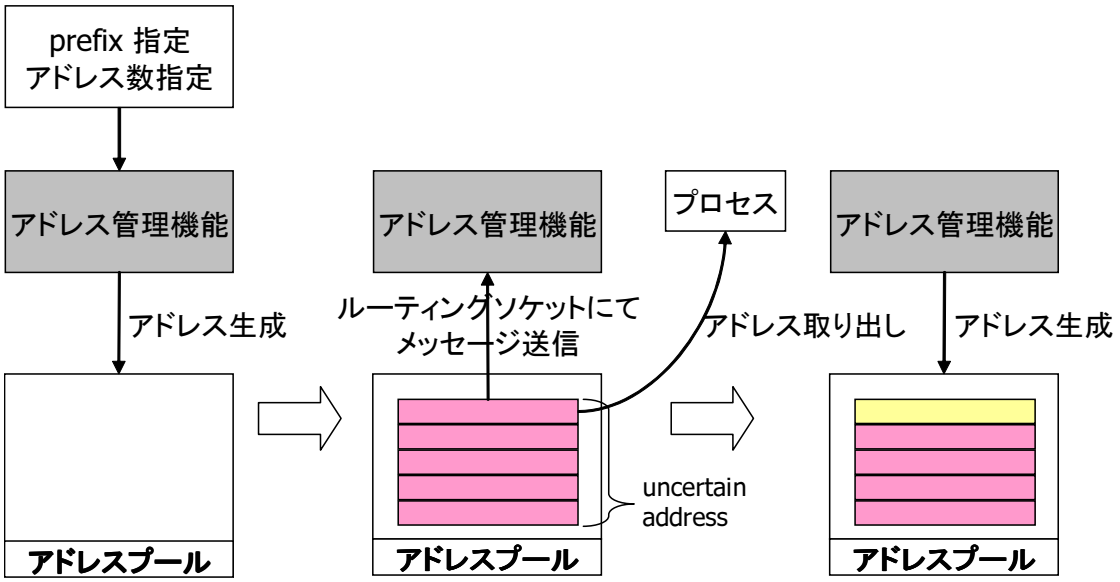


図 11: アドレス管理機能におけるプールアドレス数の制御

5.2 送信元アドレス選択機能の設計

IPv6 において送信元アドレスの選択である source address selection [11] は宛先アドレスと最もアドレスが近いアドレスを選択するべきとしている．これはネットワークインタフェースに割り当てられているアドレス全ての検索を行うことによって実現されている．それに対し，Unified Multiplex 通信の 送信元アドレスの選択は宛先アドレスの prefix とアドレスプールに保持されているアドレスの prefix の比較で行われるべきである．まず，アドレスプール構造は Legacy 通信の様に ネットワークインタフェース に設定されているアドレスを単一のリスト構造で持つのではなく，ネットワークインタフェース，prefix，アドレスとするべきである．それによって，送信元アドレス選択は prefix 部分の検索を行い，最も近い prefix のアドレスプールから取り出すことが可能になる．図 12 にアドレスプール構造と 送信元アドレス選択 を行う際に検索する範囲を Legacy 通信のアドレスが格納されている様子と Unified Multiplex 通信のアドレスプールの構造を比較し示す．

Legacy 通信 ネットワークインタフェースに割り当てられている全てのアドレスを検索し，最も宛先アドレスに対して近いアドレスを送信元アドレスとして選択する．

Unified Multiplex 通信 ネットワークインタフェースに安全な割り当てられているアドレスを prefix ごとのアドレスプール内に prefix による分類を行い，格納する．送信元アドレスの選択は prefix 情報の比較により宛先アドレスの prefix に近い prefix を格納しているアドレスプールの選択を行い，そのアドレスプール内のアドレスからは最も取り出しやすいアドレスを選択する．

以上にアプリケーションに対してサービス専用アドレスを提供し，ユーザがサービス専用アドレスを用いた Unified Multiplex 通信による通信を行うために必要なアドレス管理機能の不足箇所の実装方式を示した．これらの機能により，アドレスの生成から削除までが管理され行われるので，ユーザはより，Unified Multiplex 通信を使用しやすくなったといえる．

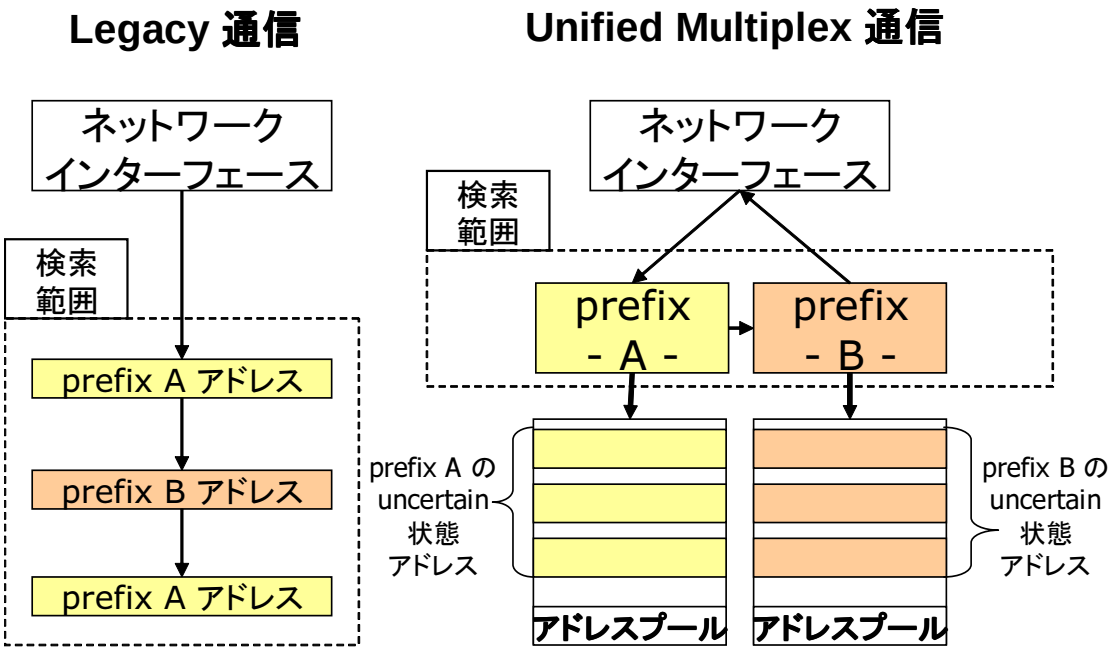


図 12: アドレスプールの構造とアドレス検索範囲

6 Uncertain 状態の検証

Uncertain 状態は従来の通信アーキテクチャにない状態であるのでこの新しいアドレスの状態が既存の環境に対して悪影響を及ばさないことは重要である．そこで既存のネットワーク上で Uncertain 状態を持つ Unified Multiplex 通信アーキテクチャが存在した際に悪影響を及ぼさないのか，Uncertain 状態が正常に機能するのかを実験によって確認する．

6.1 実験環境

図 13 に Uncertain 状態を検証する際の実験環境を示す．Unified Multiplex 通信を可能にするカーネルを実装したノードを下記の OS を搭載するノードが存在するネットワークに接続する．

1. FreeBSD 6.2R
2. Windows XP Home Edition version 2002 SP3
3. Windows Vista Ultimate
4. Mac OS X 10.3.9
5. Ubuntu 8.04.1

6.2 実験手順

以下に述べる手順で実験を行った．

1. Unified Multiplex 通信アーキテクチャを実装した FreeBSD で 2001:380:500d:1::222 (以下，アドレス A) ，2001:380:500d:1::333 (以下，アドレス B)，を Uncertain 状態として保持する．
2. 実験対象クライアントは アドレス A を インターフェースに対して割り当てを行う．
3. 実験対象クライアントは アドレス B に対して通信を試みる．(ping6 による疎通テスト)

このとき，実験対象クライアントは FreeBSD， Windows XP，Windows Vista，Mac OS X， Ubuntu を用意した．これらのクライアントで上記の 2. と 3. の手順を繰り返し行う．

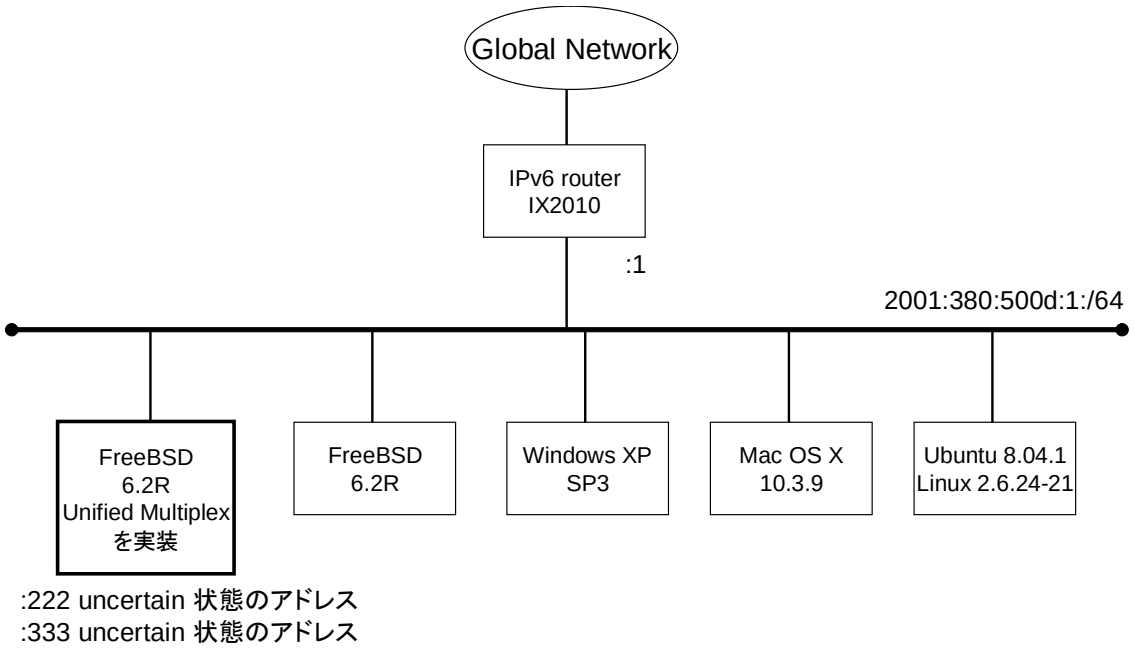


図 13: Uncertain 状態を検証する実験環境

6.3 実験結果と考察

以下に各実験対象クライアントごとに インターフェースにアドレスを割り当てた際の DAD 処理の様子と 通信をする際に L2 address 解決を行ったときの結果と考察を述べる.

通常の FreeBSD

- DAD
  以下のコマンドでアドレス A をインターフェースに割り当てる.

  ```
  $ ifconfig xl0 inet6 2001:380:500d:1::222 alias
  ```

  設定後のアドレス状態.

  ```
  $ ifconfig xl0
  xl0:  inet6 2001:380:500d:1::222 prefixlen 64 duplicated
  ```

  となり，2001:380:500d:1::222 の状態が duplicated つまり，重複していることを通知していることが確認できる.

- L2 address
  以下のコマンドでアドレス B に対して通信を試みる.

```
$ ping6 2001:380:500d:1::333
PING6(56=40+8+8 bytes) 2001:380:500d:1:208:74ff:fe41:51d7 -->
2001:380:500d:1::333

--- 2001:380:500d:1::333 ping6 statistics ---
1 packets transmitted, 0 packets received, 100.0% packet loss
```

  となり，通信ができないことが確認できる.

Windows XP

- DAD
  以下のコマンドでアドレス A をインターフェースに割り当てる.

```
$ netsh interface ipv6 add address 5 2001:380:500d:1::222
```

  設定後のアドレス状態.

```
$ ipv6 if 5
Interface 5: Ethernet: ローカル エリア接続
    duplicate global 2001:380:500d:1::222, life infinite (manual)
```

  となり，2001:380:500d:1::222 の状態が duplicate つまり，重複していることを通知していることが確認できる.

- L2 address
  以下のコマンドでアドレス B に対して通信を試みる.

```
$ ping6 2001:380:500d:1::333

Pinging 2001:380:500d:1::333
from 2001:380:500d:1:b843:7461:bb38:a392 with 32 bytes of data:

Reply from 2001:380:500d:1:b843:7461:bb38:a392:
```

```
宛先アドレスに到達できません
Reply from 2001:380:500d:1:b843:7461:bb38:a392:
宛先アドレスに到達できません
Reply from 2001:380:500d:1:b843:7461:bb38:a392:
宛先アドレスに到達できません
Reply from 2001:380:500d:1:b843:7461:bb38:a392:
宛先アドレスに到達できません

Ping statistics for 2001:380:500d:1::333:
   Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

となり，通信ができないことが確認できる．

Windows Vista

- DAD
  以下のコマンドでアドレス A をインターフェースに割り当てる．

```
$ netsh interface ipv6 add address 9 2001:380:500d:1::222
```

設定後のアドレス状態．

```
$ netsh interface ipv6 show address 9

Address 2001:380:500d:1::222 Parameters
------------------------------------------------------------
Scope Id            : 0.0
Valid Lifetime      : infinite
Preferred Lifetime : infinite
DAD State           : Duplicate
Address Type        : Manual
```

となり，2001:380:500d:1::222 の DAD State が Duplicate つまり， DAD 処理を行った結果，重複していることを通知していることが確認できる．なお， Windows Vista の場合アドレス A を設定するとアドレスが重複していることを示す window がポップアップ表示される．そのポップアップ表示を図 14 に示す．

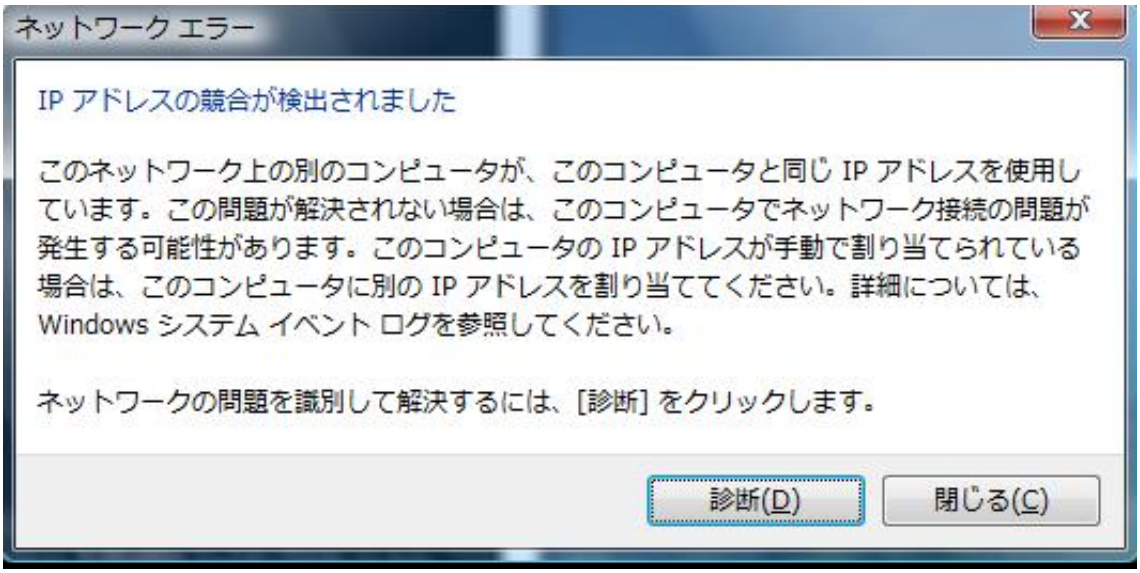


図 14: Windows Vista でアドレスの重複を検知した際のポップアップ表示

- L2 address
  以下のコマンドでアドレス B に対して通信を試みる.

```
$ ping 2001:380:500d:1::333

Pinging 2001:380:500d:1::333 from 2001:380:500d:1:5c0c:1e14:720b:bdab
with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 2001:380:500d:1::333:
   Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

  となり，通信ができないことが確認できる.

Mac OS X

- DAD
  以下のコマンドでアドレス A をインターフェースに割り当てる.

  ```
  $ ifconfig en0 inet6 2001:380:500d:1::222 alias
  ```

  設定後のアドレス状態.

  ```
  $ ifconfig en0
  en0: inet6 2001:380:500d:1::222 prefixlen 64 duplicated
  ```

  となり，2001:380:500d:1::222 の状態が duplicated つまり，重複していることを通知していることが確認できる.

- L2 address
  以下のコマンドでアドレス B に対して通信を試みる.

  ```
  $ ping6 2001:380:500d:1::333
  PING6(56=40+8+8 bytes) 2001:380:500d:1:20d:93ff:fe3d:bb36 -->
  2001:380:500d:1::333

  --- 2001:380:500d:1::333 ping6 statistics ---
  1 packets transmitted, 0 packets received, 100% packet loss
  ```

  となり，通信ができないことが確認できる.

Ubuntu

- DAD
  以下のコマンドでアドレス A をインターフェースに割り当てる.

  ```
  $ sudo ifconfig eth0 add 2001:380:500d:1::222/64
  ```

  設定後のアドレス状態.

  ```
  $ ifconfig eth0
  eth0      inet6 アドレス: 2001:380:500d:1::222/64 範囲:グローバル
  ```

となり，他の OS の様に重複していることは確認できない．しかし，dmesg を参照すると以下のように，

```
$ dmesg
[ 9627.644768] eth0: duplicate address detected!
```

と重複していることを検知したことを確認することができる．

- L2 address
  以下のコマンドでアドレス B に対して通信を試みる．

```
$ ping6 2001:380:500d:1::333
PING 2001:380:500d:1::333(2001:380:500d:1::333) 56 data bytes


--- 2001:380:500d:1::333 ping statistics ---
2 packets transmitted, 0 received, 100% packet loss, time 1009ms
```

  となり，通信ができないことが確認できる．

以上のように 5 種類の OS を用意し， Uncertain 状態が存在する環境でも異常をきたすことなく，正常に重複を検知し．また，通信できないアドレスに対しては，通信を行えないことを検出することが確認できた．

7　まとめと今後の課題

本報告では，Unified Multiplex 通信アーキテクチャで導入する新しいタイプのアドレスとしてサービス専用アドレスのの特性を分析し，それらを実現するための実装方式を示した．そして，サービス専用アドレスの実現に不足していた機能であるアドレスプールの管理機能を追加した．今後の課題としてはアドレス生成，管理の効率性を考え，アドレスプールの共有化を行い，エンドサイト全体で統計多重を行い，適切なアドレス数をもとめ管理を行うなどアドレス管理機能に効率的な機能を追加していく予定である．

謝辞

本報告を終えるにあたり，御指導，御教授をいただきました大阪大学大学院情報科学研究科の村田正幸教授に深く感謝いたします．また，本報告において終始直接御指導頂いた大阪市立大学の阿多信吾准教授には多くの助言をいただきましたことを，心よりお礼申し上げます．また，厳しく様々な指導をいただいた日本電気株式会社の北村浩氏に心よりお礼申し上げます．

また，平素から適切なご助言をいただいた大阪大学大学院情報科学研究科の若宮直紀准教授，大阪大学大学院情報科学研究科の荒川伸一助教，大阪大学大学院経済学研究科の大下裕一助教に深く感謝いたします．

最後に，常日頃から様々な相談に応じて頂き，支えてくださった古田晋也氏，黄恵聖氏をはじめとする村田研究室の皆様方，榊間慧一氏，佐藤寧洋氏，田路祐介氏をはじめとする大阪市立大学情報ネットワーク工学研究室の皆様方に心からお礼を申し上げます．

参考文献

[1] 北村 浩, 阿多 信吾, 村田 正幸, “IP 通信のセッション多重化を刷新する Unified Multiplex 通信アーキテクチャ,” **電子情報通信学会研究会** *(IN2006-134)*, vol. 106, pp. 121–126, Dec. 2006.

[2] H. Kitamura, S. Ata, and M. Murata, “IPv6 ephemeral addresses,” *Internet-Draft,* *draft-kitamura-ipv6-ephemeral-address-00*, Oct. 2008. work in progress.

[3] T. Narten, E. Nordmark, W. Simpson, Daydreamer, and H. Soliman, “Neighbor discovery for IP version 6 (IPv6),” *RFC4861*, Sept. 2007.

[4] P. Mockapetris, “Domain names-concepts and facilities,” *RFC1034*, Nov. 1987.

[5] H. Kitamura, S. Ata, and M. Murata, “Harmless IPv6 address state extension (uncertain state),” *Internet-Draft,* *draft-kitamura-ipv6-uncertain-address-state-00*, Oct. 2008. work in progress.

[6] W. R. Stevens, *TCP/IP Illustrated, Volume 2: The Implementation.* Reading, Massachusetts: Addison-Wesley, Jan. 1995.

[7] R. Hinden and S. Deering, “IP version 6 addressing architecture,” *RFC 4291*, Feb. 2006.

[8] 島 成佳, 北村 浩, “サービス専用アドレス情報の伝播機構 (DNSO) を基盤とした IPv6 グローバル通信アーキテクチャ,” **電子情報通信学会研究会** *(IN2008-29)*, vol. 108, pp. 17–22, July 2008.

[9] 阿多 信吾, 北村 浩, 村田 正幸, “サービス専用のアドレスを実現する Unified Multiplex 通信アーキテクチャ　〜 アーキテクチャの設計 〜,” **電子情報通信学会研究会** *(IN2007-239)*, vol. 107, pp. 479–484, Mar. 2008.

[10] S. Thomson, T. Narten, and B. Jinmei, “RFC 4862: IPv6 stateless address autoconfiguration,” Sept. 2007.

[11] R. Draves, “Default Address Selection for Internet Protocol version 6 (IPv6),” *RFC3484*, Feb. 2003.