

トラフィック削減用ネットワークノードの多段構成と評価

古田駿介[†] 大高友樹[†] 成田明子[†]

動画配信に代表されるように一つのホストから複数のホストへ短時間内に内容が重複する大量のデータが配信されるようになった。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、ノード間では元のデータより小さなデータに圧縮してトラフィックを削減する手法を提案し、単純なネットワーク構成においては高い削減率を得られることを示してきた。ここでは、ノードを多段化することでより複雑なトポロジーに対応させた場合の結果と問題点について述べる。

Design and Evaluation of a Network Node for Traffic Reduction

FURUTA SHUNSUKE[†] OTAKA YUKI[†] NARITA AKIKO[†]

In recent years, a large amount of duplicate contents are often delivered in a short time. For example, a video streaming server sends data to multiple clients concurrently. We proposed a network nodes for traffic reduction with packet cache. Data transported between them are changed into small identifiers. In our previous study, we obtained nearly ideal reduction rate. Then we apply our method to more complex network topology with multiple-hop by improved nodes, and present results and problems.

1. はじめに

近年、コンピュータネットワークの普及や通信機器の発達により、一つのホストから複数のホストへ短時間内に内容が重複する大量のデータが配信されることが増えている。身近な例としては、ニコニコ生放送や USTREAM などのライブ中継やニュース配信が行われる動画配信サービスがある。しかし、これらのサービスは同じ内容のデータを複数のホストに送るため、送受信ホスト間の通信帯域を冗長なデータで圧迫してしまい、その区間を使用するほかの通信の妨げとなることがある(図 1.1)。この問題を解決する代表的な手法として IP マルチキャストがある。これは、複数のホストが要求するデータを経路上の分岐点で複製して配信する手法であり、同じデータを何回も送信しなくて済むためトラフィックを削減できる。しかし、この手法では経路上のルータ全てがこの手法に対応していなければならない。また、トランスポート層プロトコルには信頼性を保証しない UDP を使用する。このため、経路の一部が対応していない場合や通信の信頼性を保証する TCP を使用する場合にはトラフィックの削減を行うことができない。

そこで[1][2][3]では、信頼性を保証する TCP を用いることができ、通信の一部区間でも適用可能な、トラフィック削減手法を提案している。この手法は、短時間内に内容が重複するデータを削減対象とし、削減区間の上流側ノードを上流ノード、下流側を下流ノードとする。各ノードに通過したデータを記録するための通信キャッシュを設け、一

度通過したデータをキャッシュに記録する。上流ノードにデータが到着したら、到着したデータとキャッシュ内のデータを探索し比較をする。もし、これらのデータが一致した場合は、元のデータより小さい復元情報として下流ノードへ送信する。そして、下流ノードでは、復元情報から元のデータに復元し、受信ホストへ送信する。これにより、ノード間のトラフィックを削減することができる。また[4][5]により、[1][2][3]の手法は先ほどの短時間内に内容が重複するデータ(以下、削減対象データ)のみが流れるようなネットワーク環境では有効であるが、削減対象データ以外のデータ(以下、削減対象外データ)が流れることも考慮して削減対象外データを増加させたネットワーク環境で評価を行った場合、この手法のトラフィック削減効果が低下してしまうことが明らかになった。これに対し[6]では、保持しておく必要のない削減対象外データを優先的に追い出すために置換アルゴリズムを変更した。これにより、トラフィック削減効果の低下を防ぐことができた。

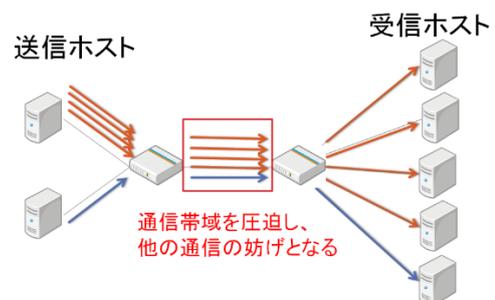


図 1.1 背景となるネットワーク構成

[†]弘前大学大学院理工学研究科
Graduate School of Science and Technology, Hirosaki University

以上の研究により、トラフィック削減を安定して行うことができるようになったが、トラフィックを削減できるのは上流ノードと下流ノードの間だけであった。本研究では、下流ノードを複数、多段に接続することによって複数のネットワーク間でトラフィックの削減を可能とすることを目的とする。

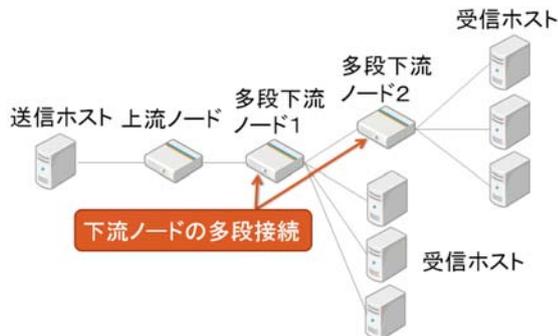


図 1.2 下流ノードを多段接続したネットワーク構成

2. 通信キャッシュによる TCP トラフィック削減手法

2.1 概要

まず、先行研究となる[1][2][3]で提案されている通信キャッシュによるトラフィック削減手法の概要を示す。この手法では、トラフィック削減区間の上流側（送信ホスト側）に位置するノードを上流ノード、下流側（受信ホスト側）のを下流ノードとし、各ノードに通信キャッシュを設ける。各ノードは受信したデータをキャッシュに記録し、上流と下流で同じデータを保持する。前章で述べた削減対象データは短時間内に内容が重複するため、各ノードにはキャッシュに記録されているデータと同じデータが再び到着する可能性がある。この特徴を利用し、キャッシュに記録された削減対象データと同じものが到着した場合、上流ノードは受信データを元のデータよりも小さな復元情報（詳細は後述）に書き換えて下流ノードへ送信する。下流ノードはその復元情報に該当するデータをキャッシュ内から探し、それを元のデータに戻す。これにより、ノード間では元のデータよりも小さな復元情報で通信できるため、トラフィックの削減が可能となる（図 2.1）。

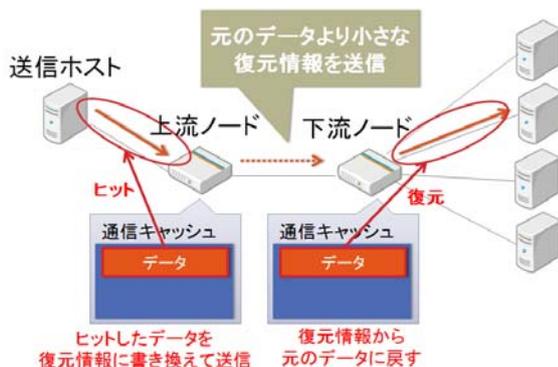


図 2.1 [1][2][3]のトラフィック削減手法の概要

2.2 通信キャッシュ

通信キャッシュは複数のエントリ（以下、キャッシュエントリ）で構成されており、各キャッシュエントリには番号が振られている。キャッシュエントリには、受信パケット（図 2.2）のヘッダ情報（送信元・宛先 IP アドレス、送信元・宛先ポート番号、TCP シーケンス番号、オフセット）、受信パケットのデータ本体（TCP データ部）、新エントリ、古エントリ、参照リストが記録される（図 2.3）。受信パケットのヘッダ情報とデータ本体は、上流ノードでデータ探索を行うときや下流ノードでデータを復元する際に用いる。キャッシュエントリ全体の数を表すキャッシュエントリ数は図 2.3 では N と表記されているが、この数は任意に指定することができる。また、新エントリと古エントリは、[2][3]の手法で用いられ、参照リストは[3]で用いられる。

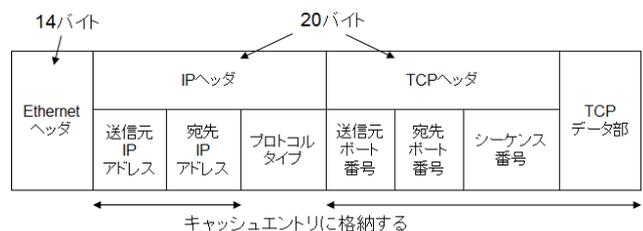


図 2.2 受信パケットの構造

キャッシュエントリ番号	送信元 IP アドレス	宛先 IP アドレス	送信元ポート番号	宛先ポート番号	シーケンス番号	オフセット	TCP データ部	新・古エントリ参照リスト (先頭・終端)
0								
1								
...								
N								

図 2.3 通信キャッシュの構造

2.3 復元情報

復元情報には、復元パケット数、復元情報の種類、復元用識別情報が格納されている（図 2.4）。この復元用識別情報には、TCP データ部が一致したキャッシュエントリの送信元・宛先 IP アドレス、送信元・宛先ポート番号、シーケンス番号などが格納される（図 2.5）。そして、下流ノードで復元情報から復元をする際に、復元用識別情報と通信キャッシュのキャッシュエントリ内の送信元・宛先 IP アドレスなどを比較し、それらが全て一致した場合は復元すべきデータが見つかったこととなり復元を行う。復元情報は最小で 31 バイトとなるため、元のデータが TCP データ部の最大である 1460 バイトであった場合、ヘッダなどを含めたパケット全体として復元情報は約 5% となり、データ転送量を削減できることになる。復元情報の詳細については[1]を参照していただきたい。



図 2.4 復元情報のフォーマット

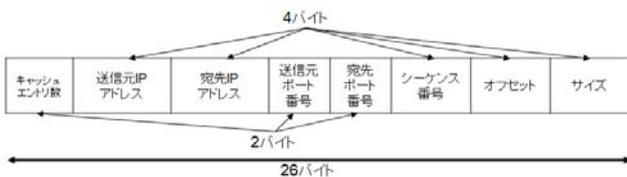


図 2.5 復元用識別情報の構造

- ※復元情報の最小サイズ 31 バイトについて
復元パケット数が 1 の場合、
- ・復元パケット数 4 バイト
 - ・復元情報種類 1 バイト
 - ・復元用識別情報 26 バイト
- の計 31 バイト

2.4 キャッシュ置換アルゴリズム

本研究のノードプログラムの動作には、上流ノードと下流ノードのキャッシュのデータが完全に同一である必要がある。さらに、キャッシュエントリ単位でも同一でなければならないため、上流ノード側でデータを記録したキャッシュエントリの番号を下流ノード側に伝え、下流ノードはその番号に同じデータを記録しなければならない。そのため、IP ヘッダのパケット長フィールドを一時的に使用して下流ノードに記録するキャッシュエントリ番号を伝えている。上流ノードでパケット長フィールドをキャッシュエントリ番号に書き換え、下流ノードでパケット長を計算し再び書き換えることで実現している。(図 2.6, 図 2.7)。

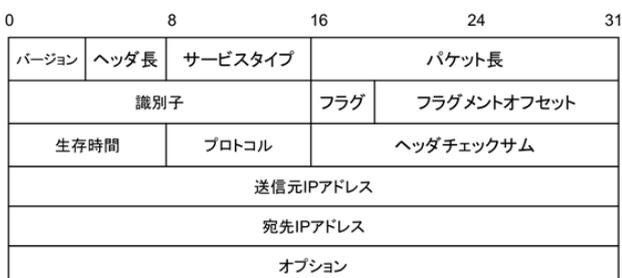


図 2.6 パケット長フィールド書き換え前の IP ヘッダ



図 2.7 パケット長フィールド書き換え後の IP ヘッダ

2.5 削減手順

図 2.8.1~2.9.4 を用いて、通信キャッシュによるトラフィック削減の手順を説明する。図 2.8.1 では、上流・下流ノードの通信キャッシュには何も記録されていない状態から開始したとする。まず、上流ノードに送信ホストからのパケットが到着したとき、上流ノードでは到着パケットの TCP データ部と、キャッシュに記録されている TCP データ部とを比較する。もし、この内容が一致しなかった場合は、上流ノードの通信キャッシュに記録する(図 2.8.1)。そして、下流ノードの通信キャッシュにも記録するために、下流ノードに送信するパケットの IP ヘッダ部のプロトコルタイプを 0xFC に書き換える。この 0xFC は通信キャッシュに記録を指示するためのものであり、下流ノードでパケットタイプが記録指示だった場合、通信キャッシュにデータ部を記録する(図 2.8.2)。本研究のノードプログラムの場合には、前述したように IP ヘッダ部のパケット長フィールドも上流ノードでデータを記録したキャッシュエントリ番号に書き換えており、上流ノードと同一のキャッシュエントリ番号にデータが記録される。そして、下流ノードから受信ホストへとパケットを送信する(図 2.8.3)。

次にキャッシュ探索で内容が一致する場合を考える。まず、上流ノードにパケットが到着したらキャッシュ探索を行う(図 2.9.1)。この場合、キャッシュ探索によって同じデータが見つかるので、送信するパケットのデータ部を復元情報に書き換え、さらに、ヘッダ部のプロトコルタイプを復元指示である 0xFD に書き換えて下流ノードに送信する(図 2.9.2)。下流ノードでは、受信パケットが復元指示と判断し、復元情報から該当するデータを下流ノードのキャッシュ内から探す。そして、見つかった場合はパケットのデータをその該当するデータに書き換えて(図 2.9.3)、受信ホストへ送信する(図 2.9.4)。このように、キャッシュヒット時には上流ノードで復元情報に書き換えたものを下流ノードで再び元のデータに復元するという処理が行われることになる。以上の様に、一度記録したデータが再び上流ノードに到着した場合は、復元情報を送ることで削減区間のトラフィックを削減している。また、[1][2][3]を含め本研究では、TCP フロー制御などによってデータ部が分割された場合でも、一部一致情報というものを用いて中間データ探索を行うこ

とにより、十分な削減効果を得られている。

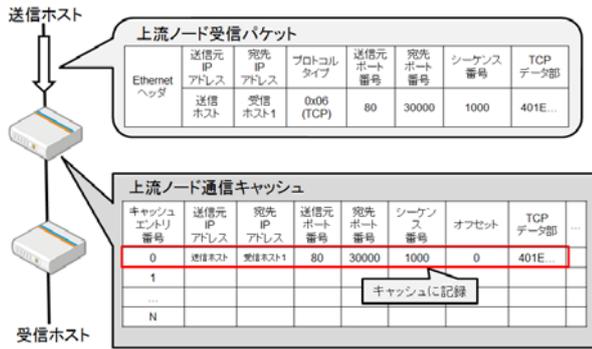


図 2.8.1

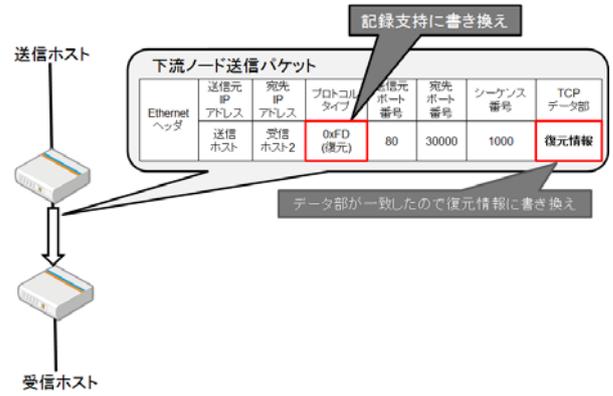


図 2.9.2

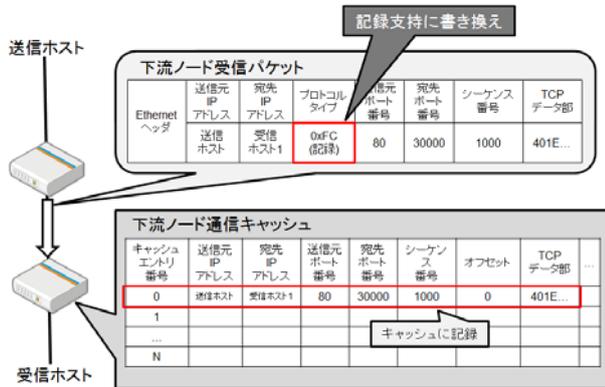


図 2.8.2



図 2.9.3



図 2.8.3



図 2.9.4



図 2.9.1

3. 多段下流ノードの機能

3.1 概要

従来の下流ノードに機能を追加し、多段接続を可能にした下流ノードを、多段下流ノードと呼ぶ。従来の下流ノードは、上流ノードから送られてきた指示に従い、データの記録や復元を行っていた。多段下流ノードは従来の機能に加え、パケットの宛先を参照し自身に直接つながっているネットワーク宛か、そうでないかを判断し、それによって動作を決定する。

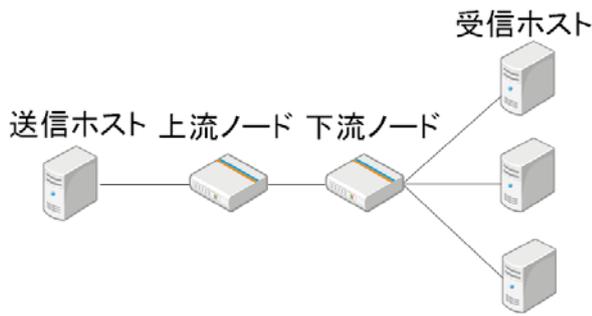


図 3.1 従来のトラフィック削減ネットワーク構成

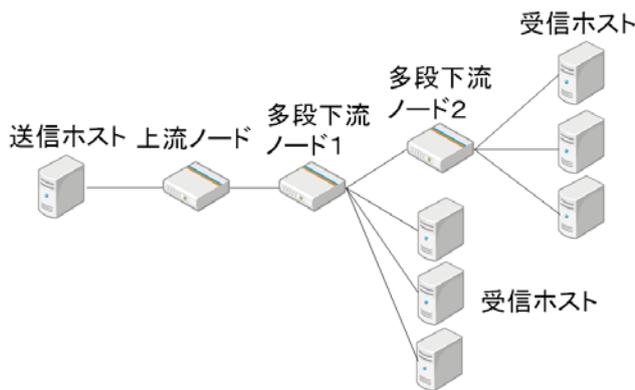


図 3.2 下流ノードを多段接続したネットワーク構成

3.2 直接つながっているネットワーク宛の場合

・記録指示

多段下流ノード1は自身キャッシュにパケットを保存してホストAに送信する。同時に、受信したパケットを複製し、多段下流ノードBに送信する。複製したパケットは記録のみの指示となっており、多段下流ノードBではこのパケットをキャッシュに記録し、送信せずに破棄する。こうすることで多段下流ノード1と2で同じ内容のキャッシュを作ることができる。そのため、次にホストB宛てに同じ内容のパケットが送られたとき、多段下流ノード1で復元せずに、多段下流ノード2で復元することができる。これで、削減区間を上流ノードから多段下流ノード2の間に拡大することができる。

・復元指示

従来と同じ動作で、キャッシュを元にパケットを復元して受信ホストAに送信する。

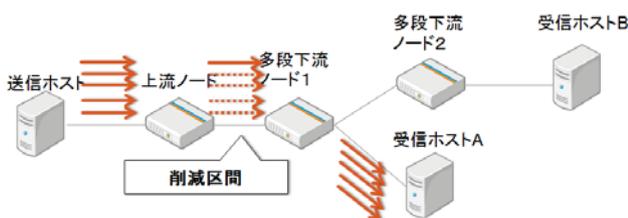


図 3.3 直接つながっているネットワーク宛の場合

3.3 直接つながっていないネットワーク宛の場合

・記録指示

多段下流ノード1は受信したパケットを自身のキャッシュに記録し、記録支持のまま多段下流ノード2に送信する。多段下流ノード2はこれを記録し、ヘッダの内容をTCPに書き戻してホストBに送信する。

・復元指示

多段下流ノード1は復元を行わず、復元情報のまま多段下流ノード2に送信する。多段下流ノード2は復元情報を受け取り、キャッシュを元に復元してホストBに送信する。この場合も削減範囲は上流ノードから多段下流ノード2の間となる。

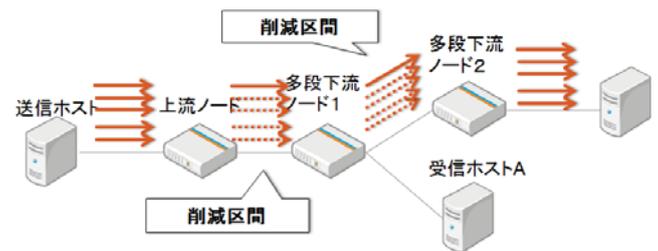


図 3.4 直接つながっていないネットワーク宛の場合

4. 動作確認

実装したノード機の評価を行うため、表 4.1 に示す機器で図 4.1 のようなネットワークを構成した。各ノードは PC を用いて構築し、100BASE-TX のイーサネット接続する。複数の受信ホストは 1 台のマシン上で 5 つの受信プロセスを同時に実行することで代用した。上流ノード、多段下流ノードは Linux アプリケーションとして実装する。この環境で送信ホストから受信ホストへデータを送信することにより、送信ホスト、受信ホスト間で同一のデータが 5 つ重複して送られる。

測定は受信ホスト A が受信する場合と受信ホスト B が受信する場合に分けて行う。測定はそれぞれ 300 秒とし、キャッシュエントリ数は 1000 とする。受信ホスト A が受信する場合は、送信ホストから上流ノードに送信されたデータ量(削減前の転送量:区間 1)、上流ノードから下流ノード1に送信されたデータ量(削減後の転送量:区間 2)を測定する。受信ホスト B が受信する場合は区間 1, 2, に加え、下流ノード1から下流ノード2に送信されたデータ量(削減後の転送量:区間 3)を測定する。

またキャッシュエントリ数を変化させたときの削減率の変化を調べるため、キャッシュエントリ数を 100 から 3000 の間で変化させてデータの削減率を測定する。このときのホスト数は 5 台とする。

表 4.1 測定に使用した機器

	多段下流 ノード2	受信ホスト A	それ以外
CPU	Intel Pentium 4 CPU (1.7GHz)	Intel Pentium G645 (2.9GHz)	AMD Opeteron1210 (1.8GHz)
OS	CentOS 5.8 (Linux 2.6.18- 308)	CentOS 5.8 (Linux 2.6.18- 308)	Debian 4.0 (Linux 2.6.18- 6-486)
メモリ	256MB	2GB	1024MB
NIC	100BASE-TX		

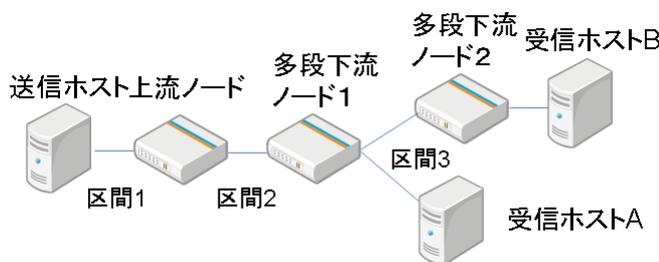


図 4.1 構築したネットワーク

5. 結果

図 5.1 にホスト A が受信した場合の区間 1, 2 のデータ通信量を示す。区間 1 の転送量は 1 秒間あたりおよそ 12Mbytes で、100BASE-TX のほぼ上限である。これに対し区間 2 では 1 秒間あたり 3Mbytes で、約 25% に削減されている。本手法では削減対象が TCP データ部のみで、ヘッダ部はそのまま送信されるため、この削減率は受信ホストが 5 台の場合のほぼ上限である。これは従来の下流ノードと同等の動作であり、従来の下流ノードと互換性があることがわかる。

図 5.2 はホスト B が受信した場合の結果をである。区間 2 と区間 3 は転送されるデータ量がほとんど同じであるため、図では重なって見えにくくなっている。ホスト A が受信する場合と同様に、区間 1 では 1 秒間に 100BASE-TX の上限の約 12Mbytes が流れ、区間 2 ではそれが 3Mbytes 程度に削減されている。区間 3 は区間 2 のデータが下流ノード 1 によってそのまま転送されるため、区間 2 とほぼ同様の転送量となっている。

図 6 はホスト B がデータを受信する場合において、キャッシュエントリ数を 100 から 3000 個の間で変化させた場合のデータの削減割合の変化である。この時の受信ホストの数は 5 台である。この場合、キャッシュエントリ数が 100 個では削減率が低く、400 個でおよそ 72% に達し、それ以降はほぼ変化しないことがわかる。これは、キャッシュエントリ数が受信ホスト数に対して少なすぎるため、キャッ

シュエントリが次々に上書きされてしまい、キャッシュヒット率が低下するためである。そのため、受信ホスト数に対してキャッシュエントリ数が少なすぎると削減割合が低下する。受信ホスト数が 5 台の場合にはキャッシュエントリ数 400 個程度で削減割合が上限近くに達し、それ以上増やしても削減割合はほとんど変化しない。

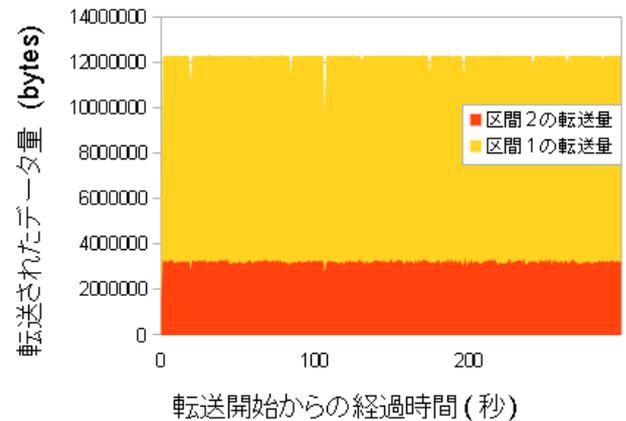


図 5.1 受信ホスト A が受信する場合の測定結果

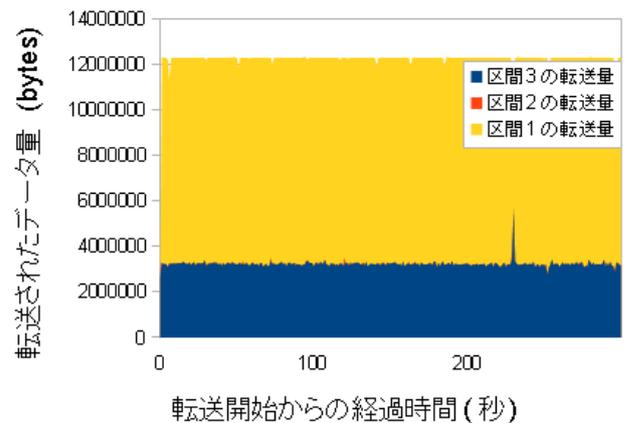


図 5.2 受信ホスト B が受信する場合の測定結果

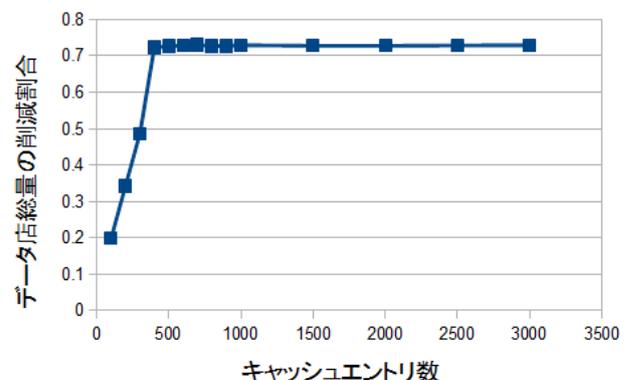


図 5.3 キャッシュエントリ数の増加による変化

6. まとめ

本研究では、[1][2][3]のトラフィック削減手法で使われていた下流ノードプログラムを改良し、多段接続機能を実装した。本研究の多段下流ノードは従来の下流ノードの機能に加え、複数の多段下流ノードによる多段接続の機能を実装した。従来の構成では上流ノードと下流ノードの間の1つのネットワークでしかトラフィック削減を行うことができなかったが、多段下流ノードにより、複数のネットワーク間でのトラフィック削減を行うことができた。多段下流ノードは受信したパケットの宛先を参照し、自身と直接つながっているネットワーク宛かそうでないかを判定し、それによって動作を決定する。また複数の多段下流ノード間でキャッシュを同一にするため、上流側の多段下流ノードは記録のためのパケットを生成し、下流側の多段下流ノードに送信する。下流側の多段下流ノードはそのパケットを受け取ると、キャッシュに内容を保存し、送信せずに破棄する。こうすることで複数の下流ノード間でキャッシュを同様にすることができ、トラフィックの削減範囲を拡大することができる。変更したプログラムを実装し実験を行った結果、従来と同様の削減率を保ちつつ、トラフィック削減範囲を拡大することができた。

今後の課題としては、下流ノードを多段にしたことで一瞬スループットが下がる現象が見られるため、その原因の究明と改善、より多くの多段下流ノードを接続した際の動作の確認などが挙げられる。

参考文献

- 1) 齋藤靖之, 「TCP/IP ネットワークにおけるデータ転送量のキャッシュによる削減」平成20年度弘前大学大学院修士論文 (2009).
- 2) 山本小百合, 「IP ネットワークにおけるデータ転送量削減プログラムの高速化」平成20年度弘前大学卒業論文 (2009).
- 3) 木村健悟, 「トラフィック削減用ネットワークノードの処理の効率化と評価」平成21年度弘前大学卒業論文 (2010).
- 4) 鹿内智也, 「トラフィック削減用ネットワークノードのキャッシュにおけるデータ記録制限に関する研究」平成22年度弘前大学卒業論文 (2011).
- 5) 近藤憲, 「トラフィック削減用ネットワークノードのキャッシュにおけるデータ追い出し優先順位に関する研究」平成22年度弘前大学卒業論文 (2011).
- 6) 大高 友樹, 「トラフィック削減用ネットワークノードにおける通信キャッシュの有効利用」平成23年度弘前大学卒業論文 (2012)