

仮想リンクを利用した送受信ノード間の RTT 分割 による転送スループットの向上

小林 弘和[†]

中山 雅哉[†]

Improvement of Transfer Throughput dividing RTT between source node and destination node with virtual links

Hirokazu Kobayashi

Masaya Nakayama

概要 TCP を用いてデータ転送を行う際、TCP の基本則である帯域遅延積によって、遅延の大きな 2 地点間でデータ転送を行う場合にはリンク間帯域を十分使うことはできず、転送スループットを上げることができない。この問題を解決するために、我々は本稿において、データ送受信ノードの間に転送を中継するノードを置き、送信ノード - 中継ノード、中継ノード - 受信ノードと、2 つの TCP のセッションを利用し、送受信ノードを直接結んだ RTT を仮想的に分割し、送受信ノードを直接結んだ RTT よりも短い RTT でデータ転送を行える手法を提案する。そして、提案手法を用いることにより、RTT のみが転送スループットに影響を与える理想環境において、帯域遅延積によって受ける制限の約 2 倍の転送スループットを実現することに成功し、また実環境においては、RTT 分割を実現できる中継ノードを選定できることを示した。

1 はじめに

地球環境の観測データを取り扱う研究分野では、各地で生成された観測データを収集して解析が行われている。また、高エネルギー物理学、気象学などのデータインテンシブコンピューティングを必要とする分野においても、一度の実験やシミュレーションで生成されるデータを各地に分配し解析を行う。これらの分野では一度に転送されるデータ量が年々増加の傾向を示している。例えば、各地の観測データに関しては、ただ値を得るだけでなく、カメラなどを利用して映像としてデータを保持しておくことが始められている。またデータインテンシブコンピューティングの分野においては、年間にペタバイトスケールでのデータを生成し、それを世界中の数十ヶ国、数千人の人達に配る、といったことが必要とされている。このように、大容量のデータ処理を必要とする研究分野では、数 GB ~ TB のオーダのデータを一度に転送することが必要となると考えら

れる。

このようなデータ転送を行う際には、従来では FTP (File Transfer Protocol) を用いてデータ転送を行う方法が主流であった。しかし、FTP のように TCP [9] を用いて作成されているアプリケーションは、TCP の基本法則である帯域遅延積の影響を受ける。帯域遅延積とは、式 (1), (2) で表される様に、リンク間の帯域と、リンク間の RTT によってアプリケーションでの最大スループットが制限されることである。

ここで、

$$\begin{cases} \text{ReceiveWindowSize}(RWS) = w(KB) \\ RTT = r(ms) \\ \text{リンク間の帯域} = z(Mbps) \\ \text{MaximumTransmissionUnit}(MTU) = m(KB) \end{cases}$$

としたとき、アプリケーションが実現可能な最大スループット $Th(Mbps)$ は、パケットロスやビットエラーがない理想環境においては、理論的に

$$\frac{8w}{z} \leq r \text{ のとき、} Th = \frac{8w}{r + \frac{8m}{z}} \quad (1)$$

$$\frac{8w}{z} \geq r \text{ のとき、} Th = z \quad (2)$$

[†] 東京大学大学院新領域創成科学研究科

Graduate School of Frontier Sciences, The University of Tokyo

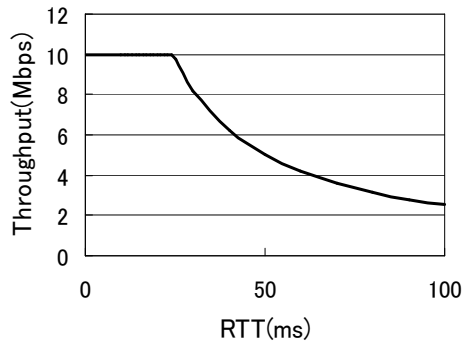


図1 $z = 10$, $w = 32$ のときの r と Th の関係

として計算される。すなわち (2) 式が成立する、 r が小さい場合には、 Th は r の変化に影響されない。一方、(1) 式が成立する場合は、 Th は r の影響を受け、 r が増加する程、 Th は低下することになる。Linux OS では、標準の RWS の値は 32KB であり、ファイル転送を行うノード間の最低リンク間帯域 z が 10Mbps となる環境では、 Th と r の間の関係は図 1 に示される様になる。このように、リンク間の帯域は 10Mbps あるのに対し、理論値であっても、 $r = 60$ で、約 4Mbps、 $r = 100$ になると、約 2.5Mbps 程度しかスループットを出すことができない。そのため、広域環境でデータ転送を行う場合は、帯域遅延積の制約で高速なデータ通信を行うことができない。このため広域環境でのデータ転送のスループットを向上させる方式の検討は大きな課題となっている。

これまで行われてきたいくつかの研究では、RWS やリンク間の帯域を調整することで、データ転送のスループット向上が図られてきたが、前者では最適 RWS を見つけ出すことが困難であり、後者では送受信ノード間を直接結んだ経路以外のリンクの帯域を知る必要があり、それらを容易に知ることはできない。

そこで我々は RTT に着目し、送受信ノード間の途中にデータ中継ノードを置き、中継ノードにおいて、TCP のコネクションを切断することにより、(1) 式が成立する環境の r を短縮する効果を用いてスループットを向上させる方式の提案を行う。この場合、中継ノードによるデータ転送のオーバーヘッドが無視できるほど小さければ、 Th を向上させることが可能である。

以後、第 2 章においては、スループット向上に関する関連研究を上げ、第 3 章では我々の提案する中継ノードを用いた RTT 分割手法を述べ、RTT の

みがデータ転送に影響を与える理想環境においてその有効性を検証する。次の 4 章では、実環境において提案手法が有効である一例を示し、最後に第 5 章でまとめを行う。

2 関連研究

この章では、データ転送における、スループット向上に関する関連技術をあげる。

2.1 w を変更する技術

(1) 式の環境において Th を決めるパラメータの一つに、 w がある。このパラメータを変更し、 Th をあげる転送技術として、GridFTP [10] が挙げられる。GridFTP とは、広域分散環境におけるデータ転送の効率を上げることを目的とし、従来用いられてきた FTP に様々な改良が加えられたファイル転送プロトコルである。そのうちの一つとして、送受信ノード間で RWS の交渉を行い、動的に RWS を変更することにより、転送時のスループットを向上させている。動的に RWS を変更することにより、予め RWS を固定させておく場合に比べ、より大きな RWS でデータ転送を行うことができる。RWS の動的変更を行う手法として提案されているものの一つとして、実際にリンク間にテストデータを送信し、リンク間の帯域を測定し、RWS を決める手法 [2] がある。しかし、この手法を用いると、最終的に求められる RWS がテストデータを送信したときの RWS に依存してしまう問題点が考えられる。

現在のところ、RWS の交渉を動的に行う具体的実装や、動的交渉を行う手段も確定的なものは存在せず、現在も研究が進められている。 [11]

2.2 z を向上させる技術

次に、 z を向上させる技術を紹介する。一般に、送受信ノード間で直接データ転送する場合、インフラネットワークは固定的であるため、 z は一通りしか存在しない。しかし、他のノードを迂回してルーティングを行うことができれば、送受信ノード間を直接結んだ z とは異なる z を用いてデータ転送を行い、 Th を上げることができるケースがある。RON [3, 4], File Mover [5] などはこの方式に基づく研究例である。これらの方式では、その場に存在する利用可能なノード間においてフルメッシュのオーバーレイネットワークを構築し、作成した仮想リンク間において、データ転送の際に最も大きい Th を出すことができる経路を選択する方法が用いられている。実際に転送経路を見つける際、両者の

手法とも、予め各リンク間の利用可能帯域を定期的に測定しておく。そして、RON の場合は、それらの情報を全てのノードで交換し合い、データ転送が起こるときにはその情報に基づいて経路の選択が行われる。また、File Mover の場合は、リンク間の情報は Information Server と呼ばれるノードに集められ、そのノードが送信ノードから受信ノードまでの全経路のうち、Dijkstra のアルゴリズムを利用して最大 Th が実現できる経路を発見する。

これらの手法はスループットを向上させるために、実際にリンク間の利用帯域を測定することが要求される。そのため、実際のデータ転送とは関係のない、リンク間の状態を計測するためのデータを定期的に送信し、それらのデータから得られる情報を利用して経路の選択を行う。そのため、より精度を高く、リンク間の状態を得ようとする、データ転送のための帯域を圧迫することになる。

フルメッシュからリンクを削除することにより、計測トラフィックを削減する方法として、PLUTO [8, 7] が提案されている。この手法は、BGP のルーティングテーブルや voronoi 図を利用することにより、インフラネットワークでの経路が重複するオーバーレイネットワーク上のリンクを削除することで検査するリンク間状態の数を削減している。これにより、フルメッシュからリンクを削減することはできるが、高い精度でリンク間の状態を得ようとするためには、より短い間隔で計測トラフィックを流さなければいけない、という点は解決されていない。

またこれらの方法は、データ送受信ノード間で利用帯域が最大である経路を用いて転送することに主眼が置かれているが、選ばれた経路におけるデータ送受信ノード間における r の値については注目していない。すなわち、(1) 式において、 z が大きいところであることが第一として考えられており、その経路を選んだときの r の影響は考えられていない。このため、 $r + \frac{8m}{z}$ のうち、 r のほうが $\frac{8m}{z}$ よりも十分大きくなると、 Th の向上は見込めないことになる問題がある。

2.3 複数のコネクションを用いる技術

その他にも Th を向上させる手法として、データを様々な場所に分散配置させておき、複数の送信ノードから受信ノード間の間に TCP コネクションを並列に確立し、データ転送を行う方法がある [10]。この方法は、利用帯域に関して経路上に大きなボ

トルネックがない場合は、 r が大きな環境では有効である。しかし、経路上に大きなボトルネックがある場合は、そのリンクの転送性能を超える並列度でデータ転送が行われるとパケットロス等により、却って転送するーブットが低下することになる。

2.4 Performance Enhancing Proxies(PEP)

PEP [6] とは衛星を利用したデータ転送時の RTT が非常に大きい環境やモバイル環境と有線環境が複合した環境において、途中にプロキシを置き、ACK を早期に返すことによって、転送効率を上げる手法である。この手法は RTT に着目すると言う点で Th を向上させることができるが、衛星や無線 LAN 環境に適応する研究例がほとんどで、一般の通信環境への適用を検討している事例は見られない。

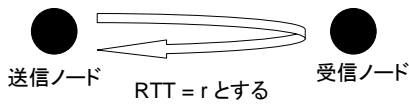
3 提案手法とその評価

3.1 提案手法の概要

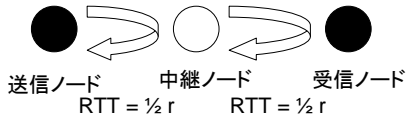
前章であげた関連研究では、(1) 式における w, z に注目して Th を向上させる方法について述べた。しかし、それぞれの方法には前章で指摘した問題点があり、前者は決定的手法がないこと、後者は特に大規模なネットワークにおいて問題が顕著となる。そこで、我々は (1) 式におけるもう一つのパラメータである r に着目して、 Th の向上を行う。 r の値は、(1) 式において $r \gg \frac{8m}{z}$ の環境であれば、 Th に影響を与える要因が最も大きなパラメータである。そこで我々は、送受信ノード間の RTT に注目し、より短い RTT でデータ転送を行うことによって、スループット向上を行う手法を提案する。具体的には、データ転送を行う場において利用可能なノード間でオーバーレイネットワークを構築することにより、仮想的なリンクを作成し、送受信を行うノード以外の、ある一台のノードを中継ノードと見立て、その中継ノードを経由することによって、より短い RTT でデータ転送を行い、データ転送完了時間を短縮し、スループットを向上させる、という方法である。中継ノードを利用することにより、データ転送を行いたい 2 地点間を単純に結んだ RTT よりも短い RTT でデータ転送をすること、すなわち RTT を仮想的に分割してデータ転送を行うことが可能になる。提案手法のイメージ図を、図 2 に示す。

以後、本章では RTT 分割の有効性を理想環境を用いて行った実験と共に述べる。なお、作成した理

従来の転送方法



提案手法



中継ノードにおけるオーバーヘッドが無視できれば
送受信ノード間を $1/2r$ で結んでいるのと等価
Thは上図の約2倍期待できる

図2 提案手法の概略図

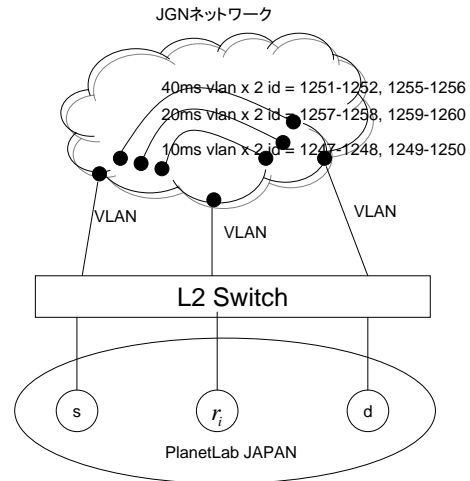


図3 構築した理想環境

想環境については、後に詳しく述べる。

3.2 中継ノードの利用の有効性

3.2.1 理論的側面による有効性の検証

本稿では、送受信ノード以外に、その場に存在する利用可能なノードの、ある1ノードを中継ノードとして利用することを提案する。この手法を用いることにより、中継ノードを利用することによって、送受信ノード間を直接結んだRTTよりも短いRTTでデータ送受信を行える。

なぜなら、ここで仮に、送信ノードを s 、受信ノードを d 、中継ノード候補を r_i とし、送受信ノード間のRTTを RTT_{sd} 、送信ノードと中継ノード候補間のRTTを RTT_{sr_i} 、中継ノード候補と受信ノード間のRTTを $RTT_{r_i d}$ とし、リンク間の利用可能帯域をそれぞれ、 BW_{sd} 、 BW_{sr_i} 、 $BW_{r_i d}$ とすると、(1)式が s, r_i 間、 r_i, d 間、 s, d 間に成立している環境であり、 $RTT_{sr_i}, RTT_{r_i d}$ のうちの大きいほうが、 RTT_{sd} よりも小さい、すなわち $\max(RTT_{sr_i}, RTT_{r_i d}) < RTT_{sd}$ となる r_i を選ぶことができれば、中継ノードにおけるデータ中継のオーバーヘッドが無視できる環境では、RTTが $\max(RTT_{sr_i}, RTT_{r_i d})$ の2地点間でデータ転送を行っているに等しく、 s, d を直接結んだ転送よりもスループットを出すことができる。

3.2.2 実験的側面からの有効性の検証

次に、それが実際に行えるかを行った実験と共に示す。この実験では、RTTのみがデータ転送のスループットに影響を与える理想環境を作り出し、その理想環境の中で提案するRTT分割手法でスループットが向上するのを示す。

理想環境の実現には、JGN2ネットワークとPlanetLab JAPANの環境を用いて図3の環境を

構築して行った。理想環境を実現させるため、ノードの負荷対策として当実験以外のユーザプロセスを起動させないことや、ノード間においてネットワークトラフィックを発生させない環境を構築した。そのため、データ転送にはRTTのみが影響する理想環境と考えることができる。

より具体的には、東京、仙台、大阪、岡山、福岡、金沢を繋いでいるJGN2ネットワークを利用し、東京にあるノード内でのループによりRTTが10msのvlanを二本(vlan id = 1247-1248, 1249-1250)、東京~仙台~東京のループを作ることによりRTTが20msのvlanを二本(vlan id = 1257-1258, 1259-1260)、東京~大阪~岡山~福岡~金沢~東京のループを作ることによりRTTが40msのvlan二本(vlan id = 1251-1252, 1255-1256)用意した。これらのvlanをPlanetLab Japanのノードの直近のL2 Switchで組み合わせることにより、RTTが10ms~100msまで、10ms刻みで変化させ、データを取得した。

図3に示す様に s, r_i, d とともにPlanetLab JAPANのノードを利用し、各ノード間を、JGN2ネットワーク上で10ms, 20ms, 40msの遅延時間に調整したVLANをL2 Switchで組み合わせることによって適切な遅延時間を作り出し、2つの実験を行った。1つは、 $RTT_{sr_i} = RTT_{r_i d} = 1/2 RTT_{sd}$ となる r_i を用いた場合のスループットの結果を、もう一つは、 $RTT_{sr_i} = 10ms$ 、 $RTT_{r_i d}$ を10ms~70msに変化させた時のスループットを測定した。 $RTT_{r_i d}$ は10msと固定して実験を行った結果を図4に、また $RTT_{sr_i} = RTT_{r_i d}$ として実験を行った結果を、図5に示す。図4, 5とも、横軸

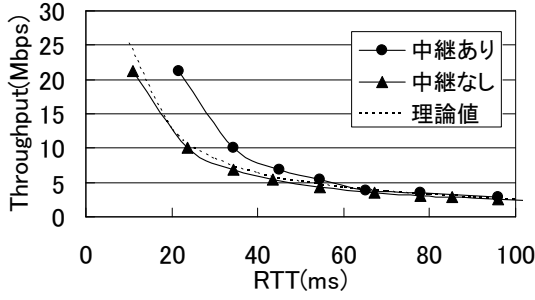


図4 $RTT_{r_i,d} = 10\text{ms}$ と固定した場合の RTT とスループットの関係

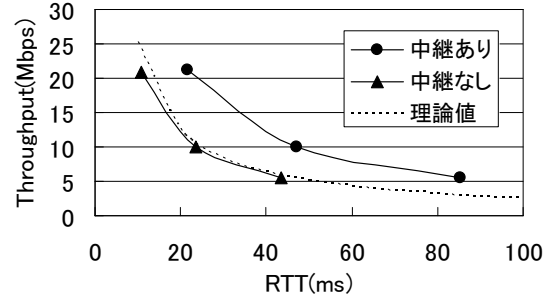


図5 $RTT_{sr_i} = RTT_{r_i,d}$ 時の RTT とスループットの関係

である RTT は、送受信ノード間の RTT であり、 $RTT_{sr} + RTT_{rd} = RTT_{sd}$ が成立している。また、両者とも、送信したデータサイズは 10MB であり、理論値のグラフに用いた値は、 $z = 100, w = 16$ である。

図 4, 5 から見てわかるように、中継ノードを利用して RTT 分割を行ったときのほうがデータ転送のスループットは向上していることがわかる。図 4 の場合、横軸の RTT が大きいほど、さほど効率化は見られない。これは、 $RTT_{r_i,d}$ を 10ms に固定しているため、 RTT_{sd} が大きくなると、 $\max(RTT_{sr_i}, RTT_{r_i,d})$ は RTT_{sd} に近づく。その影響を受けることにより、効率化はさほど行われていない。

その一方で、図 5 の場合を見てみると、こちらの場合は、 $RTT_{sr_i} = RTT_{r_i,d}$ としているため、 RTT_{sd} が大きい環境でも、スループットは理論値の約 2 倍の結果を示すことができている。

4 中継ノード選定手法とそのパラメータの検討

前章では、中継ノードを利用することの有効性について述べた。ここでは、利用する中継ノードを実際にどのようにして選ぶかについて説明する。

4.1 中継ノード選出手法

選ぶ中継ノード、 r_i の条件は、 $RTT_{sd} > \max(RTT_{sr_i}, RTT_{r_i,d})$ を満たすことであり、この条件下において最もスループットを高めるのは、 $\max(RTT_{sr_i}, RTT_{r_i,d})$ が小さくなる時なので、ほぼ $RTT_{sr_i} = RTT_{r_i,d} = 1/2RTT_{sd}$ が成立する r_i を探すことになる。ここでは、この条件を満たす r_i を選定する方法について説明する。

まず、前提条件として、「全てのノードは、任意のノード間にあるルータとルータまでの RTT を調

べることができる」ということを仮定し、以下の方法により、 r_i を選定する。この仮定を踏まえ、選定方法を説明する。

1. データ転送要求が生じた時、受信ノードは送信ノードにデータを要求する
2. 送信ノードは、受信ノードに関する情報を他の全てのノードにブロードキャストする。それを受けたノードらは受信ノードまでのインフラネットワーク上における経路、経路上の各ルータまでの RTT を調べ、送信ノードに報告する
3. 全てのノードから受信ノードへの経路を受け取った送信ノードは、後述する手法で各ノードの経路情報を解析することで中継ノードを選択する

上記 3 番目の解析手法について詳しく説明する。まず送信ノードは、自分から受信ノードまでの経路を基に、経由したルータを調べる。そして、経由したルータの個数を RN_{sd} とおく。次に、他の全てのノードからの経路情報を基に、経由したルータを調べる。同じように、経由したルータの個数を $RN_{r_i,d}$ とおく。ここで、図 6 のように、各ノードについて、データ送信ノードからの経路と共通するルータの個数を RN_{same_i} を調べる。ここで、下式、

$$\begin{cases} p \leq \frac{RN_{same_i}}{RN_{sd}} \leq q \\ RN_{r_i,d} \leq RN_{sd} \end{cases}$$

が成立しているノードを中継ノード候補とする。ただし、

$$\begin{cases} 0 \leq p \leq 1 \\ 0 \leq q \leq 1 \\ p \leq q \end{cases}$$

であり、この p, q についての最適値については、現在検討中である。

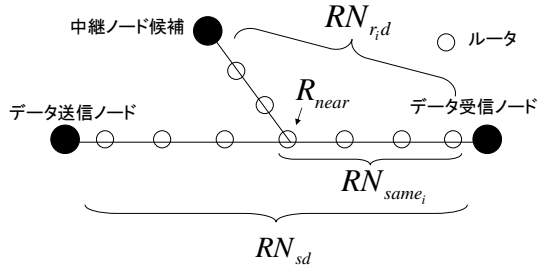


図 6 中継ノードの選択

次に、得られた経路上の RTT を解析する。送信ノードと受信ノードの経路上にあるルータと、中継ノード候補と受信ノード間の経路上にあるルータのうち、共通するルータで最も送信ノードに近いルータを R_{near} とおく。ここで、送信ノードと R_{near} までの RTT を $RTT_{sR_{near}}$ とおく。このとき、

$$x \leq \frac{RTT_{sR_{near}}}{RTT_{sd}} \leq y \quad (3)$$

を満たしているノードを最終的な中継ノードとする。ただし、

$$\begin{cases} 0 \leq x \leq 1 \\ 0 \leq y \leq 1 \\ x \leq y \end{cases}$$

であり、この x, y の最適値も p, q と同じく現在検討中の段階である。

ここで、RTT のみにおいて、中継ノードを選定せず、ルータの個数を用いることについて説明する。ルータの個数を用いずに RTT のみで中継ノードを選定した場合、

$$\frac{RTT_{r_i d}}{RTT_{sd}} \quad (4)$$

の値のみで r_i を選定することになり、このとき、

$$RTT_{sr_i} > RTT_{sd} \quad (5)$$

となってしまう可能性が考えられる。ルータの個数を考慮することにより、(5) 式の可能性を排除することができる。

次に、以上までの中継ノードの条件を満たしているノードが複数存在した場合について説明する。この場合、中継ノード候補として残っているノードは、

$$\begin{cases} RTT_{sr_i} < RTT_{sd} \\ RTT_{r_i d} < RTT_{sd} \end{cases}$$

を満たしている。そこで、それら全ての候補ノードに対して、 $RTT_{sr_i}, RTT_{r_i d}$ を測定し、この二つの

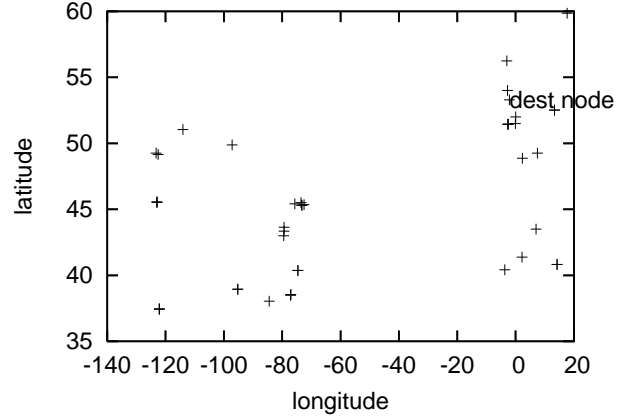


図 7 利用したノードの地理的分布

値の最大値が最小であるノードを中継ノードとして選択する。

中継ノードの条件を満たすノードが存在しなかった場合は、直接送受信ノード間でデータ送信を行う。

4.1.1 パラメータの検討

ここでは、前項で登場した 4 つのパラメータ、 p, q, x, y について、実環境において実際に実験を行った結果に基づき検討を行う。

実験環境としては PlanetLab [1] を利用した。PlanetLab とは、オーバーレイネットワークに関するテストベッド環境である。世界中の研究、教育機関が、それぞれが持つ計算資源を提供することによって成り立つ環境であり、現在全世界中に約 600 台のノードが分散している。ユーザはそのノードのうち、使いたいものを選択し、利用することができる。

ここでは、PlanetLab 環境の中から地理的に分散する様に 50 ノードを選定し、ランダムに s および d を選んだ時に、 p, q, x, y をどのような値にすれば提案する選定方法で r_i を選定できるかの結果についてまとめている。図 7 に利用したノードの地理的分布図を示す。

まずはじめに、パラメータ p, q についての検討を行う。利用した 50 台のノードのうち、一台のノード (planet1.manchester.ac.uk) をデータ受信ノードに固定し、その他の全てのノードを送信ノードとしたときの $\frac{RN_{same_i}}{RN_{sd}}$ の値がどのように分布するかを調べた。結果を図 8 に示す。

この図から、中継ノードとしては利用できないノード、すなわち比が 0 であるノードが多数検出されていること、また、ネットワーク的にちょうど

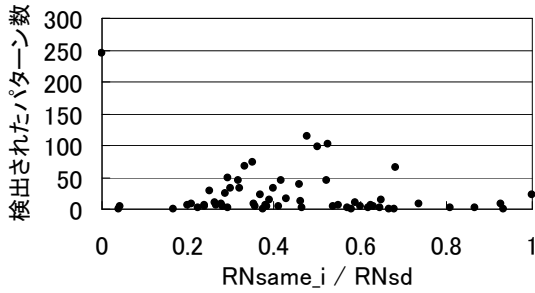


図 8 重複するルータ数の分布

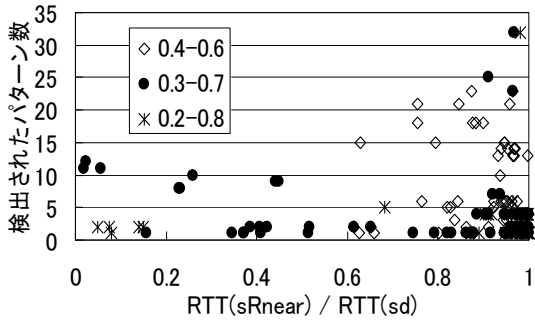


図 9 p, q を変化させたときの (3) 式の分布

中間点に位置するノード、すなわち比が 0.5 付近のノードの存在が確認できる。

次に、 p, q の値の組が、 $(0.4, 0.6)$, $(0.3, 0.7)$, $(0.2, 0.8)$ となるように範囲を広げていったとき、 $\frac{RTT_{sRnear}}{RTT_{sd}}$ の値がどのような分布になるかを調べた。結果を図 9 に示す。ただし、共通する部分に関しては、例えば $p = 0.3, q = 0.7$ のとき、 $0.4 \sim 0.6$ の値は $p = 0.4, q = 0.6$ の取得分としてプロットしているため、新たに増えた点のみをプロットしている。

この図からわかるように、ルータの個数比が 0.5 付近であっても、最終的な RTT 比が必ずしも最適になるとは限らないことがわかる。この場合はルータの個数比を $0.3 \sim 0.7$ とした場合に、RTT 比が 0.5 に最も近い中継ノードを選定できている。

これらの結果から、一番多く分布しているのは 0.9 以上であるが、これは図 10 のようなパターンであり、こういったノードを中継ノードとして選択すると、スループットの向上はさほど期待できない。

今回の 1 パターンにおけるノードの分布においては、4 つのパラメータ p, q, x, y を、 $p = x = 0.3, q = y = 0.7$ と定めると、中継ノードを利用してデータを送信したほうがよい s, d の組が存在することが確認できた。(ただし、ここでは d は planet1.manchester.ac.uk に固定したままであ

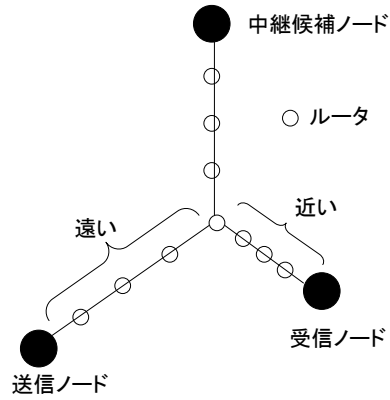


図 10 候補にはなるが中継ノードに適さない場合

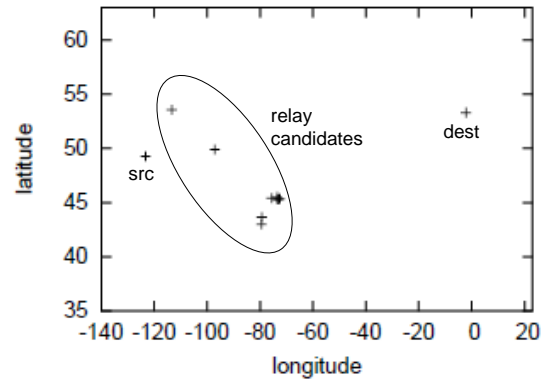


図 11 中継ノードとして選択できる例

る。)今回行った実験のパターンでは、 s になりうる 49 台中、設定したパラメータを満たすノードが見つけたのは、11 ノードであった。すなわち、アメリカ大陸の内陸部にあるノードが、海岸に近いノードを中継ノードとして選択するパターンである。そして、それら 11 ノードの s のうち、中継ノードとなりうる条件を満たしているノードを最も多く見つけられた $s(\text{planetlab1.cs.ubc.ca})$ は 9 台のノードを中継ノードとして見つけることができた。具体的な s, r_i を、図 11 に示す。

以上のように、RTT のみが転送に影響を与える環境において、RTT 分割によりデータ転送のスループットを向上させられることを示し、また、送信ノード、中継候補ノード、受信ノードが適切な位置関係に存在すれば、RTT 分割を行うことができる中継ノードは存在することを示した。

5 まとめ

本稿では、データ転送時のスループットを向上させるため、RTT に着目し、送受信ノード間における RTT を、他のノードを中継ノードとして利用す

ることにより分割し、直接転送する RTT よりも短い RTT でデータ転送を行う手法を提案した。

また、RTT のみがデータ転送に影響を与える理想環境において、実際に RTT 分割手法によってデータ転送のスループットをあげれることを示した。RTT をちょうど 2 等分するようなノードを中継ノードに選択すれば、スループットは理論値の約 2 倍出せることも示した。

実環境において行った実験では、50 台のノードを利用して中継ノードとして利用できるノードが存在するかどうかの実験を行い、中継ノードとして選定できるパラメータの検討を行い、中継ノードの存在可能性を示した。

今後の課題としては、

- 選定パラメータの決定
今回の実験はある特定の 1 パターンのみの環境において行ったものであるため、具体的なパラメータ値を設定するには不適當である。これにはより多くのパターンにおいて実験を行う必要がある。
- 本手法を用いることによるオーバーヘッドの考慮
本稿では、中継ノード発見にかかるコスト、中継ノードにおけるデータ中継のコストを問題にしていない。これらに関しても検討が必要である。また、オーバーヘッドを考慮した場合、提案手法以外の手法、例えば本稿でも述べている、ルータの個数を使わず RTT のみで中継ノードを選定する手法などが、より低コストで中継ノードを選定できる可能性がある。これらに関する検討が必要である。
- 実環境においての、本手法を用いたスループットがどの程度向上するかを検証
- RTT のみならず、利用可能帯域も考慮したデータ転送中における動的な経路変更

を考えている。

参考文献

[1] PlanetLab. <http://www.planet-lab.org/>.
[2] W. Allcock, J. Bester, J. Bresnahan, A. Chervenak, L. Liming, and S. Tuecke. Gridftp: Protocol extensions to FTP for the Grid. *Internet Draft*, May 2001.

[3] David Andersen, Hari Balakrishnan, Frans Kaashoek, and Robert Morris. Resilient Overlay Networks. *In Proceedings of the 18th ACM Symposium on Operating Systems Principles(SOSP)*, pp. 131–145, Oct 2001.
[4] D.G. Andersen, H. Balakrishnan, M.F. Kaashoek, and R. Morris. The Cace for Resilient Overlay Networks. *In Proceedings of the 8th Annual Workshop on Hot Topics in Operating Systems*, pp. 152–157, May 2001.
[5] Cosimo Anglano and Massimo Canonico. The file mover: An efficient data transfer system for grid applications. *Cluster Computing and the Grid(CCGrid)*, pp. 625–632, 2004.
[6] J.Border, M.Kojo, J.Griner, G.Montenegro, and Z.Shelby. Performance enhancig proxies intended to mitigate link-related degradations. *Request for Comments(RFC) 3135*, June. 2001.
[7] Aki Nakao and Larry Peterson. Scalable Routing Overlay Networks. *ACM SIGOPS Operating Systems Review*, Vol. 40, pp. 49–61, Jan 2006.
[8] Aki Nakao, Larry Peterson, and Andy Bavier. A routing underlay for overlay networks. *Proceedings of SIGCOMM*, pp. 11–18, 2003.
[9] Information Science Institute University of Southern California. Transmission control protocol. *Request for Comments(RFC) 793*, Sept. 1981.
[10] The Globus Project. Gridftp: universal data transfer for the grid. *White Paper*, Sept. 2003.
[11] 伊藤建志, 大崎博之, 今瀬真. 広域グリッドコンピューティングにおけるデータ転送プロトコル gridftp のパラメータ設定方法に関する検討. 電子情報通信学会技術研究報告 (IN2004-39), pp. 19–24, July 2004.