PacketX:エンドホストにおける アプリケーション指向 IP パケット制御機構

井上朋哉[†] 安田真悟[†] 高野祐輝[†]
宇多仁[††] 篠田陽一[††]

各家庭への常時接続環境およびモバイルコンピューティングの普及に伴い，利用者が有線 LAN, 無線 LAN,VPN,VLAN 等を用いて物理的、論理的に複数のネットワークに接続された環境でインターネットを利用する機会が増加した．このような環境においては，現在エンドホストで利用されているネットワークの経路制御機構では，効率よく複数のネットワークを選択することが困難である．こういった経路制御機構は，ルータなどで用いられている経路制御モデルの概念を継承しており，あらかじめ宛先 IP アドレスや宛先トランスポートアドレスが決定されていない P2P の様なアプリケーションには，適用することができない．そこで我々は，エンドホストにおいて，複数のネットワークを柔軟に利用可能とするプリケーション指向 IP パケット制御機構である PacketX を提案する．PacketX では，エンドホストのネットワーク制御機構にアプリケーションの識別子を導入し、アプリケーション毎の経路テーブルを保持することにより，従来の経路制御技術では困難であったアプリケーション毎の経路制御を可能とした．本稿では，PacketX のプロトタイプの実装によって得た経験および実装手法をまとめた．

PacketX: An Application Oriented IP Packet Control System for End-Host Networks

Tomoya Inoue ,[†] Shingo Yasuda ,[†] Yuuki Takano ,[†]
Satoshi Uda [††] and Yoichi Shinoda [††]

As permanently-connected environments and mobile services became more widely used, Internet users can connect simultaneously to several networks, such as VLAN-based, wireless, wired and virtual private networks. Therefore the opportunity of utilizing the Internet has significantly increased. However, at present, end-host network control systems have difficulties in efficiently connecting to several networks in the same time. End-host routing control tools are currently based on traffic control models. However many different applications run on the end-host system. Since traffic control models can only do packet-based information routing, application-level service requirements cannot be handled using this model. We propose that the end host should control several networks based on an application-oriented IP-packet system, called PacketX. PacketX, by using application identifiers has the function to manage the routing table on an application basis. In this paper, we present the mechani and a prototype implementation of the PacketX system.

1. はじめに

現在インターネットは，常時接続環境の普及やモバイルコンピューティングの普及に伴い，自宅，会社や街中など，ありとあらゆる環境から利用することが可能となってきた．これにより，利用者を取り巻くネットワーク環境自体も多様化してきた．

インターネットは，単一組織によって運用されるネットワークではなく，複数のネットワークが相互接続されることによって構成されている．また，それらネットワークの通信速度や帯域幅などは，運用する組織によって大きな差があることが多く，運用ポリシや課金方法なども組織毎に違ってくる．インターネット利用者は，無線 LAN，有線 LAN，VPN，VLAN 等を用いて，これら複数のネットワークへ同時に接続することが可能である．しかしながら，ネットワーク毎に運

† 北陸先端科学技術大学院大学 情報科学研究科
School of Information Science
Japan Advanced Institute of Science and Technology
†† 北陸先端科学技術大学院大学 情報科学センター
Center of Information Science
Japan Advanced Institute of Science and Technology

用ポリシ，性能，特徴などが違うため，利用者はその時の状況によって利用するネットワークを効率的に使い分ける必要がある．

TCP/IP はインターネットで標準的に利用されているプロトコルである．ところが，現在の TCP/IP を用いたエンドホストでは，複数のネットワークを効率よく利用・活用することは困難である．従って，複数のネットワークを効率よく利用・活用する技術が必要となってくる．しかし従来の研究では，主に AS 間など，ネットワーク網同士を接続するための経路制御技術に注目した物がほとんどあり，エンドホストのための IP パケット制御技術は少ない．また，あったとしても，それらはルータやサーバのための技術を流用したものであるため，エンドホストの要求を真に満たすような統一的な概念や実装であるとは言い難い．

そこで我々は，プリケーション指向 IP パケット制御機構である PacketX を提案する．PacketX は，エンドホストのユーザが，多くのネットワークを自由かつ安易に使い分けることを可能とする．本論文では，アプリケーション指向 IP パケット制御機構の必要性と，その概要説明を行い，プロトタイプの実装によって得た経験および実装手法をまとめる．

2. 研究の背景

現在，インターネットを構成しているノードは，大きく分けてルータとエンドホストに分けることができる．そのルータやエンドホストは，IP アドレスを持つことで大規模ネットワークを構築している．インターネットの経路制御は，IP ルーティングにより実現されている．インターネット内の全ノードが IP ルーティングテーブルを持ち，経路情報を IP ルーティングテーブルに保持して経路到達性を確保している．この IP ルーティングのメカニズムはルータとエンドホストでその差はほとんどなく，通信に用いられる IP パケットを転送をするかしないかのみが，ルータとエンドホストを分ける違いになっている．IP ルーティングテーブルの各エントリは，宛先 IP アドレスプレフィックスとゲートウェイを組みとしたものとなっている．また，IP ルーティングテーブルには Default gateway と呼ばれる特殊なエントリがあり，これはエントリに無い宛先を持つパケットを中継するためのに用いられる．

ルーティングテーブルの有効性はルータとエンドホストとでは異なる．一般的に，インターネット基幹網で用いられるルータ間では経路制御プロトコル (RIP, OSPF, BGP 等) が動作しており，最適経路情報が自動的にアップデートされている．しかし，エンドホストで経路制御プロトコルが動作していることは稀であり，明示的な宛先 IP アドレスと最適経路の情報を持っている事は少ない．結果として，エンドホストでの経路制御は Default gateway を使用した経路制御を行うことが一般的な解となっている．

この，IP ルーティングテーブルにおける Default gateway エントリは，IPv4 では Dynamic Host Configuration Protocol(DHCP) によって，自動的に経路エントリの設定が行われることが多い．また IPv6 でも，Router Advertisement(RA) 及び，DHCPv6 により自動設定されることが一般的である．これらのプロトコルにより，エンドホストのユーザは経路表に対して特別な設定を加えることなく，ネットワークにアクセスすることが可能となっている．しかし，エンドホストも複数のネットワークに繋がるようなマルチホームの環境ではこれらの自動設定のみでは不十分となる．その原因として，IP ルーティンテーブルは，宛先 IP アドレス対中継ゲートウェイを一意にルーティングするため，複数ネットワークへの接続性を持つエンドホストでは，各ネットワークごとの Default gateway を設定することができない．さらに、VPN 利用時に一部のアプリケーションはローカルネットワークを利用する等のアプリケーション毎の利用ネットワークの切り替え等、のユーザーの要求にも応えるが出来ない．

2.1 既存研究

IP ルーティングテーブルで困難となる経路制御を行うためにパケットフィルタを用いることが可能である．パケットフィルタは ipfilter, pf, ipfw, netfilter/iptables 等と，細かな部分で有する機能は異なるが，パケットヘッダの情報を基にパケットを制御することから，経路制御を行うことも可能となっている．また既存研究としてエンドホストにおけるポリシルーティング機構が提案がされており，トランスポートアドレスベースルーティングの提案として，PISelect[1)]，マルチパスポリシルーティングの提案として，PMPATH[2)] がある．

しかし，これらの経路制御機構はパケットヘッダの情報を用いており，ルータの様なトランジットノードに用いられるトラフィック制御モデルの概念を逸脱しているものではない．トラフィック制御モデルによるポリシルーティングではアプリケーションの識別子として TCP や UDP のポートアドレスや IP アドレスを基に経路選択を行っているため，パケットに記述されている内容でしかフィードバックを得ることができない．従って，あらかじめ宛先 IP アドレスや宛先トランスポートアドレスが決定されていない様な P2P

アプリケーションなどでは，パケットヘッダベースでの処理を行うトラフィック制御モデルを適用した経路制御ツールでは，アプリケーションとパケットの完全な対を作ることができない．つまり、エンドホストにおいてアプリケーション毎に経路制御行いたい要求を完全に満たすためには，パケットに記述さている情報を利用するのでは無く，エンドホストのシステム内で用いられているアプリケーション識別子を用いたネットワーク制御機構が必要となってくる．

3. アプリケーション指向 IP パケット制御機構の提案と設計

エンドホストにおけるアプリケーション毎の経路制御を可能とする IP パケット制御機構 PacketX を提案し，その設計について議論する．

アプリケーション毎に経路制御を行うためには次の 2 つが必要である．

- アプリケーション識別子
- 複数経路情報の保持機構

ネットワーク機構における通信の識別子にはソケットディスクリプタ，トランスポートアドレス，IP アドレス，MAC アドレスなど幾つかの識別子を挙げることができる．アプリケーションは，ソケットディスクリプタを使用することでオペレーティングシステム内部のネットワークの制御機構を利用している．つまり，ネットワーク制御機構から見るとソケットディスクリプタがもっともアプリケーション側にある識別子となっているといえる．

PacketX ではネットワークソケットの情報にプロセス番号を付加することで，アプリケーションとネットワークソケットの対を作り，アプリケーション毎の IP パケット制御を可能とするシステムを構成する．しかし，ネットワーク制御機構の経路制御情報は IP ルーティングテーブルにのみ保持しており，それぞれのアプリケーションの要求を反映させた経路テーブルを保持することができない．そこで，PacketX ではエンドホストのアプリケーションに必要となる経路制御のルールテーブルを保持する機構を提供する．

3.1 PacketX の設計

PacketX は IP ルーティングテーブルと協調動作することで，ユーザやアプリケーションが持つ，ネットワークへの要求に従った経路制御を容易に可能とする IP パケット制御機構を提供する．PacketX は PacketX_rtentry， PacketX_rule と PacketX_allocate の 3 つの機構を持つことで実現される．

PacketX_rtentry はエンドホストのシステムが使用

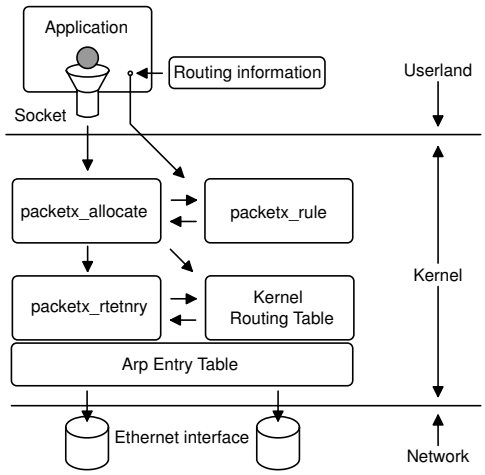

図 1 PacketX I/O 機構
Fig. 1 PacketX I/O scheme

できるネットワークを事前に収集し，ルーティングのエントリを動的構成する．これにより，提供可能なネットワークをユーザに対して示すことが可能となる．PacketX_rule にて，ユーザがどの PacketX_rtentry を使用するかを記述する．これにより，ユーザがどのアプリケーションに対して，どのネットワーク経路を使用するかを指定することが可能となる．また，実際に実行されているアプリケーションが，PacketX_rule にしたがったネットワーク経路を PacketX_rtentry から選び確保することを可能とする機構が PacketX_allocate となっている．

PacketX は柔軟な経路制御機構を提供すると言う特性上，カーネル内部のネットワークスタックに対する設計／実装を必要とするシステムとなる．**図 1** はネットワークの I/O との連携の概要となる．PacketX_rule は，ユーザからの Routing_information を適用することが可能となっている．これにより，IP ルーティングテーブルでは困難であったアプリケーション毎での経路情報の保持が可能となる．そして，Routing_information の情報に従い PacketX_allocate がアプリケーション毎に異なった経路制御ルールを利用して経路制御を行うことができる．この経路制御ルールを用いて PacketX_rtentry から対応する経路情報を利用することができる．

3.2 PacketX_rtentry

図 1 に図解してある通り，PacketX_rtentry の情報は既存の IP ルーティングテーブルの情報を基にして構成されている．その概要を**図 2** に示す．

IP ルーティングテーブルの持つネットワーク情報は RIP や OSPF などから得られるの経路情報や，DHCP

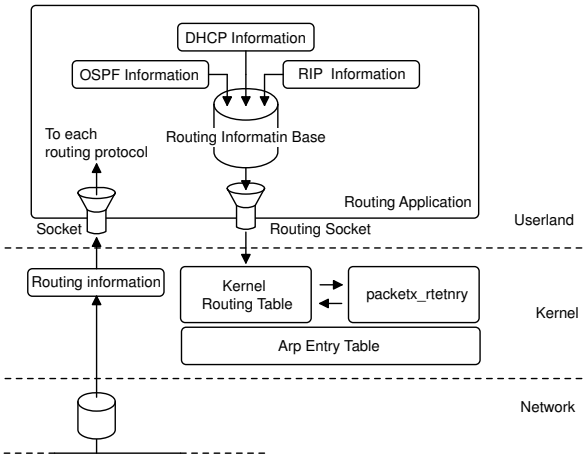

図 2 PacketX rtentry 機構
Fig. 2 PacketX rtentry scheme

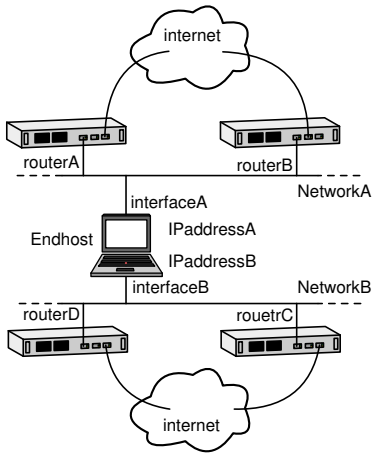

図 3 PacketX サンプルネットワーク
Fig. 3 PacketX sample network

などエンドホスト特有のネットワークコンフィギュレーション情報を保持している可能性がある．しかし，これらの情報は宛先 IP アドレスのみを用いた経路制御となっている．つまり，既存の IP ルーティングテーブルは，経路制御情報と経路情報を同時に持ち合わせている．これは，ネットワーク制御のモジュールとして見た場合には，非常に簡単化されており，扱いやすい機構となっている．しかし，その半面，経路の制御を制限している原因にもなっている．そこで，PacketX では経路情報と経路制御情報を分離して使用することができるように PacketX_rtentry にて経路情報を IP ルーティングテーブルとは別に構成する．

次に PacketX_rtentry をどのように構成するかを議論する．**図 3** のネットワークが構成されていると

表 1 PacketX ルーティングエントリデータ
Table 1 PacketX Routing Entry Data

1	interfaceA	routerA	IPaddressA
2	interfaceA	routerA	IPaddressB
3	interfaceA	routerB	IPaddressA
4	interfaceA	routerB	IPaddressB
5	interfaceB	routerC	IPaddressA
6	interfaceB	routerC	IPaddressB
7	interfaceB	routerD	IPaddressA
8	interfaceB	routerD	IPaddressB

する．Endhost のネットワークとして NetworkA と NetworkB がある．このネットワークのインターフェース A, B に対して，IPaddressA, IPaddressB が割り振られている。また，このネットワークに routerA, routerB と routerC, routerD がある．このネットワーク構成に対して，**表 1** が PacketX_rtentry の経路情報となる．この経路情報はインターフェースに対してそのネットワークが持つネクストホップルータと，エンドホストの持つインターフェースアドレスの組合せによって構成される．結果として，そのエンドホストが持つネットワークに対しての経路情報を全て持つことになる．

この情報は IP ルーティングテーブルより得ることが可能となっており，実装に際してはゲートウェイルーティングフラグの立っている経路情報をピックアップすることで実現される．

3.3 PacletX_rule

PacketX_rule では，ユーザの要求に対してアプリケーション毎の経路制御ルールを構成する．エンドホストの持つネットワーク経路エントリは，ネットワークインターフェースとネクストホップルータ，送信元 IP アドレスの組みにより表すことができる．つまり経路制御ルールとは，このネットワーク経路エントリをどのように使用するのかを決定するルールである．PacketX では，経路制御ルールを PacketX_rule にて制御する．経路制御ルールを構成するに当たり，エンドホストの通信がクライアント側である場合とサーバ側である場合とに分ける必要がある．

3.3.1 クライアント側における経路制御

アプリケーション毎に宛先 IP アドレス対 PacketX_rtetnry リストを指定することでアプリケーション毎の経路制御ルールの反映を可能とする．また，PacketX_rtentry はインターフェース IP アドレスと，ネクストホップルータのネットワークアドレスが一致する経路情報をインターフェースデフォルト経路として保持している．これはそれぞれのネットワークインター

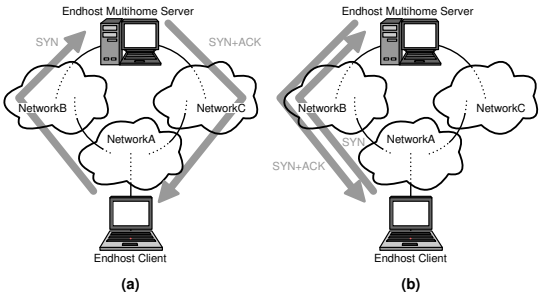

図 4 エンドホストサーバにおけるルーティング
Fig. 4 Routing for Endhost Server

フェース毎に持つ標準経路となる．PacketX_rule では，インターフェースデフォルト経路を利用しての経路選択も可能とする．

3.3.2 サーバ側における経路制御

図 4 にて，エンドホストマルチホームサーバの経路制御の例を示す．図 4(b) は，一般的な経路制御であり，正常な TCP コネクションを成立させることができる．しかし，図 4(a) の様な経路制御が行われたときに TCP コネクションを成立させたい場合は，エンドホストサーバは SYN パケットを受信した NetworkB 側のネットワークインターフェースに割り当てられている IP アドレスを用いて，NetworkC 側のネットワークインターフェースから SYN+ACK を返す必要がある．つまり図 4(a) の様な経路制御では，エンドホストクライアントが要求を行った IP アドレスとは異なった宛先からの応答となるため，正常にコネクションを成立させることができない．エンドホストサーバでは，サービスをどのネットワークへ向けて提供するかを柔軟に制御することが必要となってくる．これよりエンドホストサーバでは次の 2 つのルーティングが必要となる．

- Received Interface based Routing (RIR)
- Received Source_address based Routing (RSR)

RIR では，エンドホストサーバは，通信開始の SYN パケットを受け取ったインターフェースの，デフォルト経路エントリを選択する．次に，RSR では，始めにどのインターフェースによりルーティングを行うかを設定する．さらに，通信サービスの要求が行われた場合，エンドホストクライアントから送られてきたパケットの宛先 IP アドレスを，エンドホストサーバの送信元 IP アドレスとして使用する．つまり，指定したインターフェースを用いてエンドホストマルチホームサーバのコネクションを確立することが可能となる．これにより，既存カーネルにおける IP ルーティングテーブルやパケットフィルタリングツール等では困難であったエンドホストマルチホームサーバでの経路選択／制御が可能となる．

3.4 PacketX_allocate

PacketX_allocate により，既存の IP ルーティングテーブル と協調しながら PacketX を動作させ制御することが可能となる．図 1 に示した通り，PacketX_allocate は PacketX におけるフロントエンドインターフェースを提供する．

PacketX_allocate は，各アプリケーションに対する経路確保を行う．このときに，既存の IP ルーティングテーブルを使用するかどうかを切り分ける必要がある．システム内部ではネットワークソケット毎にその経路を確保しており，初めて利用されるネットワークソケットから送出されるパケットに対して宛先 IP アドレスを基に経路が計算される．そのため，ネットワークソケットはパケット生成過程において，既に経路を確保している場合と経路を確保していない場合に分けられる． 従って，PacketX_allocate では，既に経路が確保されているかをチェックする必要があり，経路が確保されていた場合は経路確保の処理を終える．しかし，経路を確保していなかった場合は，次にその通信がリンクローカルへの通信かどうかを調べることが必要となる．PacketX では経路エントリを PacketX_rtentry にて保持しており，その情報は通信先のネクストホップルータのエントリテーブルとなっている．しかしながら，リンクローカルへの通信においては，エンドホストの端末との直接接続により通信が行われるため，ネットワークポリシによりコントロールを必要としない．つまり，リンクローカルへの通信においては既存の IP ルーティングテーブルを用いる．最後にリンクローカルへの通信ではない場合には，PacketX_rule を参照し PacketX_rtentry の経路情報をアプリケーションに返すことで経路確保が行われる．

4. PacketX の実装

今回提案したアプリケーション指向 IP パケット制御機構 PacketX を NetBSD-2.02 上に実装した．PacketX の主な機構は PacketX_rtentry，PacketX_Rule，PacketX_allocate に分けられるが，これらの機構をカーネル内部に実装した．また PacketX の機構を利用するため幾つかユーザランドアプリケーションの実装変更をおこなった．

4.1 PacketX_rtentry

PacketX_rtentry は IP ルーティングテーブルの情報を使用して構成される． NetBSD で IP ルーティングテーブルは宛先 IP アドレスを基数 2 のツリーとし

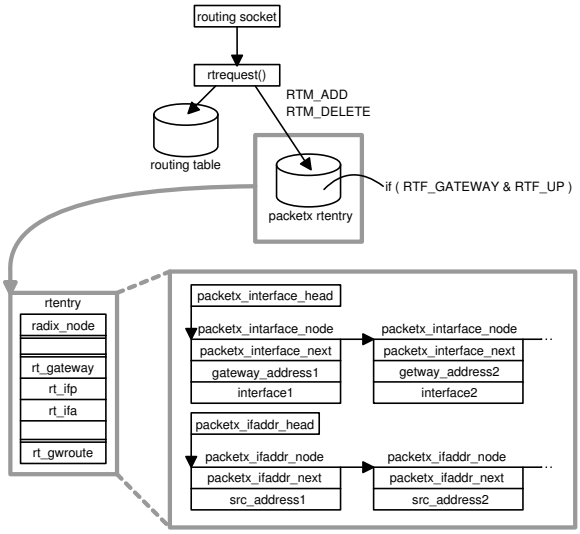

図 5 経路情報の複製
Fig. 5 Replicate routing information

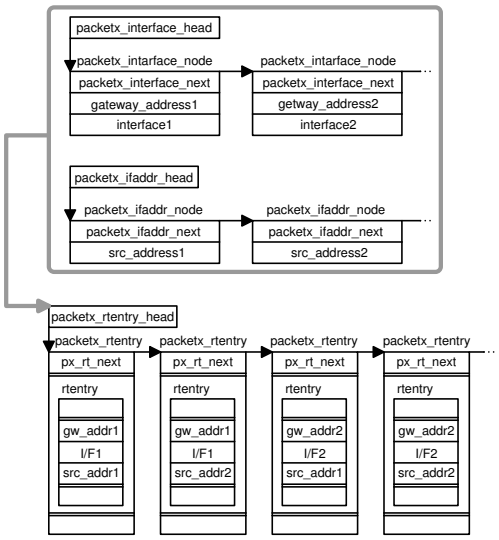

図 6 PacketX_rtentry の生成
Fig. 6 Generate a routing table for PacketX_rtentry

て情報を保持しており，そのデータ構造はラディックスツリーと呼ばれている．このツリーは radix_node と呼ばれる構造体にて構成されている．また，IP ルーティングテーブルは，radix_node 構造体により定義されている rtetnry 構造体により経路情報を保持している．カーネル内のルーティングはこの構造体を指すポインタを得ることで行われている．そこで，PacketX_rtentry では，この rtentry 構造体を利用して経路情報を保持する．PacketX_rtentry を作成するために，PacketX の実装では，IP ルーティングテーブルの経路情報を作成する段階で，ネクストホップ経路を示す RTF_GATEWAY フラグをもつ経路情報をコピーする．これにより，ルーティングを行うために必要となる情報を集めて， PacketX_rtentry を自動的に生成する．**図 5** にその概要を示す．

IP ルーティングテーブルへの経路情報入力は Routing Socket にて行われる．その経路情報を追加するための関数は rtrequest() となっている．この関数内で経路情報を複製し，PacketX_rtentry の経路情報を作成する．複製されたルーティング情報である rtentry 構造体は，インターフェースとネクストホップルータのアドレス情報を対とした packetx_interface データと，システムが持つ IP アドレスデータのリストとなる pakcetx_ifaddr に分けられる．この 2 つのルーティング情報を基に，もう一度 rtentry 構造体を再構成し，packetx_rtentry を構築する．**図 6** に，データを構築する流れを示す．このように PacketX のルーティング情報が作成される．

次に，PacketX_rtentry からの経路情報を選択するためのインターフェースをもつ関数を定義する．

```
void packetx_select_rt(num, ro)
        int num;
        struct route *ro;
```

この関数では，リストされているルーティングデータの番号を入れると，その番号のルーティングデータが ro にリターンされる．またリストされているルーティングデータよりも大きな値になると NULL が返される．

4.2 PacketX_Rule

PacketX には，ユーザ要求に従った経路制御ルールを構築する機構として PacketX_Rule がある．これは各アプリケーション毎に経路制御を行うための多くの情報を管理する．

図 7 にて経路情報ルールテーブルの構造を示す．リストのトップを示すポインタ packetx_rule_head があり，その先に経路制御ルールがリストされることにより，packetx_rule が構成されている．packetx_rule_head にリストされる packetx_rule 構造体は，プロセスグループ ID(PGID) と経路制御ルールテーブルへのポインタが対となった構成となっており，この PGID にてアプリケーションを識別している．このリストにマッチしなかった場合は，通常の IP ルーティングテーブルが使用される．リストにマッチした場合，経路計算を行うアプリケーションがアクティブオープンであるかパッシブオープンであるかを条件に，エンドホストサーバ

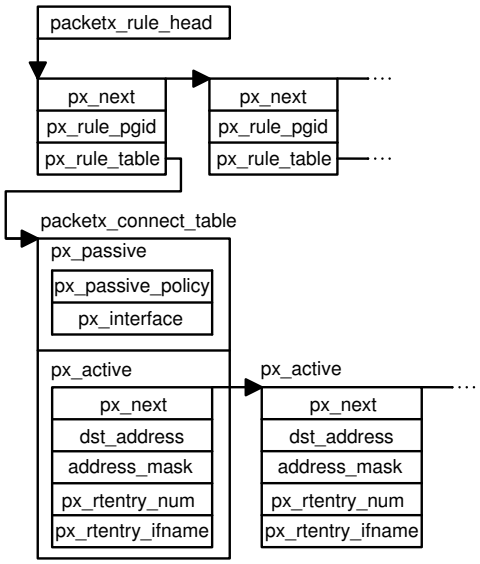

図 7 PacketX ルールテーブル
Fig. 7 PacketX rule table

用の経路制御を使用するか，エンドホストクライアント用の経路制御を使用するかを選択する．エンドホストサーバのアプリケーションに対しての経路制御だった場合，RIR，RSR の条件が packetx_connect_table により指定される．RIR では px_interface にてインターフェース名が記述されている．エンドホストサーバ用のポリシ設定がなされていない場合は px_passive 構造体は初期化状態のままとなっており，この状態では IP ルーティングテーブルが使用される．また，エンドホストクライアントのアプリケーションに対してのポリシルーティングであった場合は，px_active のルールをファーストマッチにて検索される．px_active のルールにはシステムコールの設計上，最後にアプリケーションデフォルトルートが設定されているため，リストの最後 (他のルールが無い場合は最初) はアプリケーションデフォルトルートとして使用される．

4.3 PacketX_allocate

PacketX では経路確保関数 rtalloc に対して，以下のインターフェースを持つラッパー関数を定義する．

```
void packetx_rtalloc(ro)
        struct route *ro;
```

packetx_rtalloc 関数は，PacketX_Rule の情報を参照することで，PacketX を用いた経路制御か既存の IP ルーティングテーブルを用いた経路制御が必要かを切り分ける．

PacketX で経路制御を行うには，カーネル内のいくつかの情報が必要となる．これを行う機構が PacketX_allocate となる．PacketX_allocate は socket 構造体と inpcb 構造体，また場合によっては tcpcb 構造体への情報を PacketX の機構へ提供するために動作する．まず，PacketX_allocate では，socket 構造体に対して，どのアプリケーションによって作成されたソケットであるか識別するためのエントリである pgid を作成する．次に，システムコールである socket 関数に対して，ソケット利用時に PGID が記録されるように変更する．これにより，既存のアプリケーションを変更すること無く使用することが可能となっている．つまり，ネットワーク制御機構からみたアプリケーションを見分けるための識別子を加えることができる．

エンドホストサーバとしてサービスを提供している場合に経路制御を行うときは，カーネル内部のいくつかの情報を必要とする．まず TCP サーバとして動作している時は，SYN+ACK にてコネクション確立の応答を行うときに経路計算が行なわれる．TCP ではコネクションを確立するためにエンド・エンド間にてコネクションの情報を共有する必要がある．従って，エンドホストサーバがマルチホームを行っている場合，コネクションを確立することできない場合がある．PacketX では SYN を受け取った時に，その受信ソケットは既にポリシ管理されているアプリケーションであるかどうかの判断を行う．サーバアプリケーションが PacketX を用いてポリシルーティングを行っていれば，SYN の情報を基に inpcb 構造体の宛先 IP アドレス，宛先ポートアドレス，送信元 IP アドレス，送信元ポートアドレスのエントリを tcp_input 関数の中で満たしてしまう．その後，SYN+ACK にてルーティングを行う時，コネクションの状態が SYN_RCVD の状態であった場合に，その情報を基にポリシルーティングを行う．つまり，コネクションの状態管理することによりエンドホストサーバのポリシルーティングが可能となる．

4.4 複数インターフェースと複数経路の保持

PacketX は経路情報の管理収集を IP ルーティングテーブルの機能を利用することにより，エンドホストネットワークの経路を確保することを可能としている．だが，NetBSD の IP ルーティングテーブルは，同一宛先への経路情報を保持することができないと言う問題がある．つまり，IP ルーティングテーブルはデフォルトルートを複数持つことができない．しかし，エンドホストの環境での経路制御の多くは，IP ルーティングテーブルによるデフォルトルートにより実現されている． 結果として，NetBSD に実装されてい

る IP ルーティングテーブルを用いて経路情報を収集しても PacketX_rtentry にエンドホストネットワークの情報を保持することができないことになる．つまり，PacketX はその機能を十分に生かすことができないことになる．これを改善するために既存の IP ルーティングテーブルに対して，複数の同一コスト経路エントリを持つ事ができ KameTool として提供さている Radix-MPath[3] の技術を用いる．

Radix-Mpath は，同等コストマルチパスルーティングをサポートするために Kame Project によって作られた NetBSD の拡張パッチである．これは，RFC2991(Multipath Issues in Unicast and Multicast Next-Hop Selection)[4] と RFC2992(Analysis of an Equal-Cost Multi-Path Algorithm)[5] を実現するために設計された．これにより，IPv4 と IPv6 のルーティングテーブルに対して同一宛先アドレスに対して複数のネクストホップルータを指定することが可能となる．Radix-Mpath の経路選択の方法は，既存の IP ルーティングテーブル同じであり，宛先 IP アドレスを指定することで，経路計算が行われる．つまり，Radix-Mpath のみではその機能を最大限発揮することはできない．ここで，PacketX を用いることで，PacketX_rtentry に Radix-Mpath にて確保することができる複数のデフォルトルートをを用いることができるつまり，エンドホストにおける動的経路保持を完全に行うことができるようになり，アプリケーション毎にエンドホストネットワークを選択が可能となる．

また，IP ルーティングテーブルに対して，変更を加えずに PacketX_rtentry を加えるために PacketX では，ネクストホップルータを追加する必要がある．そこで，IP ルーティングテーブルより得ることのできない経路情報の収集管理を行うために，以下の追加と削除のコマンドインターフェースを提供する．

```
#packetx_router_addition xx.xx.xx.xx
#packetx_router_delete xx.xx.xx.xx
```

追加コマンドはシステムが持っている IP アドレスのネットワークアドレスと一致しない場合は追加することを不可能としており，削除コマンドにおいては，追加コマンドにて追加したネクストホップルータ以外を削除することを不可能としている．

4.5 アプリケーションによる PacketX の利用

PacketX ではユーザランドコマンドを提供することで，経路情報のルールを記述したファイルを読み込ませ，実行するアプリケーションの経路指定を行う．ルールファイルは，IP アドレスとネットマスクの対より宛先ネットワークアドレスを指定し，宛先ネットワークアドレスに対する PacketX_rtentry を記述する．

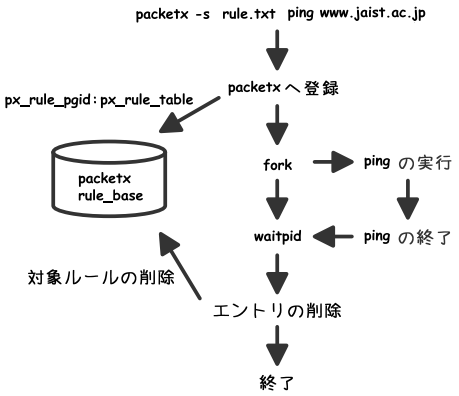

図 8 PacketX 利用
Fig. 8 Utilization of PacketX

PacketX を利用した実行の流れを **図 8** に示す．PacketX のユーザランドコマンドに対して，経路情報のルールファイルを付加してアプリケーションを実行すると，PacketX のルール管理を行うプロセスと，実行アプリケーションのプロセスとに分かれる．ルール管理を行うプロセスは，アプリケーションの終了を待ち，アプリケーションの終了後にルールを削除する．PacketX のユーザランドコマンドにおいて，ルールファイルの他に，直接 PacketX_rtentry を指定したり，インターフェースデフォルトルートを指定することも可能となっている．

5. 関連研究との比較・評価

本稿では，エンドホストにおける，経路制御は複数ネットワークをアプリケーション毎に使い分けることを可能とするネットワーク制御機構 PacketX を提案し実装した．そこで，先に述べた既存技術，関連研究との比較を行うことによりその有用性を検討する．各検討項目および各技術における適応の可否を**表 2** に示す．

アプリケーション毎の経路制御は，既存のトラフィック制御モデル基づく関連技術や研究においては要求することはできない．しかし，本稿にて提案を行った PacketX は，アプリケーション毎にその送出先インターフェース，送信元アドレス，ネクストホップルータを指定した経路制御が可能である．

動的な経路情報の収集において，インターネットは IP ルーティングテーブルにより，経路情報の収集管理と経路の選択制御を行っている．この IP ルーティング

表 2 経路制御技術の比較表
Table 2 compare

	IP Routing Table	Packet Filter	PISelect	PMPATH	PacketX
アプリケーション毎の経路制御	×	×	×	×	○
動的な経路情報の収集管理	○	×	×	×	○
同一宛先 IP アドレス経路の選択制御	×	○	○	○	○
経路計算の量	○	×	○	○	○

テーブルを中心に多くの IP 制御プロトコルが完備されている．PacketX ではこの IP ルーティングテーブルより経路情報を構築するため，RIP, OSPF, DHCP, RA などの経路制御プロトコルの情報にも対応することが可能となってている．

同一ホスト経路への選択制御については，PISelect や PMPATH では単一のルールテーブルしかもっておらず，同一宛先に対して複数のアプリケーションを切り分けることが困難となっている．しかし，PacketX では，複数の経路制御ルールテーブルを持つことができるため，同一ホストへのネットワーク経路の選択制御を可能とする．

パケットフィルタによるポリシルーティングでは，IP ルーティングテーブルにて経路制御を行ったパケットに対して，再びパケットフィルタのポリシによるパケットハンドリングを行うため，不用意な経路計算を行っている．この点から見ると PISelect や PMPATH, PacketX は IP ルーティングテーブルと切り分けて経路制御を行うため，オペレーティングシステム内部にてスマートなルーティング機構を構築している．

6. おわりに

本稿にて提案を行った PacketX は，アプリケーション毎にその送出先インターフェース，送信元アドレス，ネクストホップルータを指定した経路制御が可能となる．現在のネットワークの制御には，多くの IP 制御用のプロトコルとの連携動作が必要となる．また，ネットワーク環境の構成変化やシステム設定の変更に対して柔軟に対応することは，これからのネットワーク制御機構には必須項目であると考えらる．PacketX では IP ルーティングテーブルと協調動作することで，多くのネットワーク環境の情報を特殊な設定を行うこと無く得ることができ，既存の IP 制御用プロトコル群を使用することも可能となっている．また，複数のデフォルトルータの情報を保持することができ，IP ルーティングテーブル以上にネットワーク情報への柔軟な対応が可能となる． これより，PacketX では，ネットワークの環境変化やシステム設定の変更を柔軟に追従することが可能となり，ネットワーク制御機構として，体系的に扱い易いアプリケーション指向の IP パケット制御機構であると言える．

参 考 文 献

1) Minoru KINOSHITA, Kenichi Chinen, Y. S.: The Interface Selection According to User Policy, *WIT2004.JSSST*, pp. 99–104 (2004).
2) Yasuyuki TANAKA, Mitsunobu KUNISHI, F. T.: PMPATH:A Policy Routing System for Multihomed End-Hosts, *Trans. IEICE*, Vol. E89, pp. 219–227 (2006).
3) KameProjcet: http://www.kame.net.
4) Thaler, D. and Hopps, C.: Multipath Issues in Unicast and Multicast Next-Hop Selection, RFC2991 (2000).
5) Hopps, C.: Analysis of an Equal-Cost Multi-Path Algorithm, RFC2992 (2000).