

1. はじめに

1.1 研究背景

近年では動画配信など、1つの配信者から複数のホストに向けて短時間内に内容が重複するデータが配信されることが増えている。ネットワークに同じデータが何度も流れる状況には冗長性があり、帯域を浪費しているといえる。我々の研究グループではネットワークノードにパケットキャッシュを設け、情報を損なわずに冗長トラフィックの通信量を削減することが可能なTR (Traffic Reduction) ノードを開発している。TR ノードの処理としては①圧縮②中継③復元の3つがある(図1)。①を担当するノードでは、キャッシュされたデータと同じ内容を持つ受信パケットを圧縮しデータサイズを削減する処理を行う。図1の点線は圧縮されたデータの流れを表す。②を担当するノードでは後続に復元ノードがあるためデータをキャッシュしながら圧縮データをフォワードする。そして③のノードでは、圧縮側と同一のデータを保有するキャッシュを使って圧縮データを元に戻す。これにより、圧縮を担当するTR ノードと復元を担当するTR ノード間で繰り返し送られるデータのトラフィックを削減することが可能となる。

冗長トラフィックを削減する手法として IP

マルチキャストを利用する方法もあるが、TCP 通信に対応していないことや、経路上のすべてのノードがこの IP マルチキャストに対応していなければならないなどの制約がある。その点、TR ノードは TCP 通信に対応しており、削減区間のみが TR ノードで構成されていれば冗長トラフィックが削減でき、TR ノードは IP マルチキャストに比べて汎用性がある。以上の理由から我々研究グループでは IP マルチキャストを利用せずに冗長トラフィックの削減を目的とした TR ノードの開発を行っている。

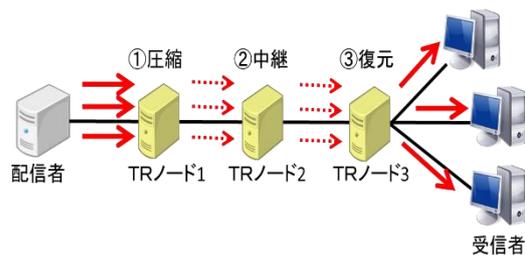


図1 TR ノードの削減処理を示す図

TR ノードではネットワーク層に IP を利用した通信に対応している。TR ノードは配信で流れるパケットのヘッダ領域を使いエンドポイント間に何台の TR ノードが存在するかを常に検査し、途中で経路上の TR ノードが増減した場合にも対応し、削減処理の内容を変更するこ

とができる。ただし、[1]の TR ノードが動作するネットワークに関しては条件がある。[1]の TR ノードでは TCP 通信で流れるパケットが往路と復路で同じ台数の TR ノードを経由して配信が行われるときのみしか最適な動作が保証されないのである。この条件を満たさないネットワークで[1]の TR ノードを動作させるとパケットのヘッダ領域から適切な削減処理を選択することができず、それが原因でエンドポイント間の通信の失敗や削減区間を最大限取れないといった問題が発生する。

本研究の目的は配信で流れるパケットの情報からのみでは最適な削減処理を選択できなかったネットワークにおいても通信の安定を保証しながら TR ノードによって最大限の削減区間を確保させることである。この目的を達成するためには配信データが通過する TR ノードの台数を TR ノード間で伝え合い、適切な削減処理を選択させる必要がある。

本論文は以下の構成である。第 2 節では TR ノードの動作と抱えている問題について述べる。第 3 節では問題の解決のために TR ノードが最適な削減処理を自律的に選択できるような機能について説明する。第 4 節では実際に動作させたときの結果を述べ、有効に動作していることを示す。第 5 節では本論文のまとめと今後の課題について触れる。

2. TR ノード

2.1 TR ノードの制御情報

[1]では TR ノードの削減処理に必要な情報をホストが送信するパケットの IP ヘッダのオプションとして領域に保持していることを前提にしている。このオプションを TR オプションと呼び、図 2 のようなパケット形式となる。上にある数字は各フィールドで要するバイト数である。各フィールドについての説明は本研究で必要なもののみ 2.2~2.3 内で行う。

| | | | | | |
|-----|----|--------|-----|----|------|
| 1 | 1 | 2 | 1 | 1 | 2 |
| タイプ | 長さ | TR_HOP | フラグ | 指示 | CE番号 |

図 2 TR オプションのパケット形式

2.2 冗長トラフィックの削減手法

[1]の TR ノードによるトラフィック削減の基本的な流れを図 3~図 4 を使って説明する。今回の説明ではエンドポイント間に 2 台の TR ノードが存在する場合を想定する。TR ノードはそれぞれ通信キャッシュを持っている。TR ノードでは受信したパケットの内容と通信キャッシュ内の中身を比較し、キャッシュ内に同じデータが無い場合はキャッシュに記録を行う (図 3)。

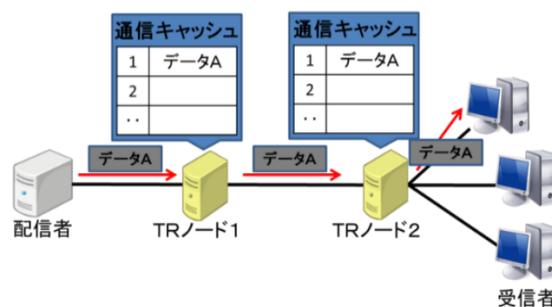


図 3 キャッシュへの記録

図 3 では配信者によって”データ A” が送信され TR ノード 1 がそのデータを受け取った時、キャッシュ内に同じものが無いためキャッシュの更新が行われる。このとき TR オプション内の「指示」部分にはキャッシュの記録を指示する値が格納される。そして TR ノード 1,2 で通信キャッシュの内容を全く同じものにするため TR ノード 1 は TR オプションの「CE(キャッシュエントリ番号)」を使い、データ A を更新するキャッシュのレコードを指定する。そして TR ノード 1 からのパケットを受信した TR ノード 2 は TR オプションの「指示」と「CE番号」をもとに TR ノード 1 と同じキャッシュの内容になるように更新を行う。

次に同じ”データ A”を TR ノード 1 が受け取

った場合には”データ 1”をキャッシュエントリ番号に置き換えた小さなパケットに変換する。その際、TR オプションの「指示」の値は変換したデータの復元を後続ノードに指示する値が格納される。そして変換したパケットを後続へフォワードし冗長トラフィックの削減を行い、図 4 のような形になる。以上の流れで TR ノード 1 と TR ノード 2 の区間で冗長のトラフィックを削減することができる。

なお図 1 のように配信のエンドポイント間に 3 台以上の TR ノードが存在する場合、削減区間を最大限にとるため図 1 の TR ノード 2 は通信キャッシュの更新を行いながら圧縮データを後続へフォワードすることになる。

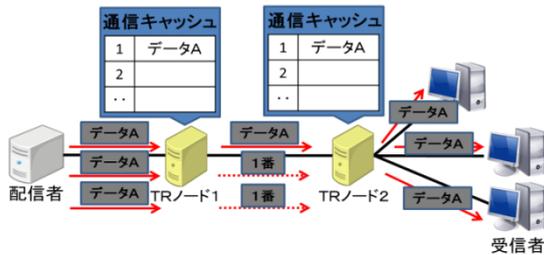


図 4 TR ノードを用いた冗長トラフィック削減

2.3 削減処理の選択方法

TR ノードが圧縮や中継の動作を行うには自身の後続に他の TR ノードが必ず存在しなければならない。TR ノードは TR オプション内の”TR_HOP”というフィールドを用い、後続の TR ノードの有無を調べ、削減処理を選択する。”TR_HOP”は「総数」と「通過数」という値で構成され、総数はエンドポイント間で合計何台の TR ノードを通過するか、そして通過数はある TR ノードがパケットを受信した時まで何台の TR ノードを通過してきたかを示す値である。

図 5 を例にして総数と通過数を使った削減処理の選択について説明する。長方形内の数字は TR オプションの”TR_HOP”の値を示し、上位 2 桁は総数を表す値で図 5 のネットワークであ

らば 3 の値が入る。下位 2 桁の値は通過数の値であり、TR ノードはパケットを受信する度にこの値を 1 加算する。

削減処理は次のように選択する。TR ノード 1 であれば総数が 3 に対し通過数が 1 なので後続に TR ノードが存在することがわかり圧縮処理を行うことになる。TR ノード 2 は総数が 3 に対し通過数が 2 なので、後続に TR ノードがあるとして圧縮データをフォワードする。そして TR ノード 3 では総数と通過数の値が一致したため後続に TR ノードが存在しないとしてキャッシュの内容からデータを復元する処理を選ぶ。

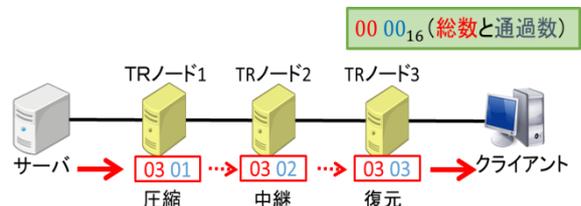


図 5 削減処理の選択方法

2.4 削減処理の選択方法に関する問題

図 6 は本研究で問題としたネットワーク環境の 1 つである。ルータ A は配信データに対しルータ B へルーティングを行い、ルータ C はクライアントからのパケットに対し TR ノード 3 へ固定されたルーティングする場合を考える。この場合、サーバからクライアントへのパケットは合計 2 台の TR ノードを通り、クライアントからサーバへのパケットは合計 3 台の TR ノードを通過する。総数の値はエンドポイントが設定しており、通過数の値を反対方向のパケットの総数のフィールドにコピーすることにより実現される。図 6 でクライアントからのパケットを受信したサーバは通過数の値 3 をクライアントへのパケットの総数の値へコピーし、それを配信データに載せて送る。TR ノード 1 はこれを受信し圧縮データを後続へ送る。しかし TR ノード 2 は総数と通過数の値が一致しないことから後続に TR ノードが存在するとして

TR ノード1が圧縮したデータをフォワードしてしまうという問題が起こる。

図7は図6のTR ノード3とルータBが入れ替わったようなネットワークであるがこの場合、TR ノード2が総数と通過数の値が一致したことから後続にTR ノードが存在するにも関わらず復元処理を行ってしまう。これも削減区間が最大限とれないという1つの問題である。

このように特定のネットワーク環境ではTR オプションの値のみから削減処理を選択すると通信の失敗や削減区間が最大限とれないという問題が[1]のTR ノードでは起こることがわかっている。

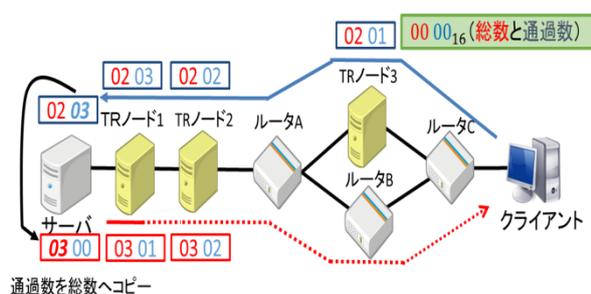


図6 通信の失敗が生じるネットワークの例

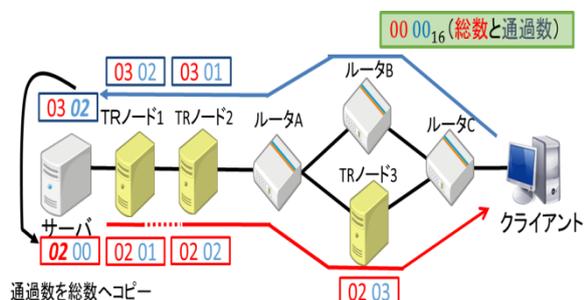


図7 削減区間が最大限とれないネットワークの例

3. 削減処理の自律的選択機能

3.1 最適な削減処理の選択に必要な情報

本研究では2.4で述べた問題を解決するためにTR ノードの削減処理を選択する仕組みを変更する。クライアントが発したパケットの総数の値にはサーバが送る配信データが通過するTR ノードの合計台数が書かれていることからTR ノードはその値を送信元がサーバ、送信先

がクライアントである通信の情報として記録しておく。なおTR ノードは配信の送受信を行うマシンの組み合わせ単位で配信データが通過するTR ノードの台数を記録し、以降説明する削減処理に関する情報を管理していく。これはTR ノード自身がルーティングを行っている配信データを送受信しあうマシンの組み合わせに応じてTR ノードが最適な処理を変更することができるようにするためである。図8はサーバの配信がクライアント1、2へ行われる例だがこの場合、TR ノード2はサーバからクライアント1への配信に関しては復元を行い、サーバからクライアント2への配信に関しては中継の動作を選択するべきである。そのために削減処理の選択に関する情報を配信データの送受信を行うマシンの組み合わせ単位で管理させることによりTR ノード2はネットワークポロジに応じて選択する削減処理を柔軟に変更することができるようになる。

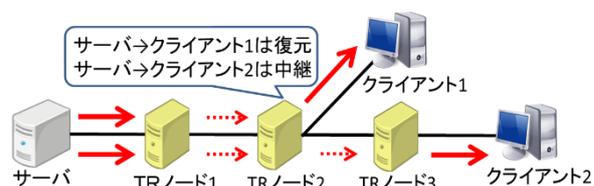


図8 削減処理をエンドポイントの組み合わせ単位で選択する例

3.2 ICMP パケットの仕様とその利用方法

図7のようなネットワークではTR ノード3がクライアントからのパケットを受信できず、3.1で述べた手法で配信データが何台のTR ノードを通過するかを知ることはできない。そこで配信データを最初に受信するTR ノードが記録した総数の値を後続のTR ノードに伝え、それをもとに削減処理を選択させる仕組みを導入する。

なお本研究ではサーバからの配信データを最初に受信するTR ノードはクライアントがサーバへ宛てたパケットも受信しており、なおか

つ図 6、図 7 におけるルータ A、C が一貫して固定されたルーティングを行っているということを想定している。

図 7 における TR ノード 1 はクライアントからのパケットを受信し、配信データが通過する TR ノードの合計台数を知ることができたら ICMP パケットを後続の TR ノードへ向けて送信する。なお、現状では TR ノード 1 は TR ノード 3 の IP アドレスを知る術がない。想定していた条件ではルータ A はクライアント宛てのパケットを一貫して TR ノード 3 へルーティングする。TR ノード 1 はクライアントを送信する ICMP パケットの宛先とすることにより、この ICMP パケットは TR ノード 3 へ到達し、配信データが通過する TR ノードの合計台数を TR ノード 3 へ知らせることができる。

図 9 は TR ノード 1 が送信する ICMP パケットにおける ICMP データ部の形式である。上にある数字は各フィールドで要するバイト数である。この ICMP パケットは配信が開始された直後に送られ、パケットのロストや経路上の TR ノードの台数に増減があった場合以外には送られない。そして ICMP データ部はホップカウンターの後ろにパディングが入るため合計 8 バイトのデータ量となり帯域を流れる。これは一般的な配信で流れる TCP データ部のデータ量と比べてもかなり小さく ICMP パケット送る機能が実装されたことによりトラフィックに与える影響は極めて小さいといえる。

| | | |
|--------------|--------|---------|
| 4 | 1 | 1 |
| サーバの IP アドレス | 通過合計台数 | ホップカウント |

図 9 冗長データを初めに受信した TR ノードが送信する ICMP データの形式

ICMP パケットを受信した TR ノードは図 9 にある「サーバの IP アドレス」と IP ヘッダ内の宛先 IP アドレス（図 7 の場合、クライアントの IP アドレスとなる。）からそれまで仲介し

ていた通信のエンドポイントの組み合わせのうち、どの組み合わせに関するデータが送られてきたのかを知ることができる。「通過合計台数」は対応するエンドポイントの組み合わせにおいて配信データが何台の TR ノードが通過するかを表す値である。「ホップカウント」はこのパケットを何番目に受信した TR ノードであるかを示す値である。パケットを送信する TR ノードは「ホップカウント」の値を初期値として 1 に設定し、後続の TR ノードはこのパケットを受信する度にこの値を 1 加算する。

図 7 の TR ノード 1 は通過合計台数を 3 とし TR ノード 2 へ ICMP パケットを送信する。その時の処理の流れを図 10 に示す。緑で囲まれた数字は ICMP データ部にある通過合計台数を表し、青で囲まれた数字はホップカウントの値を表す。ICMP パケットを受信した TR ノード 2 はホップカウントの値を 1 加算し通過合計台数と比較する。通過合計台数がホップカウントの値より大きいため後続に TR ノードが存在するとわかる。このことから TR ノード 2 はサーバからクライアント宛への通信において自身は中継の動作を行うべきだと判断できる。そして ICMP パケットを後続へ送信する。ルータ A へ送られた ICMP パケットは TR ノード 3 へ届けられる。TR ノード 3 はホップカウントを加算し通過合計台数と比較する。通過合計台数が 3 に対しホップカウントの値が 3 となり 2 つの値が一致したため後続には TR ノードが存在せず自身が圧縮データを復元すべきだとわかる。TR ノード 3 はそのパケットを破棄し TR ノード 1 へ ICMP パケットの受信が完了したことを新たにパケットを送信し伝える。

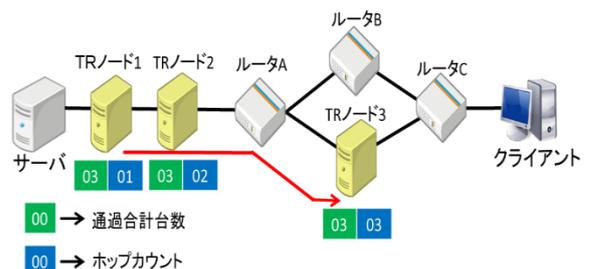


図10 TR ノードが操作する ICMP データ部の内容

なお TR ノード 1 は TR ノード 3 からの応答があるまで圧縮処理を行わない。TR ノード 1 が送信した ICMP パケットが何らかの理由でロストした場合、TR ノード 2,3 は自身が行うべき処理を判断できない。この状態で TR ノード 1 が圧縮データを後続へ送ってしまうと通信の失敗が起きてしまう可能性があるからである。

TR ノード 3 から ICMP パケット受信が完了したことが伝えられると TR ノード 1 は配信データを圧縮して送る。TR ノード 2, 3 は先ほどの処理により自身が行うべき削減処理がわかっているため、エンドポイント間で往復するパケットが往路と復路で異なる TR ノードを通過する場合も最適な動作を行うことができることになる。

4 動作検証と評価

4.1 自律的な削減処理選択機能の検証

図6のように[1]の TR ノードでは圧縮データがクライアントへ送られてしまう状況にも対応できているかの検証を検証 A、図7のように[1]の TR ノードでは削減区間を最大限取れなかった状況では今回の実装で TR ノードはどのように動作するか確かめる検証を検証 B と呼ぶ。検証 A には図11、検証 B では図12のネットワーク構成を使用した。検証 A、B ともにクライアント上では5つプロセスを起動させ5台の受信ホストを代用する。そしてサーバはクライアントの各プロセスに冗長なデータ配信を行い TR ノードが最適な削減機能を選択できているか検証する。表1に検証 A,B で使用した各マシンのスペックを表す。

検証 A ではルータがクライアントから送られてきたパケットの通過数の値を加算するようにしてある。これにより[1]の TR ノードを

11のネットワークで動作させると、TR ノード 2 は TR オプションの総数が通過数の値よりも大きいため圧縮データを中継してしまう。この状況は図6のネットワークで生じる状況と同等のものである。つまり図11のネットワークで新機能を実装した TR ノードが最適な削減処理を選択し、TR ノード 2 が圧縮データを復元することができていれば、図6のネットワークで生じる問題を解決できたといえる。検証 B では TR ノード 3 がクライアントから送られてきたパケットの通過数の値を加算しないようにしてある。これにより[1]の TR ノードを図12のネットワークで動作させると TR ノード 2 は TR オプションの総数が通過数の値と一致するため、後続に TR ノードがあるのにも関わらず復元処理を選択してしまい、削減区間を最大限にとれないという状況が発生する。この状況は図7を使って説明した状況と等しい。つまり、図12のネットワークで新機能を実装した TR ノードが最適な削減処理を選択し、削減区間を最大限にとることができていれば、図7のネットワークで生じる問題を解決できたといえる。

それぞれの検証は300秒間行い、検証 A では TR ノード 1,2 の送受信量を測定し、検証 B では TR ノード 1,2,3 の送受信量を測定した。また、図11、図12のネットワーク構成で[1]の TR ノードも動作させて新機能を実装した TR ノードの動作と比較した。

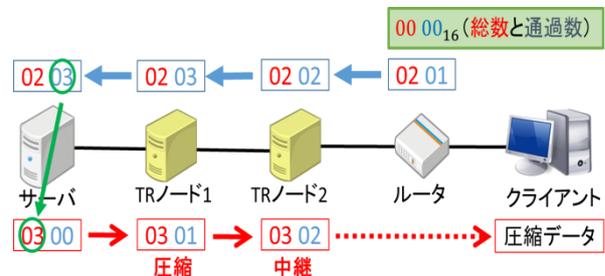


図11 検証Aのネットワーク構成

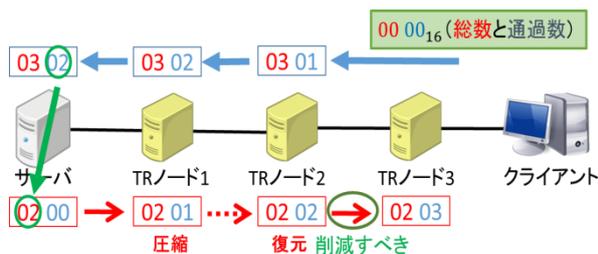


図 12 検証 B のネットワーク構成

表 1 検証 A,B で使用したマシンのスペック

| | サーバ, TRノード1, 2, クライアント | TRノード3, ルータ |
|-----|--------------------------------|--------------------------------------|
| CPU | AMD Opteron1210(1.8GHz) | Intel Celeron CPU(2.66GHz) |
| OS | Debian 4.0 (Linux2.6.18-6-486) | Ubuntu 12.04 (Linux3.0.0-32-generic) |
| メモリ | 1024MB | 512MB |
| NIC | 100BASE-TX | |

4.2 検証結果

検証 A における TR ノード 1,2 の送受信量を図 13、検証 B において本研究の TR ノードを動作させた時における TR ノード 1,2,3 の送受信量を図 14、[1]の TR ノードを動作させたときにおける TR ノード 1,2,3 の送受信量を図 15 に示した。

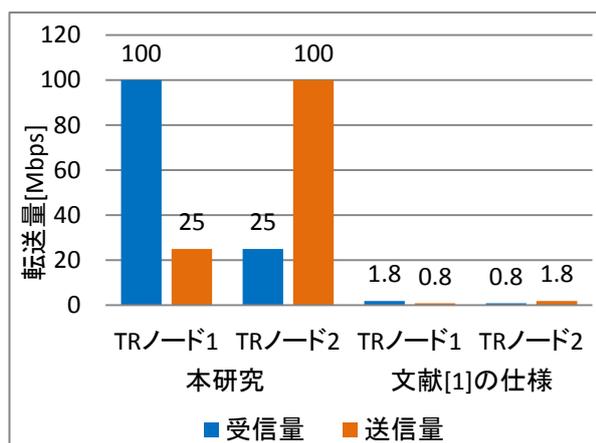


図 13 検証 A における TR ノード 1,2 の送受信量

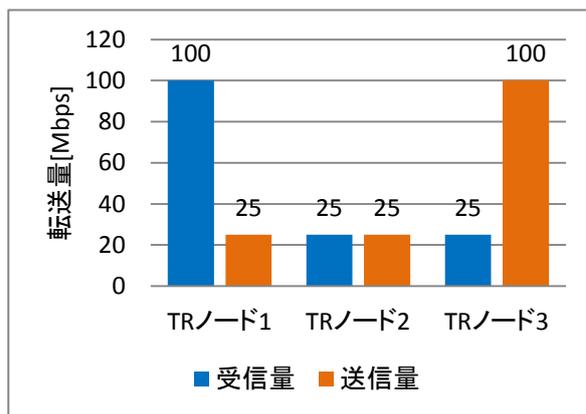


図 14 検証 B における本研究の TR ノード 1,2,3 の送受信量

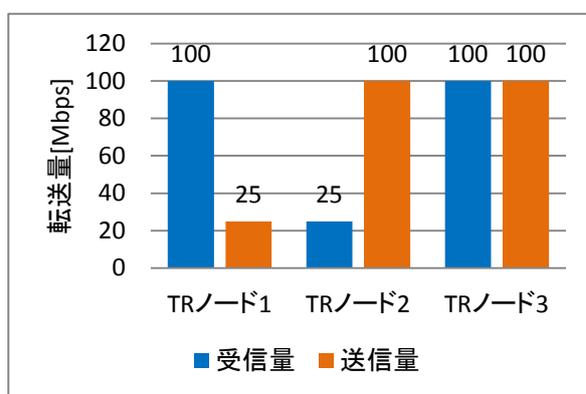


図 15 検証 B における[1]の TR ノード 1,2,3 の送受信量

図 13 より検証 A では[1]の削減処理の選択方法を用いた TR ノード 1,2 の送受信量は著しく低くなっていることがわかる。サーバが送信するパケットの TR オプションの総数には 3 が入り、[1]の TR ノードは TR オプションの情報のみから処理を選択するため、TR ノード 2 は配信データの総数と通過数の値から後続に TR ノードが存在すると判断する。そのため圧縮データがフォワードされそれがクライアントに届けられる。するとクライアントは TCP チェックサムエラーを検知しそのパケットを破棄する。これにより TCP の輻輳制御が働きサーバは配信のレートを大きく下げる。これが[1]の TR ノードの送受信量が著しく低下した理由である。しかし本研究における TR ノード 2 は圧縮データを復元できていることが図 13 から

わかる。このことから TR ノード 2 は TR ノード 1 が送信した ICMP パケットの情報をもとに自身が復元処理を行うべきであると判断し最適な処理を選択できたといえる。

図 15 から検証 B では [1] の削減処理の手法を用いた TR ノード 2 が圧縮データを復元していることが確認できる。サーバが送信するパケットの総数の値は 2 で、[1] の TR ノードならば TR ノード 2 は総数と通過数の値が一致するため後続に TR ノードがないとして復元処理を選択していた。この場合、実際は後続に TR ノード 3 が存在しており削減区間を最大限とることができていない。図 14 から本研究における TR ノード 2 の送受信量はともに約 25Mbps であり、TR ノード 2 は圧縮したデータを中継することができていることがわかる。そして TR ノード 3 は圧縮データを復元できていることも図 14 からわかる。このことから TR ノード 2,3 は TR ノード 1 が送信した ICMP パケットの情報から最適な削減処理を選択できたといえる。

5. 結論

5.1 まとめ

本研究では配信データが通過する TR ノードの台数を TR ノード間で伝え合うことにより、それぞれが最適な削減機能を自律的に選択することが可能になった。その結果、TR オプションの情報のみからでは適切な処理を選択することができなかつたネットワークにおいても TR ノードは最適な動作を行うことができるようになった。

5.2 今後の課題

図 6、図 7 のようなネットワークにおいてルータ A が配信データに対し、固定されたルーティングを行わない場合問題が起きる。ルータ A の負荷分散などの機能により圧縮データがルータ B へ届けられると TR ノードが圧縮データを復元することができず通信の

失敗が起きる。TR ノードはルータの負荷分散により配信データが通る TR ノードの台数にばらつきがあることを検知したら、TR オプションの通過数の値をこまめに監視し、確実に配信データが TR ノードを通過する区間でのみ圧縮データを流すといった対策が必要となる。

参考文献

- [1] Tomohiro Yoshida, Yamato Ikeda, Kenji Ichijo, Akiko Narita “Autonomous Decision of Function of the Redundant Traffic Reduction Node Using a New IP Option”, 2016 International Conference on Networking and Network Applications (NaNA'2016), pp.233-pp.238