複数経路を用いたIPデータグラムの配送方法とその性能評価

2002MT053 元井 郁　　2002MT095 柳田 陽平
指導教員　後藤 邦夫

1　はじめに

インターネットのトラフィックは近年増加傾向にあり，今後も増え続けるという懸念がある．単一経路によるデータ転送では，帯域幅が足りず望ましいスループットが得られないばかりか，経路に負荷がかかり，他のトラフィックに対し遅延やロスなどの悪影響を及ぼす．データ転送を複数経路へ分散することによりデータ転送に十分な合計帯域幅を確保し，単一経路にかかる負荷の軽減を図る研究がある．これまでの研究は，主にデータ振り分けアルゴリズムの考案，考察 [1][2][3]，トラフィック制御であり [1][4]，これらはトランスポート層の TCP での実装，研究である．UDP を用いた研究はあまり行われておらず，また，複数の経路が利用できることを前提としているため，データ伝送方式の考案はなされていない．

本研究では未解決であるデータ伝送方式を考案，実装し，IP データグラムを Divert Socket，Raw Socket を用いて伝送する．実験ネットワークで TCP，UDP のスループット特性を計測することによって，複数経路の実用性を評価する．なお，計測結果を用いたデータ振り分けアルゴリズムの考案，比較評価は，本年度の修士課程家田により行われる．

伝送方法の考案は共同で行い，元井は主にスループット測定と解析を，柳田は主に実験環境の用意 (設定) とプログラム作成を担当した．

2　ネットワークモデルと実験環境

本研究のネットワークモデルと実験環境について述べる．

2.1　ネットワークモデル

本研究ではデータ送信側が複数の ISP と接続する場合を想定する．複数経路を用いたデータ転送は各 ISP へデータを振り分けることにより実現を図る．各 ISP 内はルーティングプロトコルに従った単一の経路で伝送されるが，ネットワーク全体では複数経路を用いたデータ伝送となる．ネットワーク接続環境は送信側，受信側で異なる．送信側は，

- (A) PI アドレス[*1]を用いて各 ISP と接続する場合
- (B) 各 ISP から割り当てられた異なる IP アドレスを用いて各 ISP と接続する場合

を想定し，受信側の環境は，

- (a) 1 つの ISP と接続する場合
- (b) 複数の ISP と接続する場合

を想定する．送信側にゲートウェイ (以下，GW とする) を設置しデータを振り分ける．送信側，受信側の環境によっては NAT 機能を備えた GW を設置する．

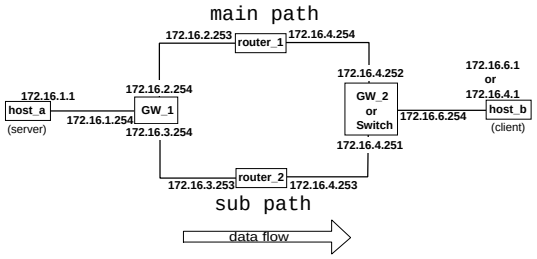

図 1　実験環境

2.2　実験環境

本研究では，ホスト (host_a，host_b) 間に 2 つの経路を持つネットワークで実験する．実験ネットワークは図 1 のようになり，

- 1 台のゲートウェイ (GW_1)，2 台のルータ (router_1，router_2)，スイッチ
- 2 台のゲートウェイ (GW_1，GW_2)，2 台のルータ (router_1，router_2)

の 2 通りの構成がある．全ての PC には Vine Linux を用いるが，kernel 2.4.22 に Divert 機能を追加し，iptables コマンドで Divert 機能が使えるものに変更する．GW_1 ではパケットを複数経路へ振り分ける機能と NAT 機能を，GW_2 には NAT 機能を実装する．GW は 3 つの NIC を搭載し，ホストと 2 台のルータに接続されている．各ルータは NIST Net[5] を搭載し各経路に模倣障害を起こし疑似ネットワークを構成する．データ伝送には，host_b から host_a では単一の経路，host_a から host_b では複数経路を用いる．ホスト間の通信において router_1 側を主経路，router_2 側を副経路とする．また本研究で用いる経路の帯域幅は ADSL などの一般家庭で用いられている 512k，1.5M，8M bits/sec の 3 種類とする．host_a と GW_1，host_b と GW_2，スイッチ間は 100M bits/sec とし，それ以外は上記のいずれかの帯域を NISTNet を用いて模倣する．

GW の機能

GW には複数経路へパケットを振り分ける機能と NAT 機能を環境に応じて実装する．異なる ISP を用いて送受信したパケットのアドレスを変換し正しく送受信できるようにするために NAT を実装する．NAT につ

[*1] PI アドレスとは ISP から独立したグローバル IP アドレスのことで JPNIC が管理する．

いての説明および NAT が必要な環境については第 4 節で詳しく述べる．

3　複数経路へのデータ振り分け方法

複数経路へのデータ振り分けに用いる技法と実現方法について述べる．

3.1　データ振り分けに用いる技法

iptables，Divert Socket と Raw Socket

iptables は Linux カーネルで動作する IP パケットフィルタの設定・管理・検査をするためのツールである．Linux に Divert カーネルを組み込むことによってパケットの横取り (Divert) 処理を可能とした．本研究では FORWARD に対してパケットを横取り (Divert) する．

Divert Socket は横取りしたパケットを iptables で指定した Divert ポートによりプログラムに取り込むソケットである．横取りしたパケットは再注入されるまで送信，受信，転送されない．また転送ではルーティングテーブル参照前に横取りする．再注入後は通常通りのルーティングがされる．

Raw Socket は Raw パケットを送受信するためのソケットである．

3.2　データ振り分けの実現

iptables，Divert Socket，Raw Socket の 3 つを用いた振り分けの手順を述べる．手順は以下の通り．

1. パケットを横取り (Divert) する．
2. 横取りしたパケットをプログラムへ取り込む．
3. Divert Socket で送信 (再注入) をせず，パケットを Raw Socket へ渡す．
4. 次のノードへ振り分けをする．

(1)，(2) は iptables でパケットの横取りをし，プログラムに取り込む処理をする．iptables の FORWARD で横取りしたパケットをプログラムへ取り込む．(3)，(4) ではパケットの振り分け処理をする．Divert Socket で取り込んだパケットを Raw Socket へ渡して各経路へ振り分ける．sendto() 関数では第 5 引数に次のノードの IP アドレスを指定することにより強制的に送信する．結果，ルーティングテーブルを無視した転送が可能となる．NAT する場合は IP ヘッダの IP アドレスを変更し，パケットを送信する．以上の手順で複数経路へのパケット振り分けを実現した (図 2 参照)．

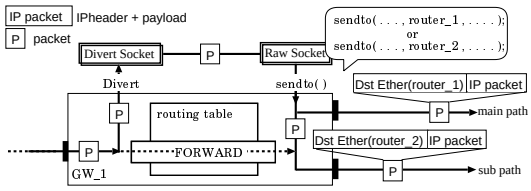

図 2　Divert Socket を利用したパケット分配

4　NAT を用いたデータ振り分けの実現

本研究における NAT とは，パケットを複数の経路へ送信するさいに SrcIP や DstIP アドレスを他の IP アドレスへ変換し，受信側でもとの IP アドレスに戻すことである．そして NAT によって複数経路へのパケット転送を可能とする．

自動的に NAT を開始するにはパケットの流れを感知し，送信側と受信側 GW 間で NAT を行うための情報をデータパケットの転送と並行して送受信しなければならない (以下 NAT セッションとする)．NAT セッションを確立するプログラムは図 1 の GW で動作する．GW では Raw Socket を用いてパケットを送信するので，SrcIP，DstIP アドレスの NAT を行っても IP ヘッダのチェックサムを意識的に計算する必要がない．NAT セッションプログラムは実験段階なので，NAT の処理に最低限必要なコマンドである NAT，FIN，+OK のみを実装している．またデータ送信ホスト，受信ホストは 1 台ずつ，一度に送信できるデータの種類も 1 種類のみ，SrcIP アドレスのみを変換する．以下に送信側環境 (B)，受信側環境 (a)，かつ片方向データ通信におけるパケットの流れと NAT セッションのための応用層の通信を時間軸にそって説明する (図 3 参照)．この例ではパケットの流れは FTP によるファイル転送とする．

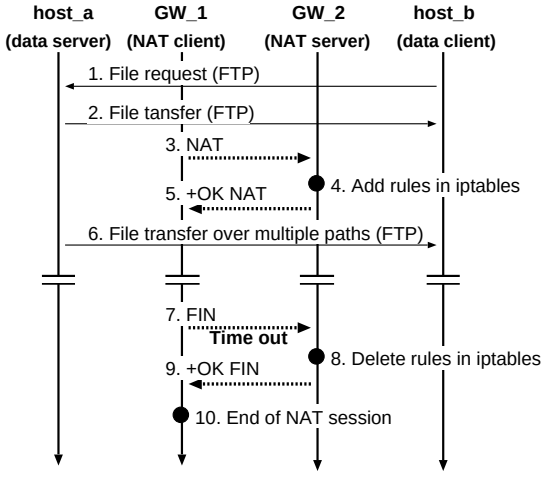

図 3　NAT セッションとデータ振り分けの流れ

説明

1〜2. host_b(データ受信ホスト) が host_a(データ送信ホスト) に対し，FTP を用いてファイル転送を要求し，ファイルの転送が開始される．

3〜5. NATclient はデータパケットを感知し NAT 開始要求メッセージを送信する．NATserver はメッセージを用いて要求された NAT を設定し，NAT 許可メッセージを送信する．

6. NATclient でパケットの SrcIP アドレスを NAT し，複数経路へ振り分け，NATserver でもとの IP アドレス

に戻す．

7〜9. ファイル転送が終了し，NATclient がデータパケットを 3 分間感知しなければ，NATclient は NAT-server に NAT 終了メッセージを送信する．NATserver は NAT 設定を解除し，NATclient に確認メッセージを送信する．

10. NATserver からのメッセージを受信後，複数経路の設定を解除し，NAT セッションを終了する．

このように，NAT を自動的に開始し，複数経路へ振り分け，終了することが可能となった．

5 実験結果

NAT 無し，NAT 有りのネットワーク環境で測定した TCP，UDP のスループット測定結果について延べる．なお 10 回の測定からスループット平均と 95% の信頼区間を求めた．

5.1 TCP の測定結果

スループット測定には FTP を用い，524288000 (500M) バイトのファイルを転送した．複数経路への振り分け方法は，比率に従い連続で振り分ける方法[*2]を用いた．

単一経路，NAT 無し，有りのスループットを測定した．512k，1.5M bits/sec のスループットも測定したが，測定結果には 8M bits/sec のみを記載した．

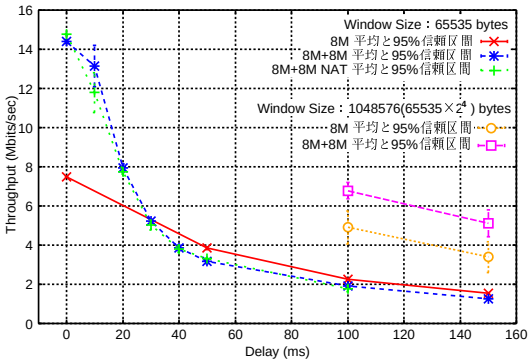

図 4　TCP スループット

図 4 より，NAT 無し，有りの複数経路で遅延が無い場合，使用経路の合計帯域幅に近いスループットが得られ，複数経路で NAT を使用してもスループットに与える影響が少ないことが分かった．遅延が発生すると合計帯域幅を大きく下回った．これはウィンドウサイズの大きさが十分でなく，遅延により ACK が遅れ先送りできるデータが少なくなり，送信ペースが低下するからである．遅延が発生していない場合，理論値の約 92% のスループットが得られたが，設定遅延が 50ms のとき，得られたスループットが理論値[*3]の約 66% であり，単一の約 77% に比べ低くなった．これは，複数経路を利用することによってパケットの入れ替わりが発生し，多くの SACK[*4]が返信されているからであると考えられる．

スループット低下の原因であるウィンドウサイズを通常の 16 倍にして再度測定した結果 (図 4 参照)，変更前に比べスループットが向上した．したがって遅延が発生している場合，ウィンドウサイズを大きくすることによってスループットを向上させることが可能であることが分かった．

そして使用経路の設定に差があるネットワークでスループットを測定した．NAT 無しの環境で，

1. 経路の設定遅延に差がある
   主経路：8M bits/sec，100ms (上り下り)
   副経路：8M bits/sec，50ms (上り下り)
2. 経路の帯域幅に差がある
   主経路：8M bits/sec，遅延無し
   副経路：1.5M bits/sec，遅延無し

場合のスループットを測定した．実験 (1) では，振り分け比率を変更してもスループットにあまり差が見られなかった (図 5 参照)．

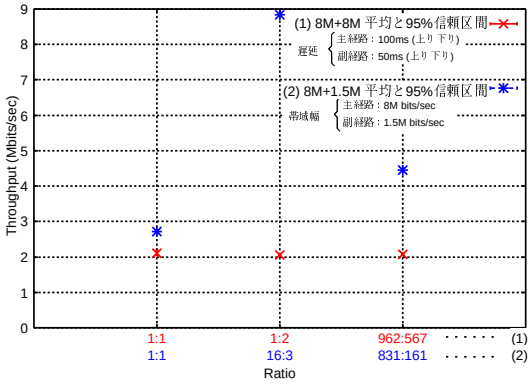

図 5　遅延差，帯域幅差がある場合のスループット測定結果

実験 (2) では，各経路のスループットの比率に合わせてパケットを振り分けた方がスループットが向上することが分かった (図 5 参照)．

5.2 UDP の測定結果

UDP 測定ではスループット測定，パケット入れ替わりを判定するためのプログラムを作成した．なぜならスループット測定アプリケーションである iperf では，遅延設定時のスループットの方が遅延を設定していない時のスループットよりも上がっており，測定結果に信頼性がないと判断したからである．スループット測定では，1 パケットのペイロードが 1000 バイトのデータを指定時間，指定した間隔をあけ送り続けるプログラムを用いた．パケット入れ替わり頻度の解析には，1 パケットの

[*2] 整数比 m：n のとき，m 個，n 個，m 個，n 個と連続して交互に送信する．

[*3] TCP の理論値はウィンドウサイズ，RTT(往復伝搬遅延 + 推定処理時間) を用いて算出できる

[*4] ロスしたパケットのみの再送が可能となり，効率の良いパケット転送が可能となる．

ペイロードが 1000 バイトのデータを 50000 個送信するプログラムを用い，パケットに ID 番号を含めた．

スループット測定結果

測定の結果，単一経路，複数経路ともに遅延に影響されず，設定送信速度に近いスループットを得た．したがって UDP ではスループット向上目的での複数経路利用が有効であると考えられる．

次にパケット振り分け，NAT プログラムの処理能力の限度を知るために，NAT 無し，NAT 有りの実験環境で最大スループットを測定した．NAT 有りは NAT 無しに比べスループットが著しく低下した (表 1 参照)．tcpdump ログを調べたところ，GW_2 でパケット数が著しく減少してたことから，スループット低下の原因は GW_2 にあると考えられる．

表 1　UDP 複数経路最大スループット

比率 (主：副)	NAT	スループット (bit/sec)
1：1	無し	84.74 ± 0.97 M
	有り	34.48 ± 0.19 M

パケットの入れ替わり

複数経路を用いた場合，各経路に帯域幅や発生遅延に差があると，図 6 のようなパケットの入れ替わりが大きくなることが分かった．計測した結果は以下の通りである．

- 各経路の帯域幅を 8M bits/sec とし，主経路に 100ms，副経路に 50ms の遅延を下りに設定
  1：1 の比率でパケットを振り分けたが，主経路，副経路のパケットは交互に受信されず，各経路のパケットが複数まとめて受信されていた．比率を変更しても同様の入れ替わりが見られた．
- 主経路に 8M bits/sec，副経路に 1.5M bits/sec とし，遅延の設定なし
  1：1 の比率ではパケット順序が非常に入れ替わった結果となったが，帯域幅の比率 (16：3) で振り分けた結果，1：1 に比べ入れ替わりの度合が小さくなった．

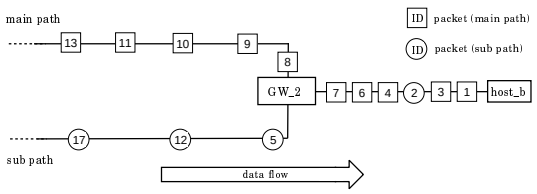

図 6　パケット順序の入れ替わり

6　おわりに

本研究結果より，Divert，Raw Socket を用いることによって複数経路への IP データグラムの伝送が可能であり，既存アプリケーションを変更せずにスループット向上目的，負荷分散目的での複数経路利用が可能であることが分かった．また，ネットワーク環境によっては必要となる NAT を自動的に開始し，複数経路へ振り分け，終了することができた．振り分けのみに比べスループットは低下するが，実用的なスループットが得られた．このことから，RTP などのマルチメディアストリーミングでの利用に利点があると考える．

トランスポート層のプロトコルが TCP の場合，発生遅延が小さいと合計帯域幅に近いスループットが得られ，複数経路利用によるパケットの入れ替わりの影響は少なかった．しかし発生遅延が大きいとスループットは著しく低下するので，スループット向上ではなく，負荷分散目的での複数経路利用が有効であると考えられる．

プロトコルが UDP の場合，パケットの順序が入れ替わったが遅延が発生していても合計帯域幅に近いスループットが得られたので，スループット向上目的での複数経路利用が有効であると考えられる．パケットの入れ替わりはパケット送信間隔を変更するなど，送信側で工夫する必要がある．

今後の課題は GW のセキュリティ，本研究プログラムの最適化，振り分けの条件の 3 つである．セキュリティでは，第 3 者による GW の不正利用を防ぐために GW で利用者認証システムを導入する必要がある．本研究プログラムの高速化では，プログラムがハードウェアに与える負荷を調べ，プログラムを改善する必要がある．そして振り分けの条件では，アプリケーションプロトコルで複数経路利用の必要性を判断する必要がある．

参考文献

[1] 川島 佑毅, 峰野 博史, 石原 進, 水野 忠則：“複数経路通信における動的トラフィック制御の検討, ” 情報処理学会研究報告, 2004-MBL-31　2004-ITS-19, pp.171-176 (2004.11).

[2] 殿内 雅晴, 峰野 博史, 石原 進, 高橋 修, 水野 忠則：“TCP を拡張した複数経路通信における再送制御に関する検討, ” 情報処理学会研究報告, 2004-MBL-28, pp.203-210 (2004.3).

[3] Lee. G. M. , Choi, J. S. ：“A survey of multipath routing for traffic engineering” (オンライン), 入手先＜ http://vega.icu.ac.kr/~gmlee/research/papers/a␣survey␣of␣multipath␣routing.pdf ＞ (参照 2005.5).

[4] 林 孝典, 山崎 真一郎, 森田 直人, 相田 仁, 武市 正人, 土居 範久：“インターネットを用いた複数経路データ伝送方式の性能評価, ” 電子情報通信学会論文誌, Vol.J84-B, No.3, pp.523-533 (2001.3).

[5] NIST Net Home Page, http://snad.ncsl.nist.gov/itg/nistnet/.