NAIST-IS-MT9751018

修士論文

IPv6 におけるエニーキャストの実装と評価

大江 将史

1999年3月19日

奈良先端科学技術大学院大学
情報科学研究科 情報システム学専攻

本論文は奈良先端科学技術大学院大学情報科学研究科に
修士 (工学) 授与の要件として提出した修士論文である。

大江 将史

指導教官 ： 福田 晃 教授
湊 小太郎 教授
最所 圭三 助教授

IPv6におけるエニーキャストの実装と評価[*]

大江 将史

内容梗概

現在のインターネットにおける通信プロトコルであるIPv4プロトコルは，近年の急激なインターネットの成長により，IPアドレスの枯渇という大きな問題を抱えている．そこで，この問題の解決のために，IPv6が登場した．IPv6では，IPv4の通信方式であるユニキャストやマルチキャストに加えて，エニーキャストが新たに定義された．エニーキャストは，クライアントから最も経路的に近いサーバへの通信を実現する．しかし，エニーキャストを使うためには，経路の変化による通信中の相手の変化が問題となる．この問題の解決策として「始点識別オプション方式」が提案されたが，その実装や運用は報告されていない．そこで，本論文では，新たな解決策「通信前エコー方式」「TCPオプション方式」を提案し，3つ方法の比較を行ない，「始点識別オプション方式」と「通信前エコー方式」の実装と実験を行う．その結果，両方式が問題点の解決に有効な方法であることを実証した．さらに，両者の実験結果の比較を行い，エニーキャストを考慮した通信方式の検討を行った．

キーワード

IPv6, エニーキャスト, 始点識別オプション, 通信前エコー, TCPオプション, 実装, 運用

---

[*] 奈良先端科学技術大学院大学 情報科学研究科 情報システム学専攻 修士論文, NAIST-IS-MT9751018, 1999年3月19日.

Implementation and Evaluation of a IPv6 Anycast*

Masafumi Oe

Abstract

As the Internet grows explosively in recent years, the current Internet protocol "IPv4" has a problem of running out of IP address. Then, "IPv6" has been defined to solve the problem. In addition to unicast and multicast which were defined as the communication method in IPv4, "anycast" has been defined in IPv6. Anycast can realize that client communicates with the "nearest" one of servers, according to the routing protocols' measure of distance. However, it is a problem that the peer of the communication using anycast may be changed when the route changing occured on the Internet. Hence, "Source Identification Option" was proposed as a solution of this problem, and neither the implementation nor operation have been reported. Thus, in this thsis, new solutions ,"Echo Before Communication" and "TCP option way", are proposed. From the conmparison of three techniques, "Source Identification Option" and "Echo Before Communication" are implemented and experimented.

As a result, It is shown that both of proposals are effective solutions of the problem. Moreover, the communication protocol which anycast is considered is considered is evaluated from the comparing of the experiment result. as a base for comparing the experiment results is evaluated.

Keywords:

IPv6, anycast, Source Identification Option, Echo Before Communication, TCP option, Operation, Implementaion

---

*Master's Thesis, Department of Information Systems, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-MT9751018, March 19, 1999.

目 次

図 目 次

表目次

1. はじめに

本章では，本研究の背景，IPv6の概要，および，エニーキャストとその問題点について述べることにより，本研究の目的とその重要性を示す．

1.1 研究の背景

インターネットは，1981年に登場したインターネット・プロトコル・バージョン4 (Internet Protocol Version 4, 以下IPv4) [1] という通信規約の下に，世界中の計算機が接続された巨大なコンピュータ・ネットワークであり，今や，人類にとってなくてはならないインフラストラクチャとしての役割を担うようになった．しかし，巨大になったインターネットは，IPv4の規約を定めた際には考慮できなかったさまざまな問題を抱えるようになった．特に深刻な問題として，「IPアドレスの枯渇」がある．

インターネットでは，接続された計算機を識別するために，IPアドレスと呼ばれる整数値を用いており，IPv4のアドレスの範囲は，32bitを有している．つまり，IPv4によるインターネットには，2^{32}台，すなわち，約43億を越える計算機を接続できる計算になる．しかし，近年の指数関数的な計算機の増加や，非効率なアドレス割り当てにより，2000年初頭には，割り当てることのできるアドレスはなくなると予想されている[2]．これが，IPアドレスの枯渇問題である．

この問題を解決するために，IETF[1] で，IPv4に代わる次世代のインターネット・プロトコルについて，論議が行われた．そして，IPv4の欠点を補い，時代に合わなくなった不要な部分を除き，新しい要求を取り入れたインターネット・プロトコル・バージョン6(Internet Protocol Version 6, 以下IPv6) が1995年に誕生した[3]．

---

[1] Internet Engineering Task Force http://www.ietf.org/

1.2 IPv6の概要

IPv6は，IPv4の抱えている問題点を解決するために，IETF IPng[2]分科会でStephan E.DeeringとRobart M. Hindenが提案したネットワーク層の通信プロトコルである．IPv6の特徴を以下に述べる．

1. アドレス空間の拡張

   IPv6のアドレス空間は，IPv4の32bitから128bitへ拡大された．すなわち，アドレス空間が，$2^{96} \cong 8 * 10^{23}$倍に拡大したことになる．IPv6で扱えるアドレス数，すなわち，インターネットに接続できる計算機の数は，$2^{128} \cong 3.4 * 10^{32}$となった．

2. ヘッダ・フォーマットの簡略化

   IPv6のヘッダ・フォーマットは，IPv4の優れた部分を引き継ぎ，不要な部分を取り除いたものになっている．図1と図2に，IPv4とIPv6のヘッダ・フォーマットをそれぞれ示す．IPv6では，ヘッダ長の固定や，チェックサムなどの使われなくなった項目を除去することにより，IPヘッダの処理の負荷の軽減を図っている．

3. 数珠つなぎ拡張ヘッダ

   IPv6では，IPv4に存在したオプションに代わるものとして，拡張ヘッダが定義された．拡張ヘッダとは，IPヘッダとTCP(Transmission Control Protocol)[4]やUDP(User Datagram Protocol)[5]のヘッダとの間に，任意の数だけ挿入できるヘッダのことで，各拡張ヘッダは，ヘッダ・タイプと次に続くヘッダ・タイプを保持し，最後の拡張ヘッダのみがペイロードタイプを保持している(図3を参照)．つまり，ヘッダは，数珠つなぎ(daisy chain)」となっている．

   拡張ヘッダには次の6つのタイプが定義されている．

---

[2] Internet Protocol Next Generation

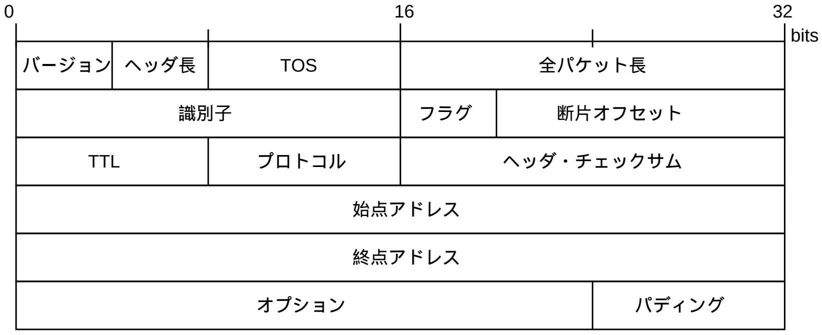


図 1 IPv4 ヘッダ・フォーマット

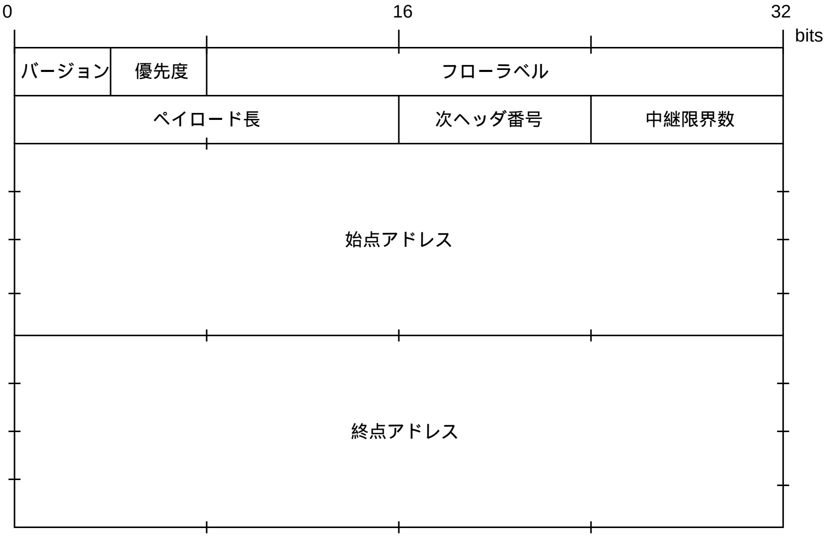


図 2 IPv6 ヘッダ・フォーマット

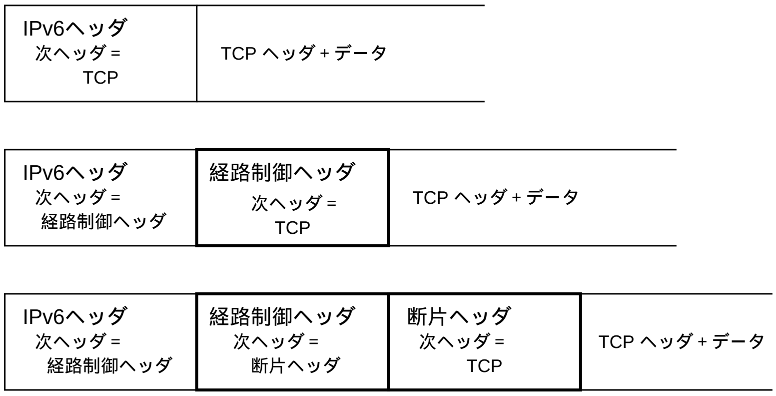


図 3　数珠つなぎヘッダの例

(a) 中継点オプション・ヘッダ (Hop-by-hop options header)

このオプション・ヘッダは，経由する中継点(ルータ)毎に処理される．

(b) 経路制御ヘッダ (routing header)

このオプション・ヘッダは，ポリシールーティングのために用いる．ポリシールーティングとは，経由する中継点 (ルータ) を指定したルーティングのことである．

(c) 断片ヘッダ (flagment header)

IPv6は，中継点において，MTU(Maximum Transmission Unit，IP層の下位であるデータリンク層におけるパケットの大きさの最大値)より大きいパケットを分割しない．よって，このオプション・ヘッダは，始点のホストにおいて，経路のMTUにあわせて，あらかじめパケットを断片化する際に用いる．終点のホストでは，断片ヘッダを元に，パケットを組み立て直す．

(d) 認証ヘッダ (authentication header)

データの正当性を明示的に保証するために用いる．

(e) 暗号ペイロード (encapsulating security header)

データを暗号化するために用いられる．これ以降のデータは，暗号化される．

   (f) 終点オプション・ヘッダ (destination options header)

   終点ホストでのみ処理されるオプション・ヘッダである．

   以上に示した数珠つなぎ拡張ヘッダは，任意の数だけオプション・ヘッダを IP パケットに組み込むことができる．これにより，IPv6 では，柔軟なオプション設定と拡張性を実現している．

4. セキュリティへの対応

   拡張ヘッダである認証ヘッダと暗号ペイロードにより，IPv6 は，IP 層でのセキュリティを実現する．

5. アドレス体系の拡張

   IPv6 のアドレスは，次の 3 種類が定義されている．

   (a) ユニキャスト・アドレス

   1 対 1 のホスト間の通信に用いるアドレス (図 4参照)

   (b) マルチキャスト・アドレス

   特定のグループに属するホスト間同士の通信に用いるアドレス (図 5 参照)

   (c) エニーキャスト・アドレス

   特定のグループに属するホストの中で最も経路的に近いホストと 1 対 1 の通信に用いるアドレス (図 6参照)

   IPv6 では，IPv4 にあったユニキャスト，マルチキャストに加えて，新たにエニーキャストが定義された．

以上に，IPv6 の特徴を列挙した．IPv6 は，IPv4 の単なるアドレス拡張したものではなく，IPv4 の運用経験を生かして大幅に改良を加え，拡張したものである．

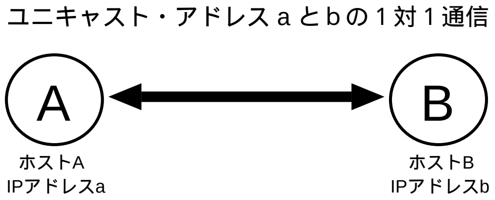


図 4　ユニキャスト

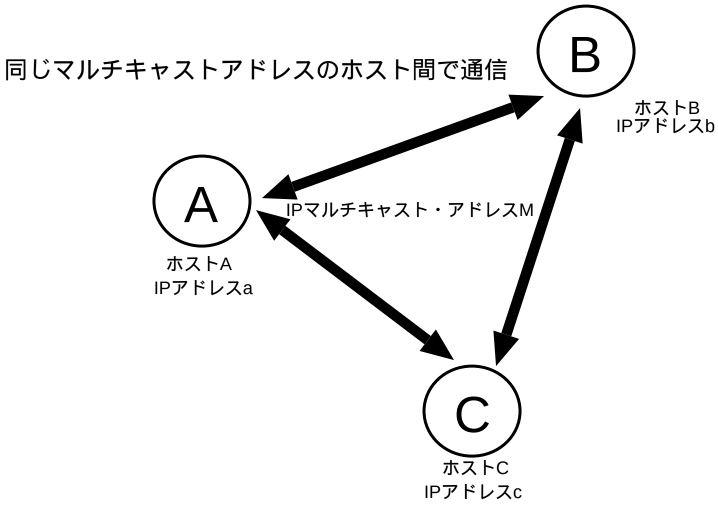


図 5　マルチキャスト

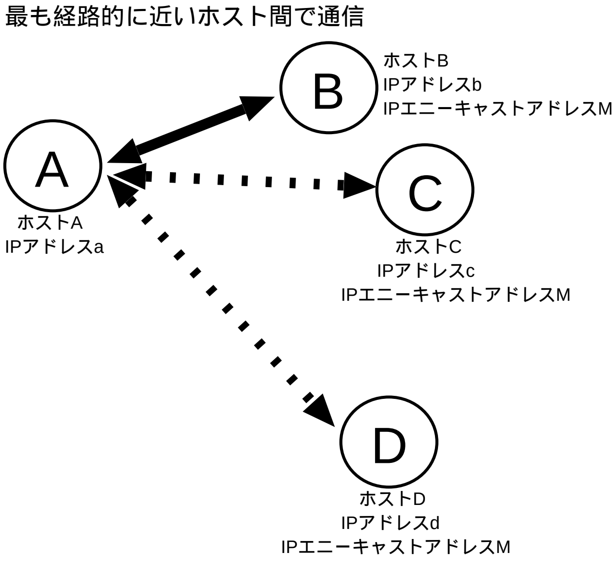


図 6　エニーキャスト

1.3 エニーキャストの概要

ここでは，エニーキャストの概要を述べる．

エニーキャストを用いた通信は，エニーキャスト・アドレスでグループ化されたホストの中から最も経路的に近いホストと通信を実現する．そして，IP パケットの伝達の方向を決定する経路制御機構が，「近さ」を判断する．

IPv6 では，エニーキャストの利用条件として，

1. エニーキャスト・アドレスを始点アドレスに使用してはならない．
2. エニーキャスト・アドレスのみを持つホストが存在してはならない．
3. エニーキャスト・アドレスは，ユニキャスト・アドレスの空間の一部である．これらの区別は，エニーキャスト・アドレスが割り当てられたホストでしかできない．つまり，「エニーキャストとして使う」としたユニキャスト・アドレスが，エニーキャスト・アドレスとなる．ただし，例外として，ルーティングやモーバイル IP のためにアドレスが定義されている [6][7]．

と定めている．そして，エニーキャストの持つ特性を利用すれば，次に示すことを実現できる．

1. 最短サーバの自動選択

   経路的に最も近いホストへのアクセスを実現する．

2. サービスに対する抽象的なアクセス

   DNS(Domain Name System, ホスト名に対応する IP アドレスの管理や提供を行う機構)[8][9] や代理サーバ (プロキシサーバ) といった特定のサービスにエニーキャストを用いることにより，サービスを提供するサーバを複数覚えておく必要がなくなる．よって，そのサービスの内容に対するエニーキャスト・アドレスを一つ覚えれば良い．

3. サービスの冗長化による信頼性の向上と負荷分散

エニーキャストによる通信とは，エニーキャスト・アドレスを持つ複数のホストの中の一つと通信することである．よって，複数のサーバによりサービスが冗長化されることになり，信頼性の向上や，サーバにかかる負荷を分散することが可能になる．

4. ポリシー・ルーティングでの利用

   組織内のルータをグループ化し，エニーキャスト・アドレスを割り当てる．そして，経路制御オプションにより，そのエニーキャスト・アドレスを中継点に指定すれば，「組織内のあるルータを経由」というような曖昧なルーティングが可能となる．

以上で示したことは，エニーキャストの概念と機能である．

1.4 エニーキャスト通信の仕組み

ここでは，1.3節のエニーキャストの概要について述べた「最も経路的に近いホストとの通信」をどのように実現するかについて詳しく述べる．

IP では，始点ホストから終点ホストへと送信された IP パケットは，複数のルータを経由して終点へ伝達される．(図 7の H_1から H_4の場合は，ルータを 2 つ経由している．H_4と H_5間の通信の場合など，終点が同じネットワークにある場合は，ルータは経由せずに直接通信する．)

ルータでは，IP パケットの終点アドレスとルータの保持している経路情報とに基づいて，適切なルータもしくはホストへ転送 (ルーティング) する．この繰り返しにより，IP パケットは，終点アドレスのホストへと伝達される．現状の IP のルーティングでは，ルーティング距離，すなわち，経由するルータの数 (ホップ数) が最短となるように経路情報を更新するという特性を持っている．

同一の終点アドレスに対する複数の経路情報がルータにある場合，ルータは，先に示した特性に従って，複数の経路情報の中から「ルーティング距離が最短となる経路」を終点アドレスに対する経路情報として採用する．

つまり，同一のエニーキャスト・アドレスを持つ複数のホストをインターネット上に配置し，経路情報を更新すれば，IP ルーティングの特性に従って，送信元

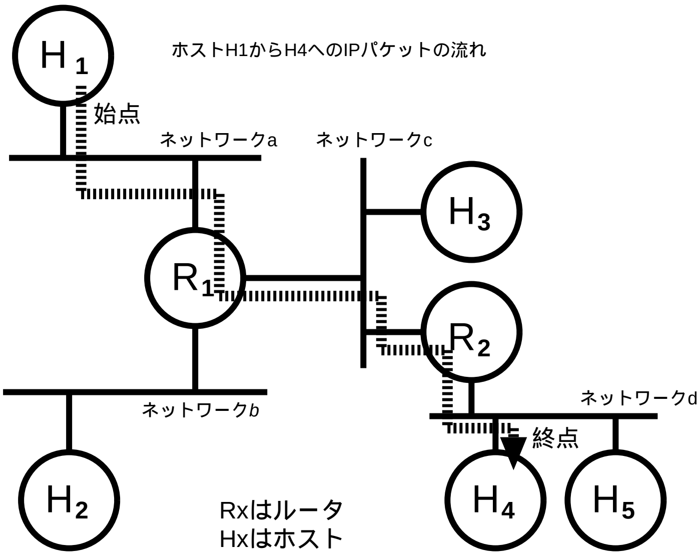


図 7　IP パケットの伝播

ホストからルーティング距離が最も近いエニーキャスト・アドレスを持つホストへ IP パケットが到達する．図 8に例を示す．ホスト H_2と H_4には，エニーキャスト・アドレス M が付加されている．この状態では，終点アドレス M へのパケットは，ホスト H_1からはホスト H_2へ，ホスト H_3からは，ホスト H_4へそれぞれ伝播する．

以上が，エニーキャストの 最も近いホストと 1 対 1 の通信」の仕組みである．

1.5　エニーキャスト通信の問題点

この節では，エニーキャストを扱う際に問題となる「エニーキャスト経路問題」について述べる．

IP では，通信中の経路の固定を保証していない．これは，図 9に示すように，ホスト H_1から H_2へパケットは到達するが，パケットの経路として，ルータ R_1，R_2のどちらを経由するかは固定できないということである．そして，エニーキャストは，経路の距離的に最も近いホストとの通信を行うので，経路の固定が保証

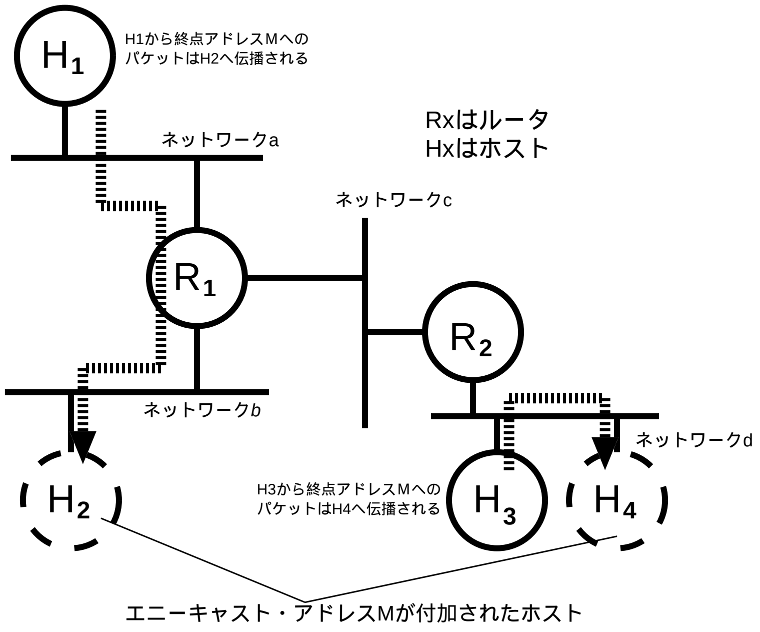


図 8 エニーキャストの伝播

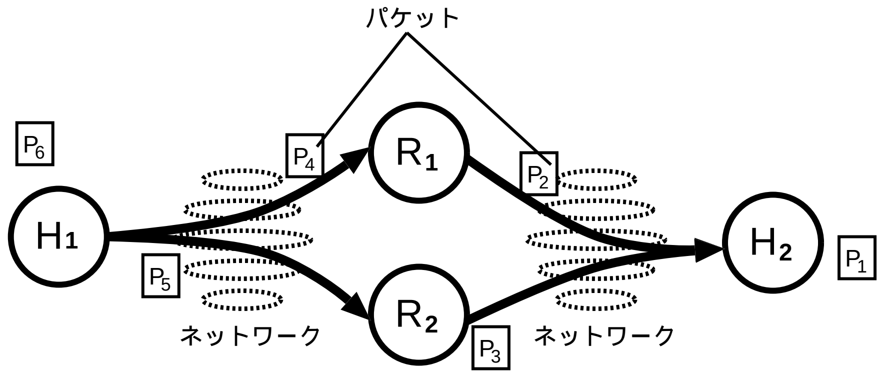


図 9 異なる経路を通るパケットの概念図

されないことは，通信相手の固定が保証されないことを意味している．これは，

図10に示すように，ステートフル・プロトコル(State-full protocol, 状態遷移を伴う通信プロトコル)を用いた場合に通信の断絶を引き起こす．これを「エニーキャスト経路問題」という．

1.6 本研究の目的とその重要性

1.5節で述べた「エニーキャスト経路問題」の解決策の一つとして，J. Boundらによって「始点識別オプション方式」が，インターネット・ドラフト(通信の規約となる前段階の草案)として1997年7月に提案された[10]．しかし，この提案は，発行から6か月以内に再提案やRFCへの進展がなかったため1998年1月に廃棄された(このため，文献[10]は現在存在しない)．

よって，現状のIPv6では，エニーキャストを利用する上で問題となるエニーキャスト経路問題に対する実装が存在せず．また，実験報告もない．

そこで，本論文では，「エニーキャスト経路問題」の解決策を新たに2つ提案し，提案方式と始点識別オプション方式についての考察と実装を行い，実験としてエニーキャストを用いたサービスを運用し，その結果，明確になった問題点についての考察を行う．

以下，第2章では，エニーキャストのための通信方式を提案する．第3章では，提案方式の実装について述べ，第4章に実験結果とその評価を示す．最後に，第5章で結論を述べる．

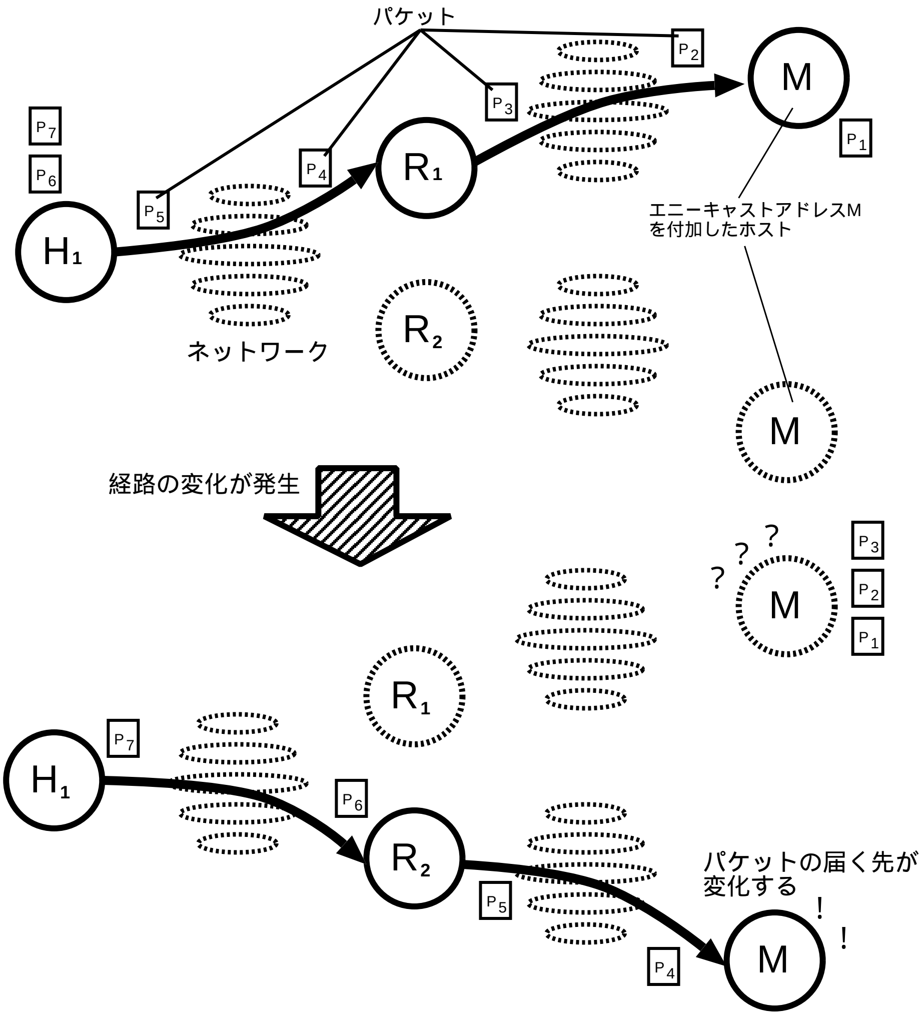


図 10 経路の変化にともなう通信先の変化

2. エニーキャストのための通信方式の提案

本章では，第1章で述べたエニーキャストのもつ問題点である「エニーキャスト経路問題」の解決策について述べる．

2.1 エニーキャストの経路問題の解決方式

ここでは，「始点識別オプション方式」と，新たに提案する「通信前エコー方式」および「TCP オプション方式」を示す．

2.1.1 始点識別オプション方式

J. Bound らが提案した始点識別オプション方式 [10] は，通信開始時は「エニーキャスト・アドレス」，通信自体は「ユニキャスト・アドレス」というように，アドレスの変更によってエニーキャスト経路問題を解決している．このアドレスの変更は，サーバで行われ，その変更をサーバからクライアントへ通知するために，IPv6 の終点オプションの一つとして「始点識別オプション」を定義している．このオプション内の識別子フィールド (図 11参照) に，変更前の始点アドレス (エニーキャスト・アドレス) を組み込み，クライアントに通知することでアドレスを変更する．

次に，このオプションのトランスポート層での利用を示す．

トランスポート層は，ネットワーク上のプロセス間でのデータ伝送路を提供し，そのプロトコルには，TCP と UDP がある．

TCP の特徴は，上位の層に対し，ストリーム型の入出力インターフェース (データの区切りを意識させない) と信頼性をもったコネクション型 (通信状態を管理型) の通信を提供する．そして，TCP では，3 ウエイ・ハンドシェークと呼ばれる方式で通信を開始し，通信が確立された状態となるまで，クライアントとサーバの通信状態が変化する．以下に，その変化とパケットのやり取りを示す (12参照)．

1. 通信前は，サーバ，クライアントとも通信を終了した CLOSED 状態となっている．

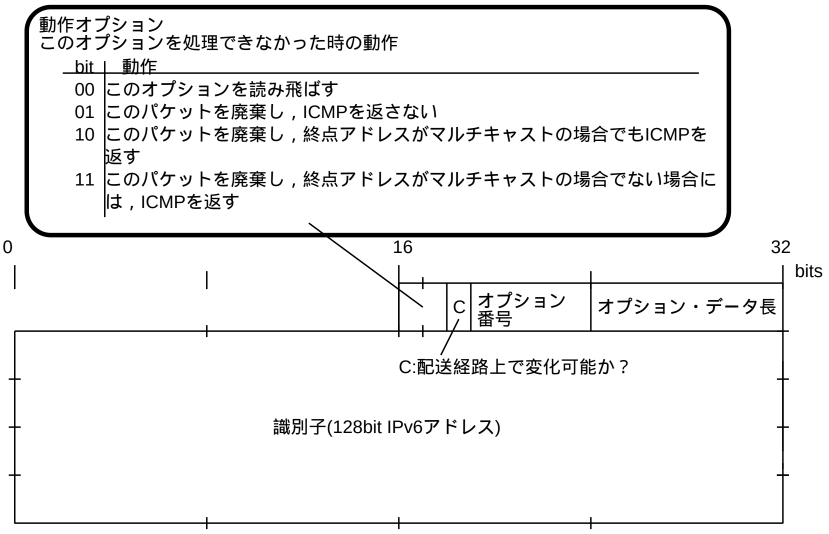


図 11　始点識別オプションのフォーマット

2. 通信を行うために，サーバは，通信を受け付け可能な LISTEN 状態となる
3. クライアントが，サーバに対して，始点アドレス A，終点アドレス B，TCP の SYN(Syncronize, 同期) を送信する (図 12中の (1))．
4. クライアントは，CLOSED から SYN_SENT 状態となる
5. SYN を受信したサーバは，始点アドレス B，終点アドレス A，クライアントの SYN に対する ACK(Acknowrage, 同意) と SYN をクライアントに送信する (図 12中の (2))，
6. サーバは，LISTEN から SYN_RECEIVED 状態となる．
7. SYN+ACK を受信したクライアントは，SYN に対する ACK を返答しする (図 12中の (3))．
8. クライアントは，SYN_SENT から接続の確立した ESTABLISHED 状態となる．

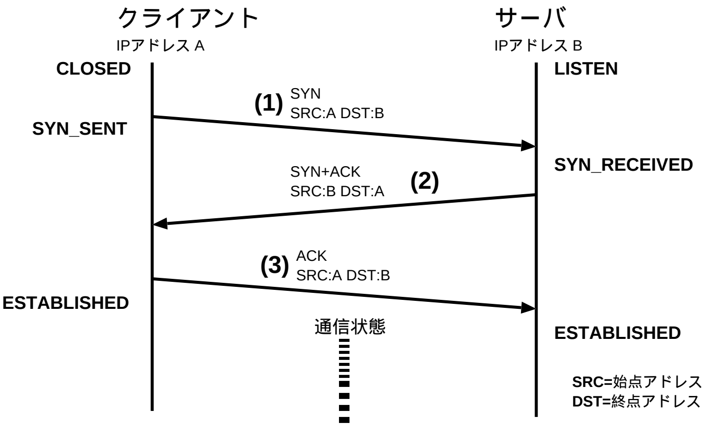


図 12 TCP 3ウエイ・ハンドシェーク

9. クライアントから ACK を受信したサーバの状態は，SYN_RECEIVED から，ESTABLISHED 状態となる．
10. 両者が接続確立状態になり，通信が可能となる．

TCP の場合，「始点識別オプション方式」の処理は次のようになる．
TCP の 3 ウエイ・ハンドシェークに従い，

1. クライアントが始点アドレス A，終点アドレス M へ TCP の SYN を送信する (クライアントは SYN_SENT 状態，図 13中の (1))．
2. この SYN は，エニーキャスト・アドレス M をもつ複数のサーバの 1 つに届く．(サーバは LISTEN 状態，図 13中の (2))
3. サーバは，TCP の SYN の終点アドレスがエニーキャストであることより，始点アドレスを M から B に変更し，クライアントへ始点アドレス B，終点アドレス A，アドレス M を識別子とした始点識別オプションを含んだ終点オプション，TCP の ACK と SYN を送信する (サーバは SYN_RECIVED 状態，図 13中の (3))．

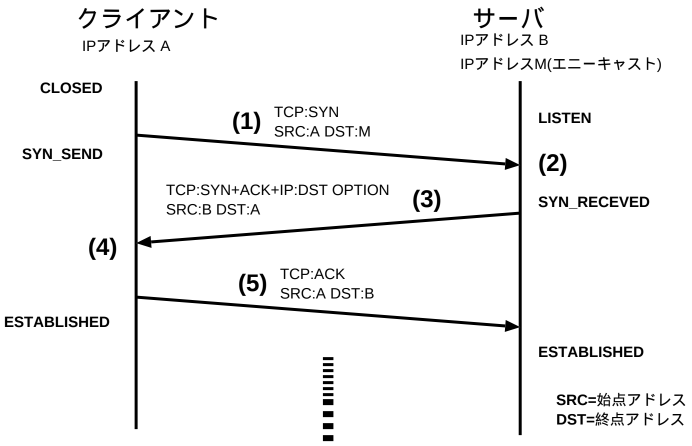


図13　始点識別オプション方式の処理の流れ

4. クライアントは，受信したIPパケット(図13中の(4))の終点オプション中に，アドレスMを識別子とした始点識別オプションが指定されていることから，受信したIPパケットは，アドレスMへ送信したSYNに対する応答であることと，アドレスMがエニーキャスト・アドレスであったことを知る．

5. クライアントは，終点アドレスをMからBに変更して，ACKをサーバに送信する(クライアントは，ESTABLISHED状態，図13中の(5))．

6. サーバがACKを受信した後，両者は，通信可能状態(ESTABLISHED状態)となる．

この一連の大まかな動作ブロックの概念を図14に示す．

図14より，この方式は，エニーキャスト経路問題の解決をネットワーク層とトランスポート層で行うことにより．アプリケーションへの影響を小さくすることができる．

一方，UDPは，上位の層に対して，データの受信確認や再送を行わずに，IPパケットをそのまま転送するコネクションレス型の通信を提供する．

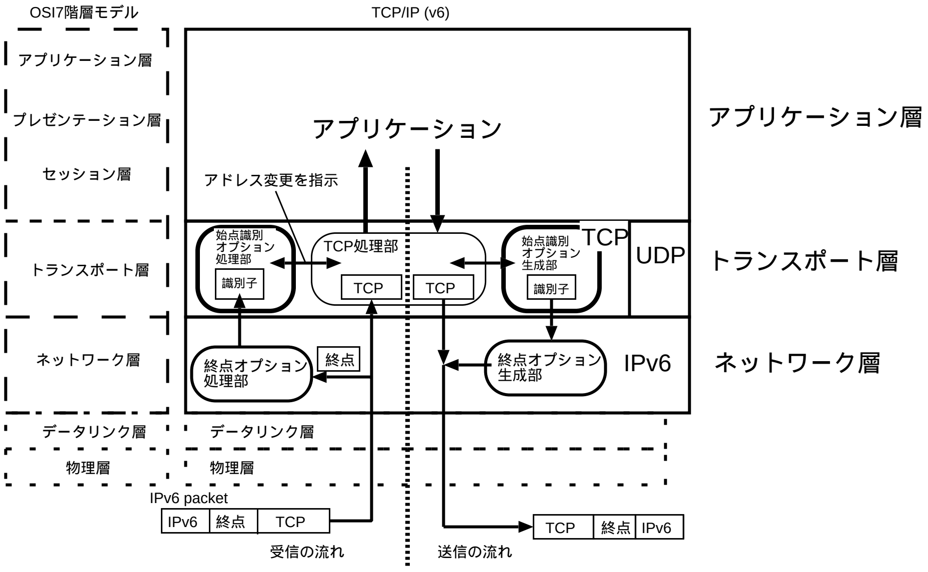


図 14 プロトコル階層単位の動作ブロックの概念 - 始点識別オプション方式 -

よって，UDP は，通信の状態遷移を持たないため，TCP で用いた状態遷移にアドレスの変更を組み込む方式が使えず，アドレス変更の処理が困難である．

この点に関して，J. Bound らは，UDP を利用するアプリケーションのほとんどは，アプリケーションで状態遷移を管理しているので，UDP は，IP からのオプション通知を上位層へ伝達し，アドレスの変更の処理は，アプリケーションに委ねるという考え方を示している．

2.1.2 通信前エコー方式

この節では，エニーキャストの経路問題を解決する方式として，「通信前エコー方式」を提案する．

通信前エコー方式は，始点識別オプション方式と同じく，アドレスを変更することにより問題の解決を行うが，アプリケーション層で問題を解決する点が異なる．

本提案では，アプリケーション層で用いられている ICMPv6(Internet Control Message Protocol for the IPv6，インターネットにおけるエラーの発生などのメッ

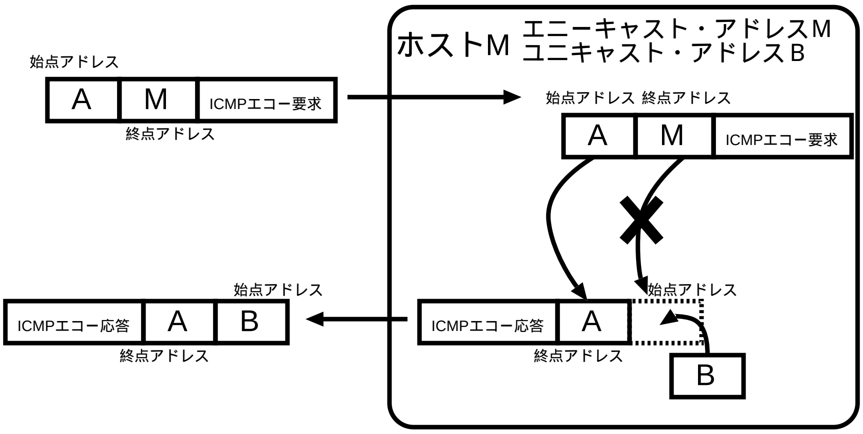


図 15 エニーキャスト・アドレスへの ICMP エコー要求

セージを送受信するプロトコル [11]) のエコー要求と応答を利用する．

IPv6 が稼働しているホストは，ICMP エコー要求に対し，必ずエコー応答しなければならないという通信規約があり [11]，図 15に示すように，ユニキャスト・アドレス A からのエニーキャスト・アドレス M へ ICMP エコー要求に対するエコー応答の終点アドレスには，終点アドレスにエニーキャストを設定することができないので，ユニキャスト・アドレス B を設定し，エコー応答する．

そして，ホスト A は，アドレス M へのエコー要求に対する ICMP エコー応答が，アドレス B から受信したことより「エニーキャスト・アドレス M に対応するユニキャスト・アドレスは B である」ことが分かり，ICMP を利用して，「エニーキャストに対するユニキャスト・アドレスの調査」をすることができる．

以下に，この調査方法を利用し，エニーキャスト・アドレス M への通信を開始するまでの流れを示す．

1. クライアントは，エニーキャスト・アドレスへ ICMP のエコー要求 (図 16 中の (1)) を行い，すべての自分あての ICMP メッセージを受信する状態で待機する．

2. エニーキャスト・アドレスへ送信されたエコー要求は，クライアントから

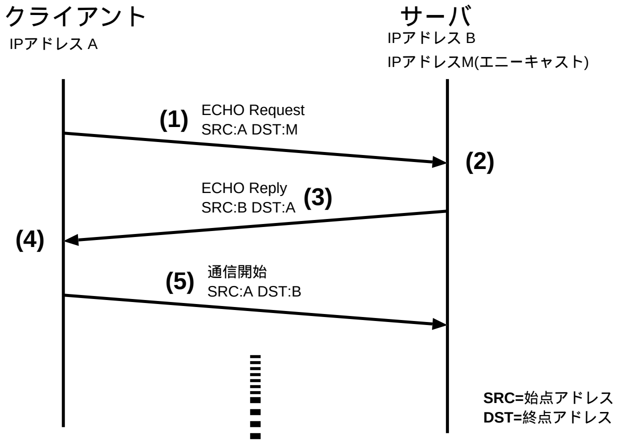


図 16　通信前エコー方式の処理の流れ

経路的に最も近いエニーキャスト・アドレスをもつサーバで受信される (図 16中の (2))．

3. サーバは，エコー要求に対して，終点アドレスにクライアントのユニキャスト・アドレス，始点アドレスにサーバのユニキャスト・アドレスをもつエコー応答を返信する (図 16中の (3))．

4. このエコー応答により，クライアントは，サーバの持つユニキャストアドレスを知ることができる (図 16中の (4))．

5. そして，ユニキャスト・アドレスをもちいて，通信を開始する (図 16中の (5))

以上に示した処理 (エニーキャストからユニキャストへの変換) を，既存のクライアント・アプリケーションに組み込むことで，エニーキャストを利用することができる．

最後に，通信前エコー方式の一連の処理の動作ブロック概念を図 17に示す．

この図から分かるように，この方式は，エニーキャスト経路問題の解決をアプリケーション層で行うことにより，下位層への影響を小さくすることができる．

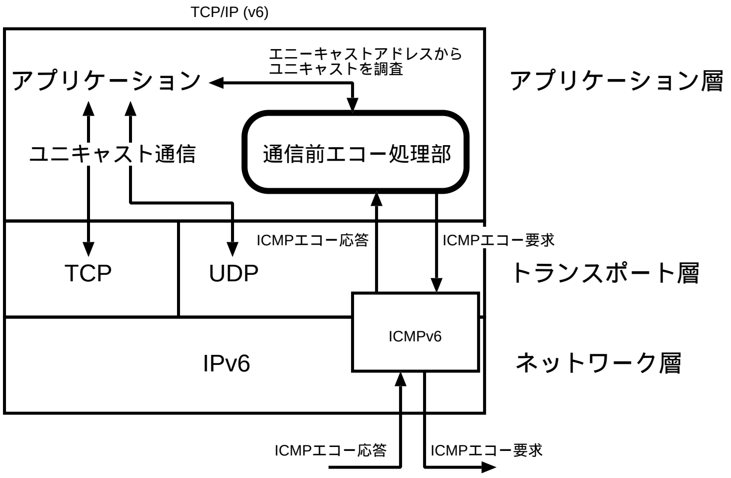


図17 プロトコル階層単位の動作ブロックの概念 - 通信前エコー方式 -

2.1.3 TCPオプション方式

ここでは，エニーキャストの経路問題を解決する方式として，「TCPオプション方式」を提案する．

これは，始点識別オプション方式や通信前エコー方式と同じく，アドレスを変換することによりこの問題の解決を行う．アドレスの変更方式は，始点識別オプション方式とほぼ同じであり，オプションにTCPオプション(図18参照)を使うことが異なるそのため，トランスポート層プロトコルであるTCPのみの拡張となる．

始点識別オプションと同様に，TCPのオプションに，エニーキャスト・アドレスを識別子として組み込む．具体的な処理の流れは，次に示すようになる．

1. 3ウエイ・ハンドシェークに従いクライアントが終点アドレスMへ，始点アドレスA，TCP:SYNとしたIPパケットを送信する(図19中の(1))．
2. このIPパケットは，エニーキャスト・アドレスともつ複数のサーバの1つに届く(図19中の(2))．
3. サーバは，SYNの終点アドレスMがエニーキャストであることより，始点

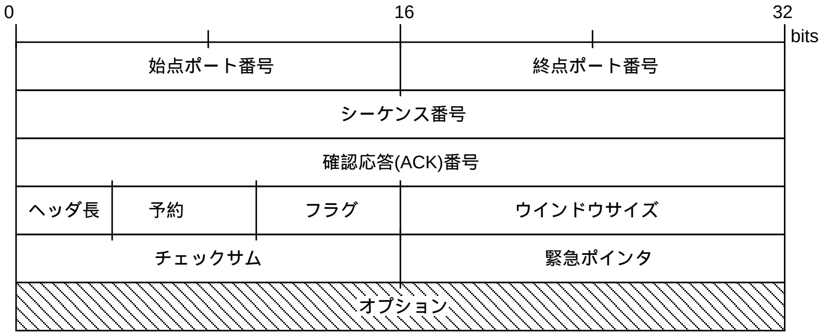


図18 TCPヘッダ・フォーマット

アドレスをMからBに変更し，クライアントAへ，始点アドレスB，終点アドレスA，TCPオプション(エニーキャスト・アドレスM)を付加したACKとSYNを送信する．(図19中の(3))．

4. クライアントは，受信したIPパケットのTCPオプションにエニーキャスト・オプションが付加されオプション中のアドレスMより，受信したIPパケットは，先にアドレスMへ送信したSYNに対する応答であり，また，アドレスMがエニーキャスト・アドレスであったことが分かる(図19中の(4))．
5. クライアントは，終点アドレスMをユニキャスト・アドレスBに変更したTCP:ACKを送信(図19中の(5))する．
6. 両者は通信状態となる．

ところで，同じトランスポート層プロトコルであるUDPには，図20に示す通り，オプションの項目がないために，トランスポート層だけでは，アドレスの変更通知ができない．よって，TCPのようにアドレスを変更することはできない．

最後に，TCPオプション方式の一連の処理の動作ブロックの概念を図21に示す．

この図から分かるように，この方式は，問題の解決をトランスポート層TCPのみで行うことにより，アプリケーション層とネットワーク層への影響を小さく

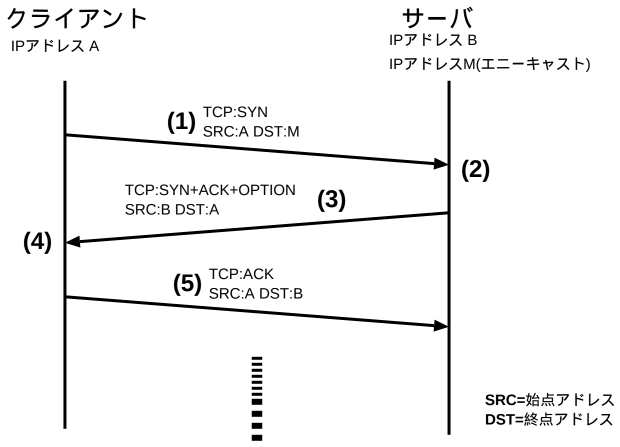
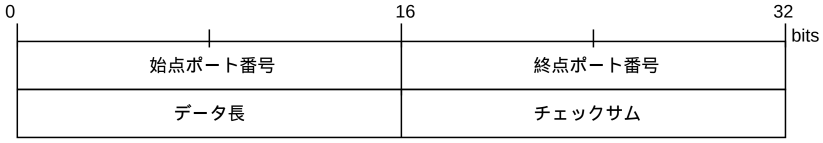


図 19 TCP オプション方式の処理の流れ



始点ポート番号	終点ポート番号
データ長	チェックサム

図 20 UDP ヘッダ・フォーマット

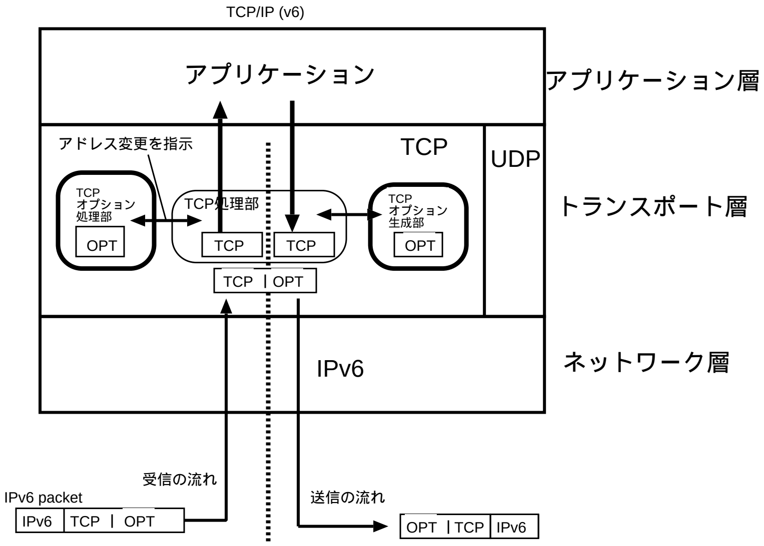


図 21　プロトコル階層単位の動作ブロックの概念 - TCP オプション方式 -

できる．

2.2　各方式の比較と本報告の実装

前節までに，始点識別オプション方式，通信前エコー方式，TCP オプション方式の 3 つの解決方式を示した．そして，プロトコル階層における各方式の実装範囲を図 22に示す．この図より，3 つの方式はエニーキャスト経路問題の解決に対するアプローチが全く異なるものであることが分かる．

次に，各方式の短所と長所をまとめた結果を表 1，表 2に示す．

表 1と表 2より，本論文では，始点識別オプション方式と，通信前エコー方式の実装を行った．TCP オプション方式を今回の実装対象から外した理由は，

- TCP オプション方式と始点識別オプション方式の差は，アドレスの変更のために利用するオプションが存在する層が違うだけなので，TCP オプション方式の実装の結果は，実装範囲の広い始点識別オプション方式の結果に含まれると予測される．

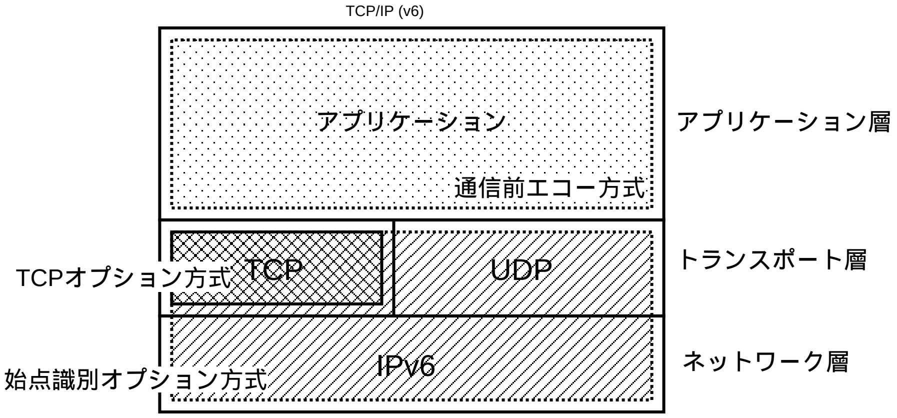


図 22　3 つの方式の実装範囲

表 1　各方式の長所と短所 (1)

比較項目	始点識別オプション	TCP オプション	通信前エコー
問題の解決方式について			
長所	ネットワーク層プロトコル IPv6 のオプションを利用するので，ネットワーク層からアプリケーション層の広い範囲でエニーキャストに対するアドレス変更処理を組み込める．	トランスポート層の TCP オプションを使うので，上位層へ与える影響は，始点識別オプション方式より少ない．	アプリケーション層で処理するので，下位層のプロトコルである TCP/UDP に影響を与えない．
短所	オプションによる挙動を各プロトコル毎に定義しなければならない．	UDP にはオプションの項目がないため，TCP に示した方式によるアドレス変更はできない．	エニーキャストを利用するには，アプリケーション毎に合った変更が必要．

表 2　各方式の長所と短所 (2)

比較項目	始点識別オプション	TCP オプション	通信前エコー
エニーキャスト・アドレスに関する比較			
長所	エニーキャスト用のアドレスを定義する必要がない	エニーキャスト用のアドレスを定義する必要がない	-
短所	-	-	エニーキャスト・アドレスの定義が必要．未定義の場合，接続先のアドレスがエニーキャストかユニキャストか判断できないため，常時，通信前にはエコー要求・応答しなくてはならない．そのため，不要なエコーが発生する．
トランザクション・通信開始までの時間の比較			
長所	トランザクションや通信開始までの時間は，変化しない．	トランザクションや通信開始までの時間は，変化しない．	-
短所	-	-	エコーの往復分のトランザクションが増える．経路の距離に応じてエコーの待ち時間が長くなるため，通信開始までの時間は，始点識別オプション，TCP オプションの場合より長くなる．

- TCPオプション方式は，UDPに対する実装がトランスポート層内で実現できない．

という二点である．

また，残りの2つの方式は，双方ともエニーキャストの経路問題を解決するのだが，始点識別オプション方式の場合は，ネットワーク層とトランスポート層で解決し，通信エコー方式は，アプリケーション層でそれぞれ解決するので，生じる問題も表1,2が示すように対照的である．ゆえに，両者の実装から得られる成果は，かなり違うことが予測される．よって，本論文では，始点識別オプション方式と通信前エコー方式の実装と運用を行い，各方式の実装における問題点や比較，考察を行う．

3. 提案方式の実装

ここでは，第2章の2つの方式の実装について述べる．本論文の実装は，FreeBSD 2.2.8[3]上で動作するkameを拡張する形行った．kameは，kameプロジェクトが実装しているBSDアーキテクチャ上で動作するIPv6プロトコル・スタックである[4]．

実装内容は，識別オプション方式の場合は，IPv6の始点識別オプションの実装と，TCPのアドレス変更機能の実装を行ない，通信前エコー方式の場合は，ICMPエコー要求・応答する機能を，DNSクライアントとHTTPクライアント(WWWブラウザ)に実装する．

3.1 始点識別オプション方式の実装

ここでは，始点識別オプション方式の実装内容を示す．

3.1.1 ネットワーク層の実装

始点識別オプションは，IPv6の終点オプションを利用する．終点オプションは，図23に示すオプション・ヘッダを持っており，オプション領域には，複数のオプションを割り当てることができる．本実装では，この領域の一つとして，表3に示す値を割り当てた始点識別オプションを用いる．

実装部分を図24に示す．図中に示している実装部分の内容は，次の通りである．

- 始点識別オプション処理部

  終点オプションに始点識別オプションが含まれている場合は，オプション中の識別子を「識別子管理変数」へ保存し，mbuf[5]のパケット管理フラグ(pkthdr flags) M_ANYCAST6_DST_OPTを立てる．本実装では，M_ANYCAST6_DST_OPTを定義し，上位層に「始点識別オプション」の有無を伝えている．

---

[3] http://www.freebsd.org/

[4] http://www.kame.net/

[5] BSDでは，データの送受信にmbufと呼ばれるバッファ管理機構を用いる．

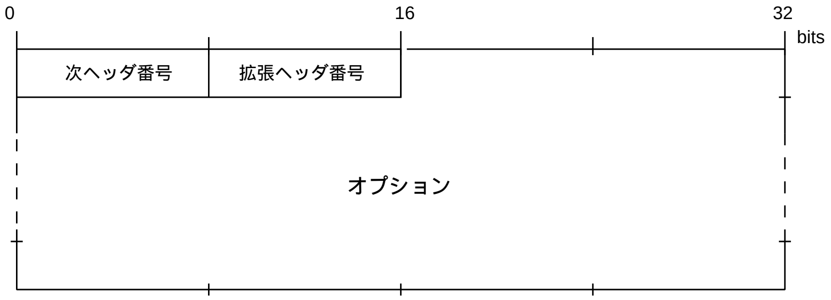


図 23 終点オプション・ヘッダ・フォーマット

表 3 オプション・ヘッダの定義-始点識別オプション-

フィールド名	役割	値	意味
動作オプション	オプションを処理できなかった時の動作	00	読み飛ばす
C	オプションの配送経路上での変化の許可	0	許さない
オプション番号	オプションの識別子	01011	始点識別オプション (空き番号を割り当て)
オプション・データ長	オプション・データ・フォーマットの長さ (オクテット単位)	0x16	
識別子	変更前のIPv6アドレス	-	エニーキャスト・アドレスが入る

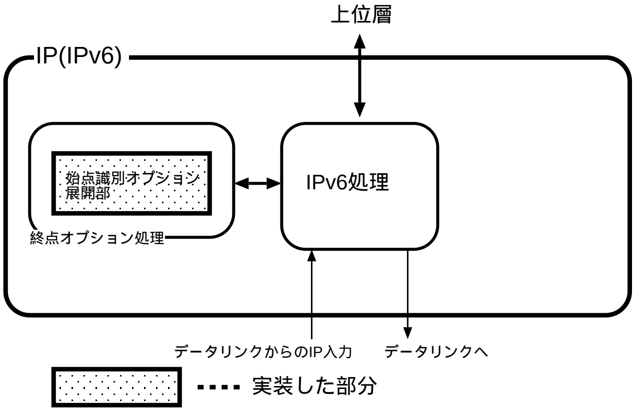


図 24 ネットワーク層の実装部分

3.1.2 トランスポート層の実装

今回の実装では，TCP に対し，2.1.1項で示した手順に従い，実装を行う．実装部分を図 25に示す．図中に示している実装部分の内容は，次の通りである．

- エニーキャスト・アドレスへの接続確認部

  LISTEN 状態で，クライアントからの SYN パケットをエニーキャスト・アドレスで受信した場合，エニーキャスト・アドレスを「変更前アドレス変数」に保存して，ホストのもつユニキャスト・アドレスを，接続管理テーブルへ登録する．

- 始点識別オプション入力処理部

  始点識別オプションのパケット管理フラグ (M_ANYCAST6_DST_OPT) が付加されていた場合，受信した始点アドレスを「識別子管理変数」のアドレスに置き換えて，SYN+ACK に対する SYN を，クライアント接続管理表の中から検索しなおす．発見した場合，接続管理表の始点アドレスを書き換える．見つからなかった場合は，そのパケットを廃棄する．

- 始点識別オプション出力処理部

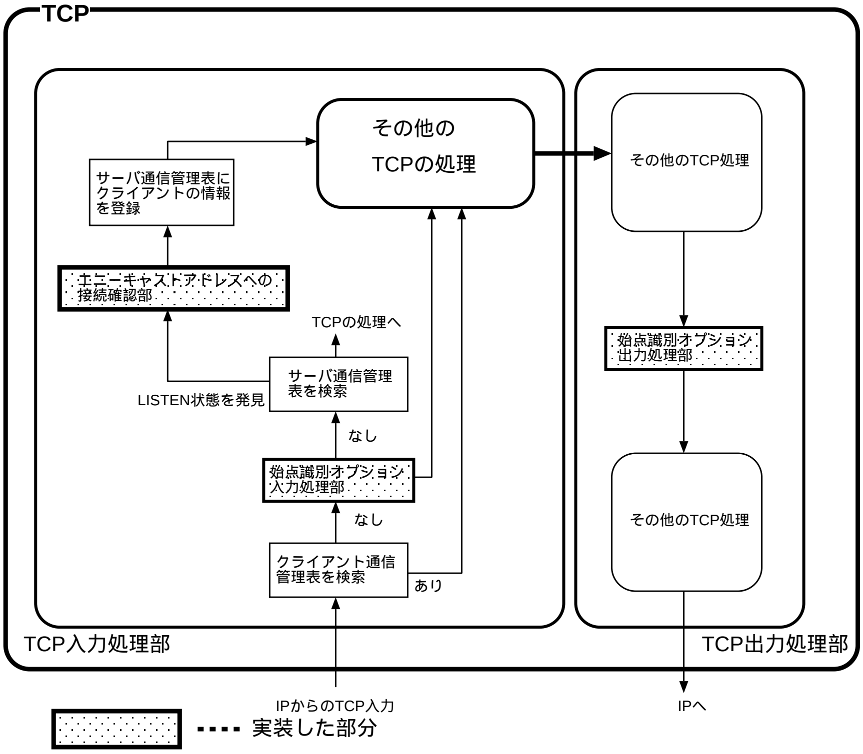


図 25 トランスポート層 TCP の実装部分

変更前アドレス変数が指定され，SYN_SENT 状態であれば，始点識別オプションを終点オプションに付加する[6]．

3.2 通信前エコー方式の実装

本論文では，通信前エコー方式を，TCP の事例として HTTP クライアント，UDP の事例として DNS クライアントの実装を行う．

これは，両者は既に，インターネットにはなくてはならない重要なサービスになっており，エニーキャストの機能である「最も近いサーバへの接続」「サービスを抽象化」「負荷分散」「信頼性」を実験するための最適なアプリケーションであると考えたからである．

3.2.1 HTTP クライアントの実装

IPv4 用の HTTP 代理サーバ ijb[7]を IPv6 への対応と通信前エコー方式の実装を行う．なお，ijb を用いた理由は，図 27に示すように，IPv4 用の HTTP クライアントと IPv6 に対応した HTTP 代理サーバ ijb を組み合わせることにより，IPv6 に対応した HTTP クライアントを実現するためである．この構成を図 26に，動作画面を図 27にそれぞれ示す．

この拡張部分の処理内容は，次のようになっている．

1. ijb は，IPv4 用の HTTP クライアントからの代理接続要求を受け取る．
2. DNS から接続先のホスト名に，IPv6 アドレスが割り当てられている場合は 3. へ，それ以外は，接続先のホストへ IPv4 で接続し，4. へ．
3. 接続先の IPv6 アドレスが，エニーキャストであるかどうかを調べるために，エコー要求・エコー応答の処理を行う．そして，調査した結果の接続先へ IPv6 で接続し，4. へ．

---

[6] 実際には，オプション・データのポインタを IP 層へ引き渡すだけである．IP 層において，そのポインタが IP ヘッダに挿入される．

[7] Internet Junkbuster http://internet.junkbuster.com/

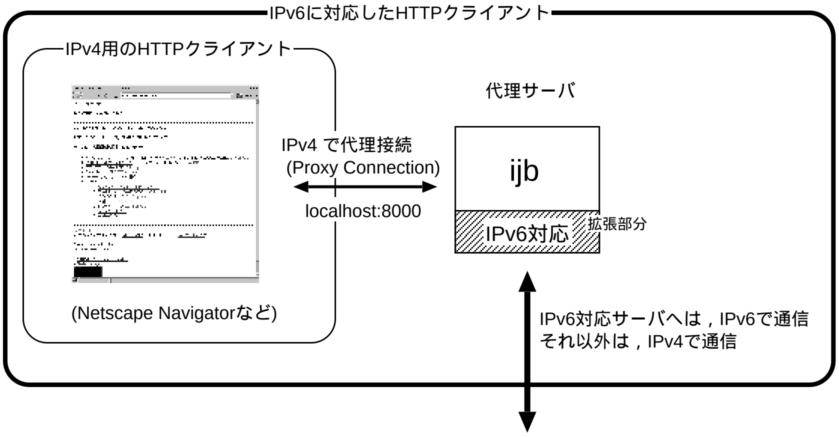


図 26　IPv6 対応 HTTP ブラウザの構成

Who is masa@fumi.org？

Masafumi's Web Space

$Id: index.html,v 1.47 1999/01/31 23:50:03 masa Exp $

日記ともいわれているぐち(VAIO関係の速報?もこちらに)

あなたは、 30224 人目のお客様です。

- VAIO でごーごーなぺーじ Xtt化．tgif4.0.6+vje[update]．完全なる ja-Netscapeの実現01/17 update
- 家庭内自己満足ネットワーク ドリームキャスト接続される1/7 update
- OCNエコノミ導入日記 01/07 update
- PalmIIIはいかが? 07/26 update
- MRTGによるトラフィック解析
- おまけ
    - fumi.orgからの風景(LIVE映像)(故障中)
    - おいしいラーメン&たべもの 10/19 update
    - りんくのページ 10/19 update
    - file倉庫 07/29 update
    - 懐かしいページ 03/12 update
    - ぶつぶつ(日記?)..
    - 私自身のこと

てまえみそ
www.fumi.org/ftp.fumi.orgは、WIDE 6bone に接続しています。(www.v6.fumi.org)

E-mail to masa@fumi.org
MY PGP PUBLIC KEY

- MRTGによるトラフィック解析
- 能楽WebPage

図 27　IPv6 対応 HTTP ブラウザの動作例

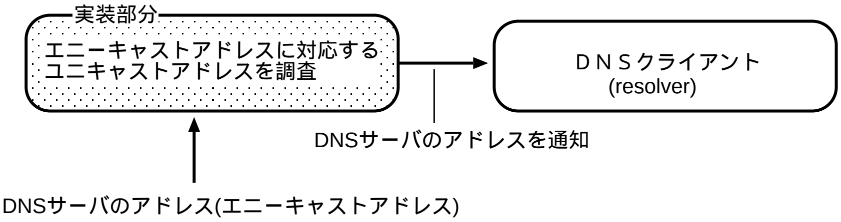


図 28 実装した DNS への拡張部分

4. 代理接続したホストからのデータを HTTP クライアントへ転送し，終了

以上に示した処理により，HTTP クライアントは，エニーキャスト・アドレスへの HTTP サーバとの通信が可能となる．

3.2.2 DNS クライアントの実装

FreeBSD 2.2.8 に実装されている DNS クライアント[8]に対する拡張を行う．実装部分を図 28に示す．

この拡張部分の処理内容は，

- 定期的に，DNS サーバのアドレスとして指定されたエニーキャスト・アドレスに対し，エコーの要求と返答の処理を行うことより，エニーキャスト・アドレスに対するユニキャスト・アドレスを調査
- そのアドレスを DNS クライアントへ通知

の 2 点となっている．

これらの処理により，DNS クライアントは，エニーキャスト・アドレスを付加した DNS サーバの利用が可能となる．また，定期的なエニーキャスト・アドレスに対するユニキャスト・アドレスの調査により，キャッシュ機能を実現している．

---

[8] リゾルバ (resolver) ともいう

4. 評価

本章では，第3章で行った実装の実証実験を行った結果を示し，考察を行う．

なお，実装した始点識別オプション方式のTCPの実証実験のために，ijbによるIPv6に対応したHTTPクライアントにより，通信前エコー方式と同じ実験を行う．

4.1 HTTP代理サーバの運用

ここでは，2台の異なるネットワークのHTTPサーバ[9]に同じエニーキャスト・アドレス設定し，クライアントが，最も近いサーバとの通信が可能かどうかの実験を行う．

図29に，実験のネットワーク構成図を示す．A_1とA_2は，同一のエニーキャスト・アドレスMが割り当てられ，HTTPサーバが稼働しているホスト，R_1，R_2，R_3，R_4はIPv6ルータ，H_1，H_2，H_3は，HTTPクライアントが稼働しているホストである．各ホスト・ルータには，ユニキャスト・アドレスが付加されている．(図29の凡例を参照)

なお，経路制御には，Routing Information Protocol for IPv6 [12] の実装であるroute6d(kameに付属)を用いる．

そして，始点識別オプション方式と通信前エコー方式の両方式について，ホストH_1，H_2，H_3からのエニーキャスト・アドレスMへのHTTPサーバへの通信の可否とパケットの到達したホスト，そして，そのホストは最も近いホストであったかの結果を表4に示す．

また，クライアントとサーバ間の接続確立までのパケットの送受信の記録を始点識別オプション方式は図30に，通信前エコー方式は図31に，それぞれ示す[10]．

表4の結果より，始点識別オプション方式と通信前エコー方式の両実装が，「3つの異なるホストから最も近いサーバに接続することを確認した．

さらに，図30に示す，始点識別オプション方式におけるクライアントとサーバ

---

[9] kame付属のIPv6化差分によってIPv6に対応したapache1.2.6を使用

[10] tcpdumpによる

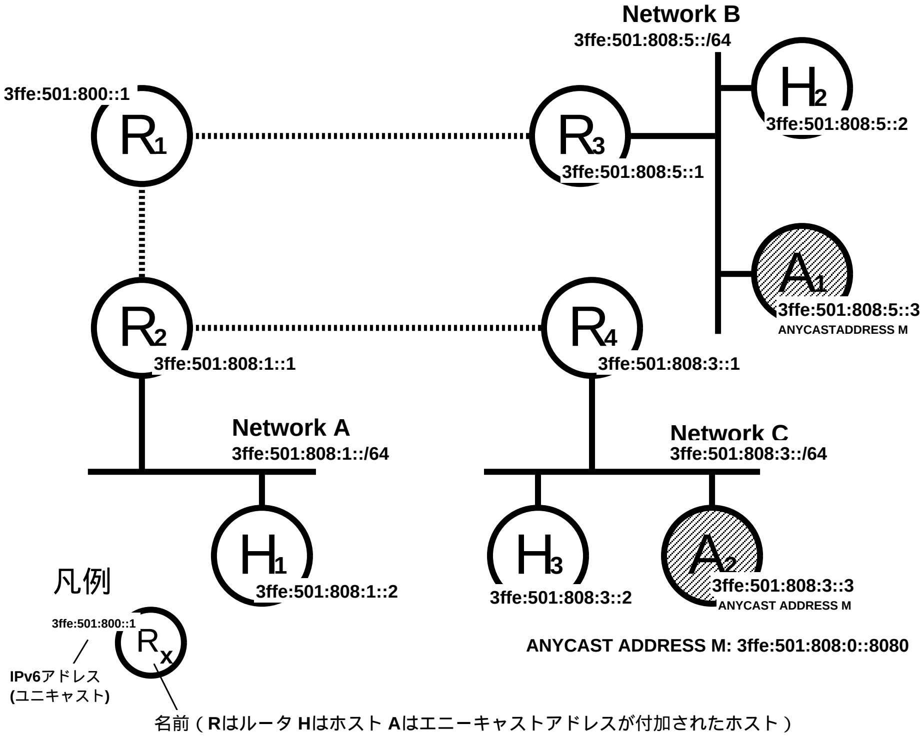
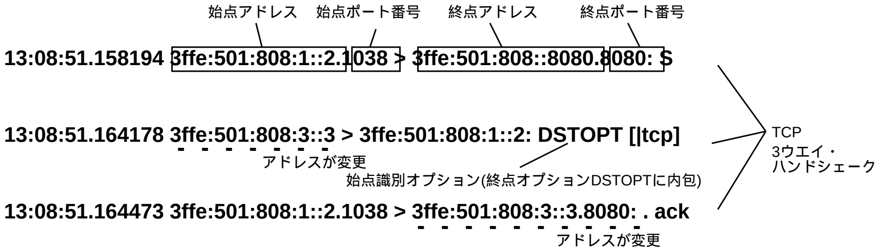


図 29 実験ネットワーク

表 4 HTTP サーバの運用結果

クライアントホスト名	通信の可否		接続先ホスト名	A_xまで距離 (hop 数)		最も近いホストと通信したか？
	識別	エコ		A_1	A_2	
H_1	○	○	A_2	3	4	○
H_2	○	○	A_1	1	5	○
H_3	○	○	A_2	5	1	○

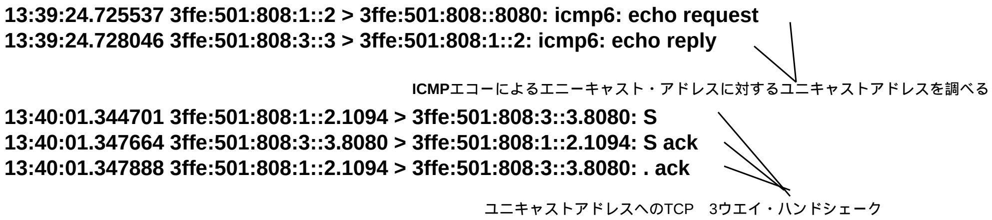


```
13:08:51.158194 3ffe:501:808:1::2.1038 > 3ffe:501:808::8080.8080: S

13:08:51.164178 3ffe:501:808:3::3 > 3ffe:501:808:1::2: DSTOPT [|tcp]

13:08:51.164473 3ffe:501:808:1::2.1038 > 3ffe:501:808:3::3.8080: . ack
```

図 30　接続確立までのパケットの送受信-始点識別オプション方式-



```
13:39:24.725537 3ffe:501:808:1::2 > 3ffe:501:808::8080: icmp6: echo request
13:39:24.728046 3ffe:501:808:3::3 > 3ffe:501:808:1::2: icmp6: echo reply

13:40:01.344701 3ffe:501:808:1::2.1094 > 3ffe:501:808:3::3.8080: S
13:40:01.347664 3ffe:501:808:3::3.8080 > 3ffe:501:808:1::2.1094: S ack
13:40:01.347888 3ffe:501:808:1::2.1094 > 3ffe:501:808:3::3.8080: . ack
```

図 31　接続確立までのパケットの送受信-通信前エコー方式-)

表 5　DNSサーバの運用結果

クライアントホスト名	通信の可否	接続先ホスト名	A_xまで距離(hop数)		最も近いホストと通信したか？
			A_1	A_2	
H_1	○	A_2	3	4	○
H_2	○	A_1	1	5	○
H_3	○	A_2	5	1	○

間のパケットの送受信から，終点オプション(図30中DSTOPT)の始点識別オプションにより，エニーキャストからユニキャストのアドレスへ変更し，接続を確立していることを確認した．

また，通信前エコー方式の場合は，図31に示すパケットの送受信から，ICMPエコー要求と応答により，エニーキャストに対するユニキャスト・アドレスを調べ，そのユニキャスト・アドレスへの接続が確立することを確認した．

4.2 DNSサーバの運用

ここでは，エニーキャスト・アドレスMを異なるネットワークに接続された2台のホストに割り当て，DNSサーバ[11]を運用し，エニーキャストからユニキャストへアドレスが変更がされるか，クライアントが最も近いサーバとの通信が可能かどうか，の確認の実験を行う．

実験ネットワークの構成は，図29と同じである．A_1とA_2は，同一のエニーキャスト・アドレスMが割り当てられ，DNSサーバが稼働している．

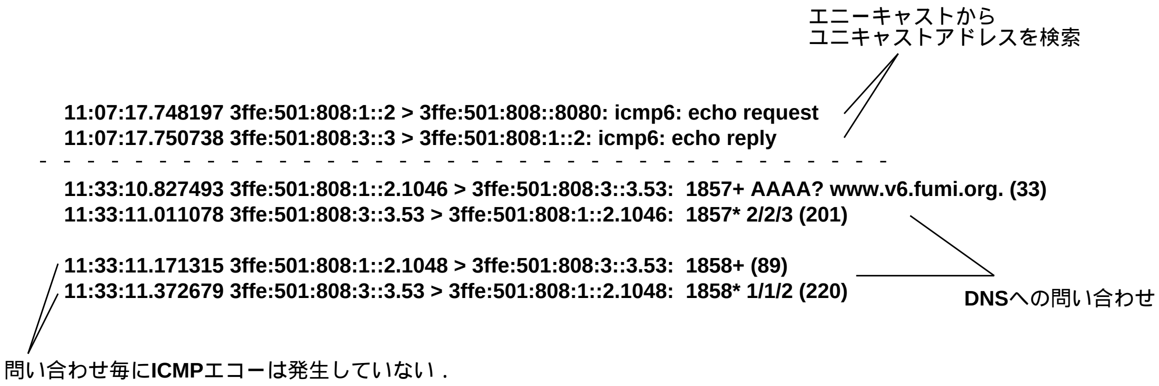
通信前エコー方式について，ホストH_1，H_2，H_3からのDNSサーバへの通信の可否とパケットの到達したホスト，そして，そのホストは経路的に最も近い相手であったかの結果を表5に示す．

また，クライアントとサーバ間のパケット流れを図32に示す．

表5の結果より，3つの異なるホストからおのおのに最も近いサーバへDNSの

---

[11] kame付属のIPv6化差分によってIPv6に対応したbind8を使用



```
                                                                                   エニーキャストから
                                                                                   ユニキャストアドレスを検索

     11:07:17.748197 3ffe:501:808:1::2 > 3ffe:501:808::8080: icmp6: echo request
     11:07:17.750738 3ffe:501:808:3::3 > 3ffe:501:808:1::2: icmp6: echo reply
  - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
     11:33:10.827493 3ffe:501:808:1::2.1046 > 3ffe:501:808:3::3.53:  1857+ AAAA? www.v6.fumi.org. (33)
     11:33:11.011078 3ffe:501:808:3::3.53 > 3ffe:501:808:1::2.1046:  1857* 2/2/3 (201)

     11:33:11.171315 3ffe:501:808:1::2.1048 > 3ffe:501:808:3::3.53:  1858+ (89)
     11:33:11.372679 3ffe:501:808:3::3.53 > 3ffe:501:808:1::2.1048:  1858* 1/1/2 (220)
                                                                                      DNSへの問い合わせ

問い合わせ毎にICMPエコーは発生していない．
```

図 32 クライアントとサーバ間のパケットの送受信 -通信前エコー方式-

問い合わせを行うことを確認した．また，図 32に示すパケットのやり取りから，ICMP エコーにより，エニーキャストに対するユニキャスト・アドレスを調べ (図 32の 1-2 行まで)，DNS への問い合わせは，ユニキャスト・アドレスで行っていること (図 32の 3-6 行目より)，また，キャッシュ機能により，ICMP エコーの回数が減り，全体のトランザクション数が軽減していることが分かった．

4.3 運用結果に関する考察

4.1節や 4.2節の両方式の実験結果は，「始点識別オプション方式」「通信前エコー方式」が，「エニーキャストの経路問題」を解決し，アプリケーションがエニーキャストを使えることを証明した．しかし，実装や運用によって両方式の抱える問題が明確になったので，その問題点についての考察を示す．

4.3.1 始点識別オプション方式に関する考察

始点識別オプション方式は，TCP の状態遷移の中でアドレスの変更を行う．よって，エニーキャスト・アドレスとユニキャスト・アドレスの区別が不可能である以上 (例外は，1.3節で述べた)．アプリケーション層では，その変更を予期できない．ゆえに，以下に示す項目についての考慮が必要である．

1. すべてのアプリケーション・プロトコルの実装は，アドレスが変更される可能性を意識しなければならない．
2. アプリケーション・プロトコル内でアドレスを扱うプロトコルは，プロトコル内でアドレス変更に関する考慮をしなければならない．
3. トランスポート層プロトコルの操作を抽象化した関数群[12]は，アドレス変更に関する考慮をしなければならない．

1. 2. の必要性は，アプリケーションの実装が，「アドレス変更はされない」という前提に基づいて行われているからである．

この前提に基づいたアプリケーションが TCP 接続で，アドレスの変更が発生する場合，以下に示す過程を通して問題が発生する (図 33参照)．

1. クライアントは，宛先アドレス M との接続を TCP に要求
2. TCP は M と接続を試みる，アドレス変更，B と接続状態となる．
3. TCP は，接続完了をクライアントへ通知，(サーバは，接続されたことを通知される)
4. クライアントは，宛先アドレス M と接続しているつもりになる．しかし，実際には宛先アドレス B と接続している．

よって，IPv6 のアプリケーションは，「接続時にはアドレスが変更される可能性がある」ことを考慮した実装を行わなければならない．例えば，エニーキャストによる接続を拒否することや，接続後に相手先を調べること等を行わなければならない．

3. の必要性は，これらの関数群は，エニーキャストを考慮していないためで，「接続時にはアドレスが変更される可能性がある」こと考慮する必要がある．例えば，ソケットの場合，TCP 接続するとき 関数 $connect()$ に相手先を指定したポインタを引き渡すが，アドレスの変更を行った場合の振舞いを考慮しなければならない．

---

[12] UNIX システムにおけるソケット (socket) がそうである

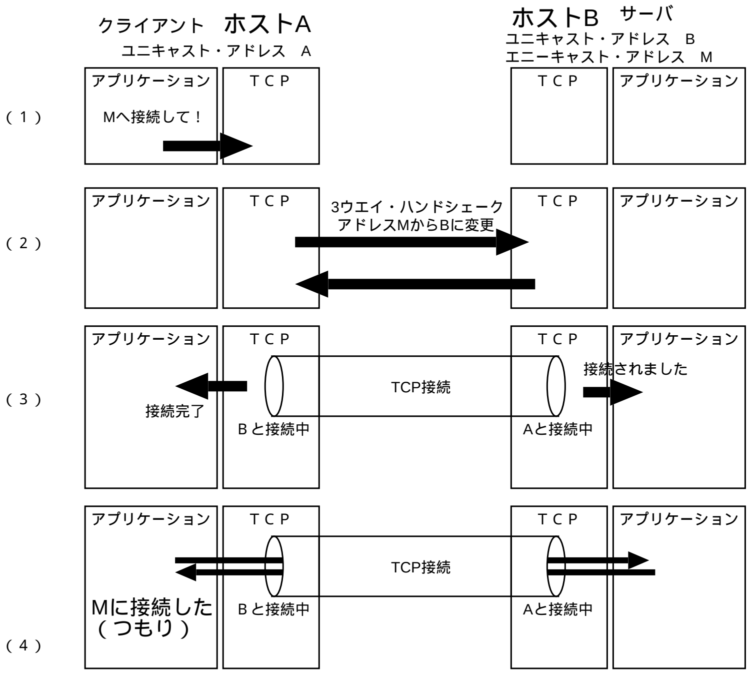


図 33　アプリケーションが TCP 接続を確立するまでの流れ

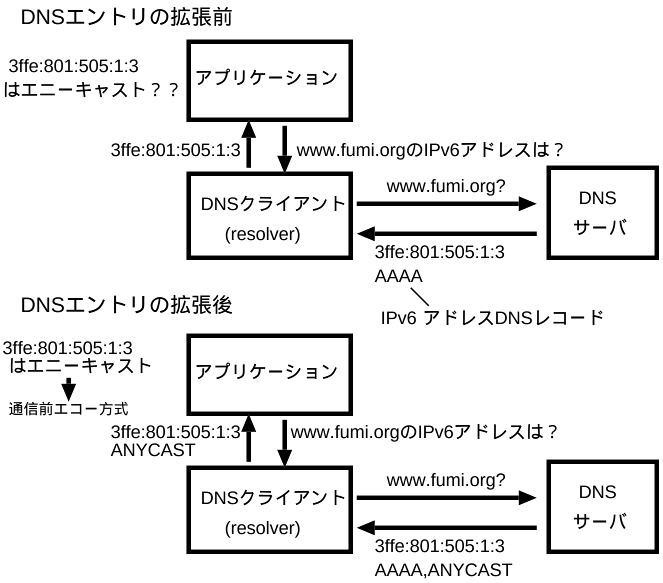


図 34　識別子エニーキャストの提案前と提案後の違い

4.3.2　通信前エコー方式に関する考察

通信前エコー方式は，通信前にエコーを行うため，トランザクション数が増える欠点がある．しかし，今回の実装に取り入れたように，通信毎にエコーするのではなく，エコー結果をキャッシュすれば，トランザクション数を減らすことができる．今後，今回のような規模が小さく，変化のないネットワークではなく，広域ネットワークで実験した結果をもとに，キャッシュの保持時間を考察する必要がある．

また，この方式は，エニーキャスト・アドレスの定義が必要という欠点がある．しかし，これは，DNS と連携することで解決できる．現在，DNS によってより多くの情報を配送する試みがある[13][14][15]．そこで，DNS のリソース・レコードとして，識別子「エニーキャスト」を新たに定義し，エニーキャスト・アドレスにこの識別子を付ける．これにより，クライアントは，DNS で名前を解決すれば，エニーキャストであるかどうかが判断でき，通信前エコー方式が利用できる(図 34参照)．

4.4 提案方式の比較と評価

最後に，「始点識別オプション方式」と「通信前エコー方式」のどちらをIPv6で使うべきなのかについて述べる．

両者の長所・短所をまとめると，

- 始点識別オプション方式は，トランザクションを増やさずにエニーキャストを利用できるという点は技術的に優れており，理想的な方式である．しかし，アドレス変更をトラスポート層で行うということは，トランスポート層への変更以外に，トラスポート層プロトコルの操作を抽象化した関数群やアプリケーションの変更も必要である．このことは，標準化という面からは非常に不利である．
- 通信前エコー方式は，問題として，通信前のトランザクションの増加や，エニーキャスト・アドレスの定義の必要性や，アプリケーション毎の実装の必要性がある．しかし，通信前にアドレスの問題を解決するので，下位層に与える影響がない．また，DNSのリソースの拡張や，アドレスのキャッシュにより，この方式の問題点をある程度解決することができる．

となり、両者は、全く異なった長所と短所をもち、互いに独立した方式であることから、IPv6プロトコル自体の実装に応じて、使い分けるべきであると考察する。

5. おわりに

この章では，本論文での成果をまとめて，今後の課題を述べる．

5.1 まとめ

次世代のインターネット・プロトコルであるIPv6は，通信方式の一つとして「エニーキャスト」を定義している．エニーキャストとは，グループに属するホストの中から最も経路的に近傍のホストとの通信を実現する通信方式で，信頼性，負荷分散，抽象化を提供することができる．

しかし，エニーキャストを用いたサービスを運用する場合，エニーキャストの利用規約から生じる「エニーキャストの経路問題」を解決しなければならない．その解決策の一つとして，J. Boundらが提案した「始点識別オプション方式」が提案されていた．しかし，実装がなく，エニーキャストを用いたサービスの運用はされていなかった．

そこで本論文では，エニーキャストの経路問題の解決策として「通信前エコー方式」と「TCPオプション方式」を新たに提案し，3つの方法の比較と考察を行ない，「始点識別オプション方式」と「通信前エコー方式」を実装し，エニーキャストを用いたサービスの運用として実験した．その結果，両方式がエニーキャストの経路問題の解決策として有効であることが確認できた．

5.2 今後の課題

今回は，機材や時間的制約から，実験で使用したネットワークは，最小の構成であり，得られた結果も十分なものではなかった．そこで，次に示す項目について，今後の課題とした．

- 実験規模の拡大

  エニーキャスト・サービスを日本のIPv6実験ネットワークである6bone-JPに導入し，運用方式や，エニーキャストが与える経路制御機構への影響を調査する[16]．

その結果は，広域ネットワークにおけるエニーキャストが実現できるのかどうかの指針になると考える．

- 通信前エコー方式の問題調査

  通信開始時のトランザクションの増加という問題の規模，アドレスをキャッシュする場合の保持時間，DNS のエントリ拡張との連携の有効性など，実運用を通して明確にしなければならない．

- 始点識別オプション方式の問題調査

  始点識別オプションを使うためには，4.3.1項で述べた，アプリケーション・プロトコルや，その実装，トランスポート層プロトコルの操作を抽象化した関数群の影響を考察しなければならない．そこで，これらの調査を行い，与える影響の規模を具体化し，始点識別オプション方式を利用のための指針を定める．

- セキュリティーの考察

  アドレスの変更に際し，悪意もった第三者が，通信を横取りする可能性がある．これに対する考察は重要であり，早急になんらかの方法を考えなければならない．

- 標準化活動

  研究成果をインターネット・ドラフトに反映させ，標準化活動を積極的に行うことが，エニーキャストの普及の鍵となると考えている．

謝辞

本研究，および本報告書作成における全過程を通じて，懇切なる御指導，御鞭撻を賜わりました本学情報科学研究科の福田 晃教授，湊 小太郎 教授，および，最所 圭三 助教授に厚く感謝の意を表します．

また，本研究を進めるにあたり，有用な議論をして頂いたIIJ技術研究所の山本和彦氏，荻野純一郎氏，本学情報ネットワーク講座の出水法俊氏に感謝します．WIDEプロジェクトIPv6分科会のみなさんやkameプロジェクトの皆さんには，メーリングリストや研究会を通じ，様々な意見を頂き，お礼を申し上げます．

この2年間の研究活動を通して，成長していくIPv6に関われたことは，私にとって，言葉では表せ切れないほどの大きな経験でした．

最後になりましたが，様々な経験と機会を与えてくださったWIDEプロジェクトに深く感謝いたします．

参考文献

[1] J. Postel, “Internet Protocol,” RFC 791 , September 1981.

[2] G. Huston, “Observations on the Management of the Internet Address Space, ” RFC 1774 , December 1994.

[3] S. Deering and R. Hinden, “Internet Protocol Version 6,” RFC 2460 , December 1998.

[4] J. Postel, “Transmission Control Protocol,” RFC 793 , September 1981.

[5] J. Postel, “User Datagram Protocol,” RFC 768 August 1980.

[6] R. Hinden and S. Deering, “Internet Protocol Version 6(IPv6) Addressing Architecture,” RFC 2373 , July 1998.

[7] D. Johnson and S. Deering, “Reserved IPv6 Subnet Anycast Addresses,” draft-ietf-ipngwg-resv-anycast-02.txt , January 1999.

[8] P.V. Mockapetris, “Domain Names - concepts and facilities,” RFC 1034 , November 1987.

[9] P.V. Mockapetris, “Domain Names - implementation and specification,” RFC1035 , November 1987.

[10] J. Bound and P. Roque, “IPv6 Anycasting Service: Minimum requirements for end nodes,” draft-bound-anycast-00.txt(EXPIRED) , August 1996.

[11] A. Conta, S. Deering, “Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6),” RFC 2463 , December 1998.

[12] G. Malkin and R. Minnear, “RIPng for IPv6,” RFC 2080 , January 1997.

[13] C. Farrell and M. Schulze and S. Pleitner and D. Baldoni, “DNS Encoding of Geographical Location,” RFC 1712 , November 1994.

[14] C. Davis, P. Vixie, T. Goodwin, and I. Dickinson, “A Means for Expressing Location Information in the Domain Name System,” RFC 1876 , January 1996.

[15] A. Gulbrandsen and P. Vixie, “A DNS RR for specifying the location of services,” RFC 2052 , October 1996.

[16] M. Oe, S. Sekiya, K. Hirose, Y. Suguru , “An Experiment on Construction and Operation of IPv6 network,” INET’99 (**投稿中**).