

卒業論文 2008年度 (平成20年度)

サンプリング手法を用いた
マルチキャスト通信品質把握機構の設計と実装

慶應義塾大学 総合政策学部

氏名：中里 恵

指導教員

慶應義塾大学 環境情報学部

村井 純

徳田 英幸

楠本 博之

中村 修

高汐 一紀

重近 範行

Rodney D. Van Meter III

植原 啓介

三次 仁

中澤 仁

平成21年2月10日

サンプリング手法を用いた マルチキャスト通信品質把握機構の設計と実装

本研究では、IP マルチキャスト通信の品質を把握するため、リアルタイム通信プロトコルである RTP/RTCP に加えてパケットペアを利用し、RTT、パケットロス率、送信者から受信者までの経路中にあるボトルネックリンクの帯域幅を推測するシステムを設計、実装した。

同報型コンテンツ配信サービスの増加にともない、通信基盤として大規模マルチキャスト網の必要性が高まっている。マルチキャストオペレーションの要求として、受信者数や受信品質を把握する指標を定義することが求められているが、これらを把握するには、送信者が受信者からデータ受信品質に関するフィードバックを収集する必要がある。しかし、マルチキャスト通信では、受信者数や受信品質が動的に変化するため、正確かつ確かな把握は困難である。また、大規模マルチキャスト環境下では、多数の受信者からのフィードバックが送信者に集中し、ネットワーク負荷が増大する。

本システムは、サンプリング手法を用いることで、ランダムに選択した受信者だけが自らの受信状況を検知するため、多数の受信者により引き起こされるフィードバックインプレッションの問題を回避することができる。また、マルチキャスト受信者数の規模に応じてサンプリング受信者数を変化させることで、大規模なマルチキャストネットワークにおいても、マルチキャスト通信品質状況を効率的に把握することが可能となる。

本研究により、マルチキャスト受信者のデータ受信品質の傾向が把握可能となった。

キーワード

1. 通信品質把握
2. 大規模マルチキャストネットワーク
3. サンプリング
4. フィードバックインプレッション

慶應義塾大学 総合政策学部

中里 恵

Multicast Transmission Quality Perception using Sampling Method
--

In this research, to discover IP multicast data transmission quality, mechanism to measure RTT, packet loss rate and bottle neck link bandwidth, using packet pair with RTP/RTCP is created.

Increasing multicast service requires large scale multicast networks. When operating a multicast session, prehension of multicast receiver number and receiving data quality is required. To meet this request, feedback from receivers must be gathered. However, multicast data trasmission quality changes, and therefore, getting the exact picture of the current network is difficult. Moreover, feedback from massive receivers will cause an feedback explosion, and will increase the network load.

By using a sampling method, only randomly selected receiver respond to the sender feedback request. This system avoids feedback implosion caused by massive receivers. Feedback volume can be adjusted by changing the sampling rate, depending on the number of multicast receiver. Therefore, multicast transmission quality can be effectively discovered.

This research realizes prehension of multicast data quality at the receiver.

Keywords :

1. transmission quality, 2. large scale multicast, 3. sampling,
4. feedback implosion

Keio University, Faculty of Policy Management

Megumi Nakasato

目次

第1章	序論	1
1.1	背景	1
1.2	本研究の目的	4
1.3	論文の構成	5
第2章	通信品質把握と関連技術	6
2.1	マルチキャスト通信品質	6
2.2	パケットペアスキームを利用した pathchar	7
2.3	RTP/RTCP	8
2.3.1	RTP	8
2.3.2	RTCP	9
2.3.3	RTCP 送信タイミングルール	12
第3章	既存技術の問題点	15
3.1	マルチキャスト通信品質把握	15
3.2	フィードバックインプロージョン	15
3.2.1	階層化による問題回避	17
3.2.2	分散による問題回避	18
3.3	問題点のまとめ	18
第4章	アプローチ	20
4.1	RTP/RTCP による品質把握	20
4.2	パケットペア理論	20
4.3	AMES: Adaptive Method for Estimating Multicast Group Size	22
4.4	提案手法	23
第5章	設計	24
5.1	システム概要	24
5.2	モジュール構成	25
5.2.1	ポーリングモジュール	25
5.2.2	RTP モジュール	26
5.2.3	RTCP モジュール	28
5.2.4	受信評価モジュール	28
5.3	設計のまとめ	28

第6章	実装	30
6.1	実装環境	30
6.2	実装概要	30
6.3	送信ノード部の実装	31
6.4	受信ノード部の実装	34
6.5	送信データ	35
6.6	実装のまとめ	37
第7章	評価	38
7.1	評価概要	38
7.2	評価環境	38
7.2.1	評価ネットワークの概要	38
7.2.2	機器環境	38
7.3	実験: パケットペア精度測定	40
7.3.1	計測結果	40
7.3.2	考察	41
7.4	実験 1: ダミーネットを含めたパケットペア精度測定	41
7.4.1	帯域幅 500Kbps	41
7.4.2	帯域幅 1Mbps	42
7.4.3	帯域幅 2Mbps	43
7.4.4	帯域幅 5Mbps	44
7.4.5	帯域幅 10Mbps	45
7.5	実験 2: 2台のダミーネットを含めたネットワークにおけるパケットペア精度測定	46
7.5.1	計測結果	47
7.5.2	考察	49
7.6	実験 3: ダウンストリームをボトルネックリンクとするネットワークにおける品質把握	49
7.6.1	計測結果	49
7.6.2	考察	51
7.7	実験 4: アップストリームをボトルネックリンクとするネットワークにおける品質把握	54
7.7.1	計測結果	55
7.7.2	考察	55
7.8	実験 5: 遅延を有するネットワークにおける品質把握	58
7.8.1	計測結果	58
7.8.2	考察	59
7.9	実験のまとめと考察	60

第 8 章	結論	61
8.1	まとめ	61
8.2	今後の課題	62
謝辞		63

目次

1.1	ユニキャストの概要	1
1.2	マルチキャストの概要	2
1.3	RP を中心に形成された配送木	3
2.1	pathchar 実行結果例	8
2.2	RTP パケットフォーマット	9
2.3	RTCP SR	10
2.4	RTCP RR パケットヘッダ	11
2.5	RTCP による RTT の取得	12
2.6	インターバル仮想式 1	14
2.7	インターバル仮想式 2	14
3.1	フィードバックインプロージョンの概要図	16
3.2	階層的なアプローチの概念図	17
4.1	パケットペア理論の概念図	21
4.2	パケットペアのボトルネックリンク通過時の概念図	21
5.1	設計概略図	24
5.2	RTP モジュールの概要	27
5.3	RTCP モジュールの概要	29
6.1	送信ノード部の実装概要	31
6.2	RTCP APP ヘッダ	33
6.3	受信ノード部の実装概要	34
6.4	音声入力設定コード	36
7.1	評価ネットワーク物理トポロジ	39
7.2	ネットワークトポロジ	40
7.3	パケットペアによる帯域幅計測結果のヒストグラム	40
7.4	パケットペアによる帯域幅計測結果	40
7.5	実験 1 におけるネットワークトポロジ	41
7.6	帯域幅 1Mbps の場合のパケットロス率の変化	42
7.7	パケットペアによる帯域幅計測結果 (ボトルネックリンク 1Mbps)	42
7.8	パケットペアによる帯域幅計測結果 (ボトルネックリンク 2Mbps)	43

7.9	帯域幅 2Mbps の場合のパケットロスの変化	43
7.10	パケットペアによる帯域幅計測結果 (ボトルネックリンク 5Mbps)	44
7.11	帯域幅 5Mbps の場合のパケットロスの変化	44
7.12	ダミーネット帯域制御の概念図	45
7.13	パケットペアによる帯域幅計測結果 (ボトルネックリンク 10Mbps)	45
7.14	実験 2 におけるネットワークトポロジ	46
7.15	case1 の計測帯域幅	47
7.16	case2 の計測帯域幅	47
7.17	case3 の計測帯域幅	47
7.18	case4 の計測帯域幅	47
7.19	case1 のパケットロス率の変化	48
7.20	case2 のパケットロス率の変化	48
7.21	case3 のパケットロス率の変化	48
7.22	case4 のパケットロス率の変化	48
7.23	実験 3 におけるネットワーク物理トポロジ	50
7.24	返答確率 100%	51
7.25	返答確率 80%	51
7.26	返答確率 50%	51
7.27	返答確率 30%	51
7.28	返答回数 1 回 (返答確率 100%)	52
7.29	返答回数 1 回 (返答確率 10%)	52
7.30	返答回数 3 回 (返答確率 10%)	52
7.31	返答回数 5 回 (返答確率 10%)	52
7.32	返答回数 8 回 (返答確率 10%)	52
7.33	返答回数 10 回 (返答確率 10%)	52
7.34	実験 4 におけるネットワーク物理トポロジ	55
7.35	返答確率 100%	56
7.36	返答確率 80%	56
7.37	返答確率 50%	56
7.38	返答確率 30%	56
7.39	返答回数 1 回 (返答確率 100%)	57
7.40	返答回数 1 回 (返答確率 10%)	57
7.41	返答回数 3 回 (返答確率 10%)	57
7.42	返答回数 5 回 (返答確率 10%)	57
7.43	返答回数 8 回 (返答確率 10%)	57
7.44	返答回数 10 回 (返答確率 10%)	57
7.45	実験 5 におけるネットワーク物理トポロジ	58
7.46	返答確率 100%	59
7.47	返答確率 80%	59
7.48	返答確率 50%	59

7.49 返答確率 30%	59
-------------------------	----

表 目 次

4.1	パケットペアが持つ問題点と解決手法	22
6.1	実装環境	30
7.1	VM ホスト計算機環境	39
7.2	dummynet 計算機環境	39
7.3	送信者計算機環境	39
7.4	帯域幅 500Kbps の場合のパケットロスの変化	42
7.5	帯域幅の組み合わせ	47
7.6	実験 3 におけるボトルネックリンク帯域幅の理論値	50
7.7	パケットロス率, 帯域幅, RTT の最低値, 平均, 分散 (RTCP RR 10 回)	53
7.8	パケットロス率, 帯域幅, RTT の最低値, 平均, 分散 (RTCP RR 5 回)	53
7.9	パケットロス率, 帯域幅, RTT の最低値, 平均, 分散 (RTCP RR 3 回)	53
7.10	パケットロス率, 帯域幅, RTT の最低値, 平均, 分散 (RTCP RR 1 回)	53
7.11	実験 4 におけるボトルネックリンク帯域幅の理論値	54
7.12		54
7.13	実験 5 において取得した RTT の最低値, 平均値, 分散	60

第1章 序論

本章では本論文の背景，目的を整理する．また，本論文の構成を示す．

1.1 背景

近年，IP プロトコルを用いた多様なサービスがインターネット上で展開されている．特に映像や音声，音楽，動画などのマルチメディアコンテンツ配信の要求が高まり，様々なコンテンツがインターネットを通してユーザに届けられるようになってきた．今後，現在の主流である蓄積型コンテンツ配信のみならず，ライブコンテンツのように，データ送信から受信，そして再生までの時間的制約が厳格な，リアルタイム性の高い同報型コンテンツ配信の需要が高まっていくと考えられる．

映像や音声を配信する身近な同報型コンテンツ配信の例として，ラジオ放送やテレビ放送がある．これらのサービスでは，専用の有線や無線（周波数）を利用して，全てのコンテンツを全てのネットワーク上のユーザに向けて配信する．この配信形態はブロードキャストと呼ばれ，送信者が常時一方的にデータを送出し，受信者は配信されたデータの中から自分の望むコンテンツを選択し，データを取得する．つまり，ブロードキャストは，送信者は受信者の存在有無にかかわらずデータを継続的に配信する．

回線を共有してパケット交換により通信を行うインターネットにおいて，ラジオやテレビの様なサービスを実現しようとした時，IP ブロードキャストを用いる．IP ブロードキャストでは，ユーザは必要としないデータであっても受信するため，自らが利用するネットワーク資源や計算機資源を不必要に消費する．そのため，インターネット上での広範囲な利用に不適である．特に，コンテンツ配信サービスの数が増えた場合，そしてそれぞれの

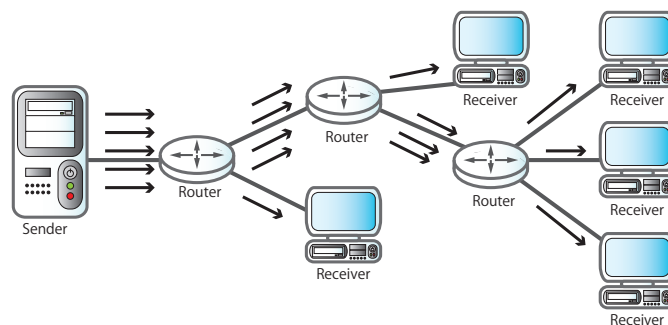


図 1.1: ユニキャストの概要

1.1. 背景

コンテンツの高品質化により消費帯域が大きくなった場合、この問題が顕著に現れる。

現在のインターネットでは、データ送信者と受信者が一対一で通信を行うユニキャストが多く用いられる。ユニキャストでは、データを必要とする受信者のみにデータ配信を行うことができるため、同報型コンテンツ配信においても不必要な帯域消費を回避することが可能である。しかし、ユニキャストでは、各受信者に対し個別にデータを送信するため、同一データを送信する場合であっても、図 1.1 の様にデータを受信者数分重複して送信する必要がある。このため、コンテンツを受信する受信者が増加するほどデータ送信サーバの負荷とデータ送信における消費帯域および転送におけるネットワークの負荷は増大する。

同報型コンテンツ配信のように、多数の受信者に対し同一データを配信する場合、送信者やネットワークの負荷を軽減し、かつ、受信を希望する受信者のみデータを受信できる効率良いデータ配信を実現することが望ましい。この要求を満たす IP 通信形態としてマルチキャストがある。マルチキャストはデータの受信希望者だけがグループを構成し、そのグループメンバーのみデータを受信することが可能な通信手法である。マルチキャスト通信では、マルチキャスト IP アドレスを送信データの IP ヘッダの宛先アドレスに指定して送出し、経路途中のルータはそのマルチキャストデータを適切に複製して各リンクに転送する。各ルータは複製と転送を繰り返し、最終的に全てのグループメンバーまでデータが伝送される。このため、データ送信者は一度のデータ送信によってデータを複数の受信者まで到達させることができる。また、マルチキャストでは、図 1.2 のようにルータがデータを複製して各リンクに転送するため、送信者とルータを結ぶネットワーク上には重複したデータが流れない。

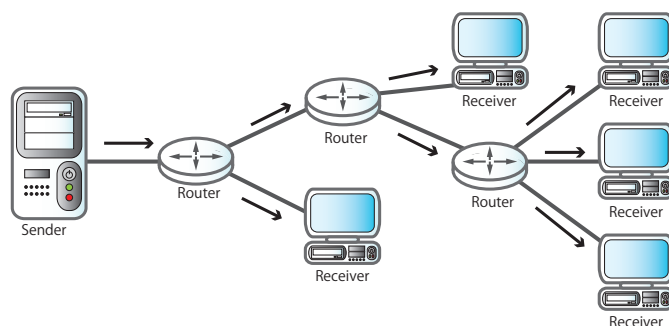


図 1.2: マルチキャストの概要

以上のように、マルチキャストはブロードキャストやユニキャストに比べ、余分な帯域を消費せず、かつ、送信者の計算機資源も受信者数に依存せずにデータ送信が行える効率的な通信方法といえる。しかし、現状のインターネットにおける同報型コンテンツ配信の多くは、ユニキャストを用いた受信者へのデータ配信により成り立っており、マルチキャストは以下の理由により局所的な利用に留まっている。

スケーラビリティの問題

まず、ルーティングプロトコル自体が複雑であり、スケーラビリティが低いという

点に関して述べる．近年のマルチキャストネットワークにおいて主に利用されている PIM-SM (Protocol Independent Multicast - Sparse Mode) ルーティングプロトコル [3] を例に挙げて説明する．PIM-SM では，マルチキャストグループアドレスプレフィックス毎に RP (Rendezvous Point) を設定し，その後，RP を中心に各送受信端末へ配送木 (RPT: Rendezvous Point Tree) を形成する．マルチキャスト送信者が RP に向かってマルチキャストパケットを送信すると，RPT 経由でマルチキャストグループメンバ全員にパケットが到達する．しかし，RP の位置によっては必ずしも配送経路が最短経路にならず，同一ネットワーク上を重複したデータが流れる可能性がある．この問題を解決するために，PIM-SM では RPT から最短経路木 (SPT: Shortest-Path Tree) に切り替える．しかし RPT から SPT へ移行する状態遷移は複雑であり，PIM メッセージ交換も頻繁に行われることから，この切り替え機能にはスケーラビリティに関する問題が多い [8] ．

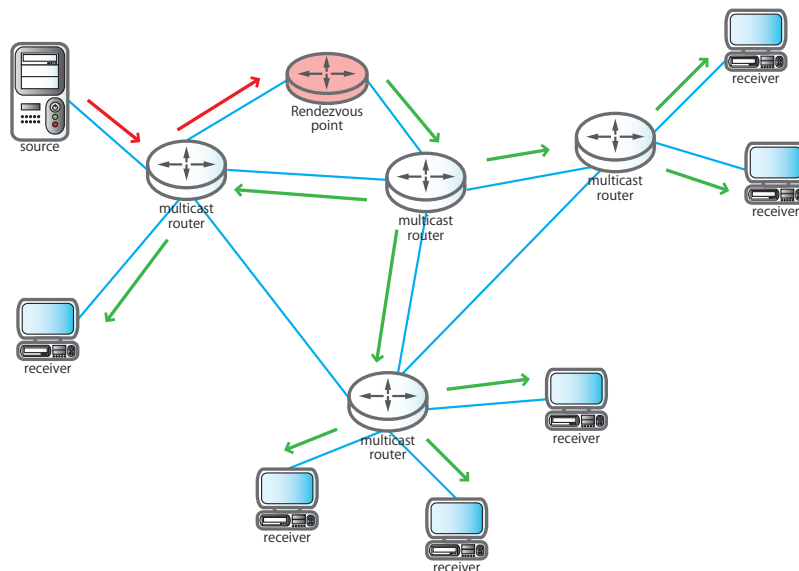


図 1.3: RP を中心に形成された配送木

運用に関する問題

次に，マルチキャストネットワークの運用に関する問題点を述べる．マルチキャスト通信では，受信者が自由にマルチキャストグループに参加・離脱し，それに応じて動的な配送木が構築される．新たな受信者が参加した場合や，受信者がマルチキャストグループから離脱した場合には，その都度，受信者に対する配送経路制御が必要になり，それまで利用していたマルチキャストグループメンバを対象とした配送木を，新しくなったグループに対して再構築する．ここで問題になるのは，マルチキャスト配送木は常に受信者の動向により変化するという点である．動的なグループの変化に合わせて配送木も変化するため，オペレーターが予期しない経路にデータが流れてしまう，あるいは特定のタイミングだけデータ流量が想定以上のものになり共有のネットワーク帯域を使い果たしてしまう可能性もある．このため，障害

1.2. 本研究の目的

発生時のトラブルシューティングだけではなく、問題の予測や対応策を検討することも困難である。

送受信制御の問題

そして、アプリケーション利用者の視点からの問題点、すなわちデータ送信者、および受信者双方にとっての問題点を述べる。マルチキャスト通信では、送信者は受信者や受信者の数、配送木の構造を知ることなく、データを送受信することができる反面、送信者が受信者を指定/特定することや通信品質を制御することは、特性上困難である。マルチキャストではトランスポートプロトコルにUDP(User Datagram Protocol) [9] を用いるのが一般的であり、送信者は自分が送信したデータが全てのグループメンバに到達したか否かを確認することができない。そして、送信者は全ての受信者がどの様な品質でデータを受信しているのか、また、データ送信帯域はどれくらいに制御すべきであったのかを判断することは不可能である。ユニキャストにおける通信品質把握では、送信者が受信者からのフィードバック情報を受け取ることにより、個々の通信品質を把握し、それに応じた制御を行い品質向上することが可能である。例えば、パケット欠損や通信障害の発見、受信者が利用しているネットワーク環境に応じた最適な配信ビットレートの発見、リアルタイム性を成立させるために必要なRTT(Round Trip Time)を満たしているか否かの確認などができる。同様の技術を用いて、マルチキャストにおいても各受信者に対して個別に通信品質把握を行う手法は存在する [4]。しかし、これは小規模ネットワークでの利用や比較的受信者数の少ないアプリケーションを想定しており、多数の受信者が存在する広域インターネット環境では、フィードバック情報が氾濫してネットワーク負荷を急激に上げるフィードバックインプロージョン(後述)という問題が発生する。結果として、各受信者のネットワークの品質をスケーラブルに把握することができない。

ここで述べた通信品質把握に関する問題は、マルチキャストアプリケーション利用の難しさを示す代表的な問題である。各ISP(Internet Service Provider)内でのサービス展開としてのみならず、インターネット全体におけるマルチキャストサービスの普及のために対応していかなければならない課題である。

1.2 本研究の目的

本研究では、マルチキャスト送信者やネットワークオペレータが、データ配信品質を把握するための枠組みを確立することを目的とする。特に、大規模マルチキャストサービスを利用したコンテンツ配信における通信品質状況を把握するため、マルチキャスト受信者数の規模に応じて品質状況を報告(フィードバック)する受信者数を変化させる。報告する受信者をサンプリングで選出し、マルチキャストセッションにおける通信品質状況を推測する機構を提案し、設計、実装を行う。

サンプリング手法を用いることで、ランダムに選択した受信者だけが自らの受信状況を通知し、フィードバック数を減少させる。結果として、送信者に対しての過剰な負荷や、その送信者が接続するリンクの負荷の増大も回避できる。

要素技術として、標準のリアルタイム通信プロトコルである RTP(Real-time Transport Protocol)/RTCP(RTP Control Protocol)に加えて、パケットペアを利用し、RTT、パケットロス率、送信者から受信者までの経路中にあるボトルネックリンクでの帯域幅を計測する。また、AMES(Adaptive Method for Estimating Multicast Group Size) [2] と呼ばれるサンプリング手法によって、マルチキャストグループの規模に応じてフィードバックを行う受信者数を変化させる。これらを組み合わせた機構を実現することで、一万以上の受信者数を有する大規模な広域マルチキャスト通信にも対応できる。送信者は受け取ったフィードバックを統計的に解析する事で、マルチキャスト通信における受信者のデータ受信品質の傾向が把握可能になる。

1.3 論文の構成

本論文の構成を以下に示す。

第 2 章では、マルチキャスト通信における通信品質の把握について整理し、既存の通信品質把握手法について述べる。第 3 章では、マルチキャスト通信におけるネットワークの品質把握における問題点を整理する。第 4 章では、本研究で提案するシステムと、それを実現するための要求事項を整理する。そして本システムの設計を第 5 章で述べる。第 6 章でシステムの実装について述べる。第 7 章では、本システムの実装に対する評価結果を示す。第 8 章に本研究のまとめを示し、結論とする。

第2章 通信品質把握と関連技術

本章では，インターネット通信における通信品質について整理する．そして，一般的なインターネットのネットワーク品質把握手法として，pathchar[11]とRTP/RTCP[4]について述べる．

2.1 マルチキャスト通信品質

ストリーミング映像配信などの同報型コンテンツ配信サービスの増加にともない，通信基盤としての大規模マルチキャスト網の必要性が高まっている．この時，マルチキャストオペレーションの要求として，受信者数や受信品質を把握する指標を定義することが求められる．特に受信品質に関して，オペレータは今後のネットワーク構築のための指針として，サービス提供者は各受信者に対する送信データの受信品質確認のために，その活用が見込まれている．

マルチキャスト受信者のデータ受信品質を把握するためには，送信者が受信者から受信品質に関する情報を収集する必要がある．品質把握を行う手法として様々なものが提案されており，ネットワーク上の途中ノードで行うものと，end-to-endで行うものの2種類に大別できる．本研究では，ネットワーク側の変更を行わず，自律分散的に運用が可能であるend-to-endの手法に着目する．end-to-endの手法により収集されるデータは，RTT，パケットロス，実効帯域などが挙げられる．

- RTT

1つの通信パケットが送信元から送信先まで伝達され，再び送信元に戻るまでに要する往復時間のことである．RTTを計測することにより，往復遅延時間が分かり，輻輳の度合いを知ることができる．また，パケットロスの指標となる．RTTは，同一の送信先まで計測した場合でも，時間によって非線形に変化するため，外れ値が存在する可能性がある．そのため，RTTを平滑化して評価する必要がある．RTTを平滑化して評価した値をRTT評価値と呼ぶ．RTT評価値の計算式を式2.1に示す． R はRTT評価値， M は新しい測定値， α は平滑化係数(推奨値は0.9)である．

$$R = \alpha R + (1 - \alpha)M \quad (2.1)$$

- パケットロス

パケットロスは，受信者に届くべきパケットが届いていないことを指す．伝送経路

における中継ルータやサーバにおいて輻輳が発生した場合に通信バッファがあふれる事で発生する。中継ルータにおいて、転送処理性能を上回るパケット数を処理する必要が生じた場合、ルータはパケットを破棄する。伝送途中でパケットが破棄されると、受信者は受信すべきパケットを取得できない。パケットロスには、受信側で対処することができず、サーバもしくはネットワークに対して対策が必要である。また、送信者は送信レートなどを調整する指標とすることができる。

- 実効帯域

インターネット上でパケットが消失する原因に、パケットの送信者から受信者までの伝送経路上で、ボトルネックとなるリンクにおける輻輳の発生がある。各ノード間で実際に利用できる帯域幅、つまりアプリケーションが利用できる帯域幅を実効帯域幅と呼ぶ。実効帯域幅に対して、過剰なトラフィックが発生した場合、輻輳が発生する。従って、実効帯域幅を知ることによって、輻輳を回避することができ、パケットロスを減少させることができる。

2.2 パケットペアスキームを利用した pathchar

pathchar は、送信ホストから宛先ホストまでに経由する各回線の経路状態を測定するツールであり、経路上の各通信路の遅延、推定バント幅、パケットロス率を調べることができる。pathchar は、traceroute を拡張する形で実装されており、TTL(Time To Live)を利用して、目的ルータまでの経路上にある各ルータに ICMP(Internet Control Message Protocol) パケットを送信し、各ルータとの RTT を計測する。様々なサイズの ICMP パケットを組み合わせる事で取得した多数の RTT データから、統計処理によって実効帯域幅を推定する。

pathchar の実行結果例を図 2.1 に示し、動作の説明をする。

I pathchar コマンドで、宛先ホストを指定する。

II 宛先ホストまでの経路上にある各ルータの IP アドレスとホップ数が対応づけられる。1, 2, 3...とホップ数の順に表示される。

III 隣接するルータ間のボトルネックリンクの帯域幅が表示される。

IV 実行結果として、localhost から指定した宛先ホストまでの合計ホップカウント数、RTT、ボトルネックリンクの帯域幅が表示される。

ただし、pathchar はユニキャストを想定して作成されたツールである。マルチキャスト環境下で用いた場合、各宛先ホストに対してパスを張り、パス毎に経路情報を把握していくため、一度に情報を把握するのは困難である。また、大量の ICMP パケットを送出するため、ネットワークに負荷がかかり、うまく作動しない場合もある。

```

ファイル(F) 編集(E) 表示(V) 端末(T) タブ(B) ヘルプ(H)
I ~# ./pathchar -m 1500 66.249.89.99
pathchar to 66.249.89.99 (66.249.89.99)
doing 32 probes at each of 45 sizes (64 to 1500 by 32)
0 localhost
| 54 Mb/s, 126 us (475 us)
II 1 203.178.128.1 (203.178.128.1)
III | 603 Mb/s, 41 us (577 us)
2 203.178.137.75 (203.178.137.75)
| 397 Mb/s, 1.22 ms (3.04 ms)
3 203.178.136.93 (203.178.136.93)
| ?? b/s, -670 us (1.68 ms)
4 203.178.136.73 (203.178.136.73)
| 2219 Mb/s, 66 us (1.82 ms)
5 203.178.136.65 (203.178.136.65)
| 256 Mb/s, 851 us (3.57 ms)
6 203.178.140.215 (203.178.140.215)
| 286 Mb/s, -710 us (2.19 ms)
7 202.249.2.189 (202.249.2.189)
| 864 Mb/s, 207 us (2.62 ms)
8 209.85.241.86 (209.85.241.86)
| 128 Mb/s, 267 us (3.25 ms), +q 3.79 ms (60.7 KB) *2
9 216.239.47.54 (216.239.47.54)
| ?? b/s, 14 us (3.21 ms), 11% dropped
IV 10 66.249.89.99 (66.249.89.99)
10 hops, rtt 2.82 ms (3.21 ms), bottleneck 54 Mb/s, pipe 22147 bytes
~# █

```

図 2.1: pathchar 実行結果例

2.3 RTP/RTCP

本節では、音声や映像などのデータをリアルタイムに配送するためのデータ転送プロトコルである RTP と、RTP データ転送を制御するプロトコルである RTCP について述べる。RTP は偶数番のポート番号、RTCP は RTP で使用したポート番号に 1 足した奇数番のポート番号を使用して通信を行う。

2.3.1 RTP

RTP は、音声や映像などのリアルタイムデータを転送するためのプロトコルである。RTP はアプリケーション層におけるプロトコルであり、パケットのシーケンス番号、タイムスタンプなどの情報を付加する。

一般的なトランスポートプロトコルに、TCP(Transmission Control Protocol) と UDP があるが、UDP を用いたデータ転送では TCP と異なり、輻輳制御や再送処理を行わない。リアルタイム性を重視した対話型の通信では、許容できる end-to-end の遅延が非常に短く、TCP の再送制御や輻輳制御による配信時間の大きな変動に対応できない。そのため、リアルタイム性を重視した場合、遅延が発生しない UDP が用いられることが多い。また、マルチキャスト通信を利用する場合も、TCP はコネクション指向のプロトコルであるため利用することができず、UDP を用いる。以上の理由から、RTP は UDP/IP 上に

構築されることが多く、タイミングの回復と欠落検出の機能を提供している。また、RTP で伝送しているデータの品質については、RTCP を併用することで制御している。

RTP ではシーケンス番号とタイムスタンプを元に、受信者側で正しい順番、タイミングにパケットを並び戻すことができる。しかし、RTP は情報を付加するのみであり、経路ネットワークや受信側の状況に応じた対応は行わない。図 2.2 に RTP ヘッダーフォーマットを示す。

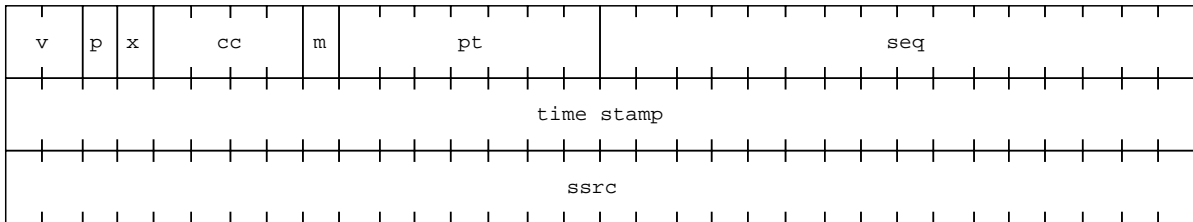


図 2.2: RTP パケットフォーマット

- シーケンス番号 (seq)

シーケンス番号はパケットの識別に使用され、受信側がパケットの欠落や正しい順番でパケットが到達しているか検知する。シーケンス番号の値は、符号なしの 16bit 整数で表現され、データパケットが送信される度に 1 ずつ加算される。最大値に達すると 0 に戻る。

- タイムスタンプ (time stamp)

RTP タイムスタンプは、パケットに含まれるメディアデータの最初のオクテットがサンプリングされた瞬間を表し、メディアデータの再生をスケジュールするために使用される。32bit の符号なし整数で表現され、値はメディアごとに異なった割合で増加する。初期値は無造作に選択された値から始まり、最大値を超えると 0 に戻る。

2.3.2 RTCP

RTCP は、RTP パケット送信者と受信者の相互間で RTP の転送統計情報を交換し、RTP のセッションを制御するためのプロトコルである。RTP の転送統計情報には、受信品質、参加者の識別、ソースについての情報、セッション内のメンバ変更通知、メディアストリームの同期に必要な情報があり、RTCP はこれらの情報を定期的にレポートする。RTCP パケットには、SR(Sender Report)、RR(Receiver Report)、SDES(Source Description)、BYE、APP(Application-Defined) の 5 種類ある。この 5 種類の RTCP パケットのうち、RTCP SR と RTCP RR がよく利用され、送信者は受信者の受信品質としてパケットロスと RTT を知ることができる。

RTCP SR

RTCP SR パケットは RTP の送信者から送信され、RTP パケットで送信されたメ

2.3. RTP/RTCP

ディアに関する情報を提供する。RTP 送信者が受信者も兼ねる場合、RTCP SR を受信すると、後述する RTCP RR パケットを RTP 送信者に送り返す。

RTCP SR パケットのフォーマットを図 2.3 に示す。

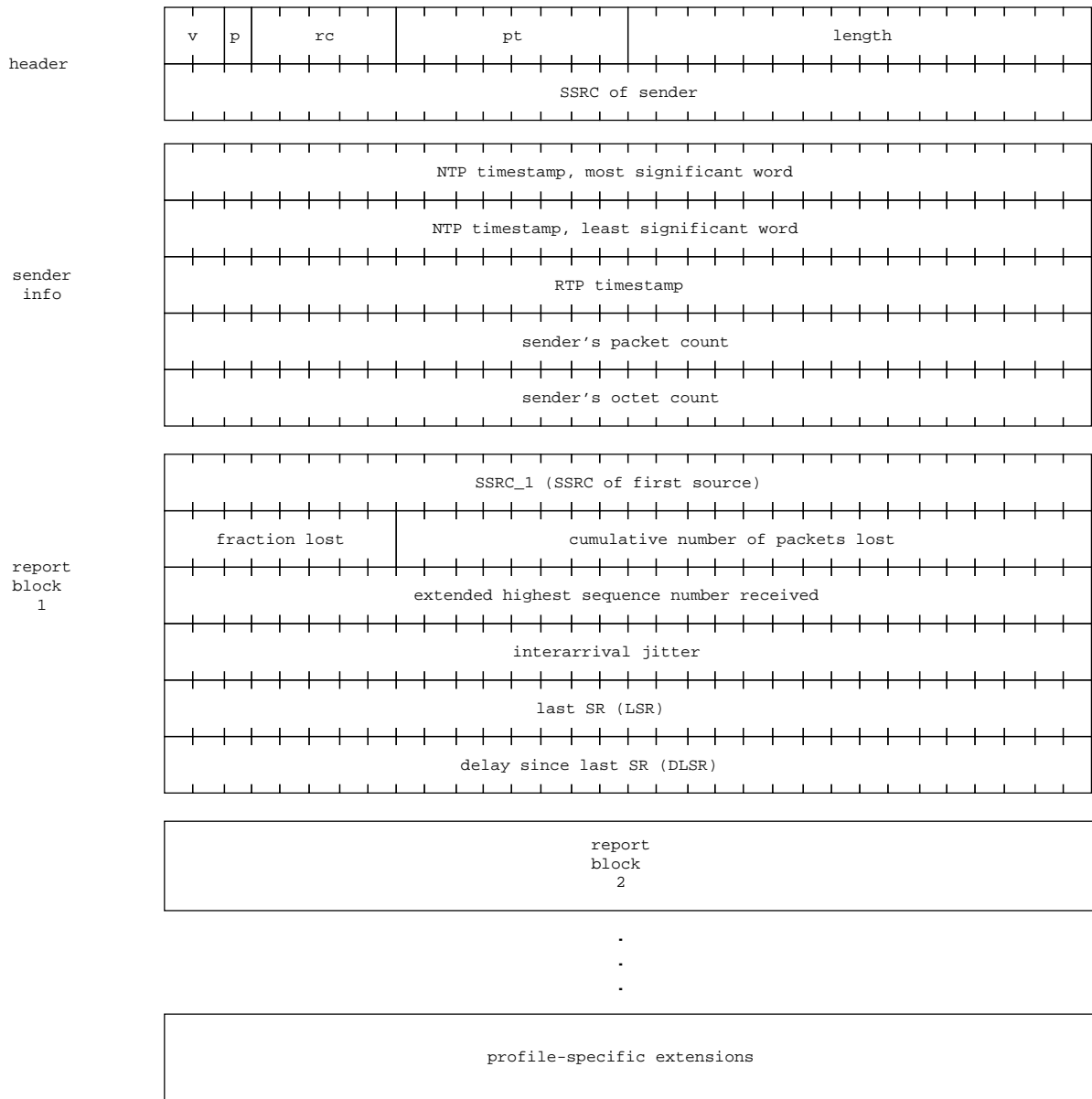


図 2.3: RTCP SR

NTP(Network Time Protocol)[6] タイムスタンプは、RTCP SR パケットが送信された時間を示す。NTP タイムスタンプの値は符号なしの 64bit 値で表現される。NTP タイムスタンプフォーマットを使用するが、クロックが NTP に同期する必要はない。

RTCP RR

RTCP RR パケットは、受信品質のレポートを目的の一つとし、データ受信者が送信する。

RTP 受信者は、RTCP SR パケットを受信すると、RTCP RR パケットを RTP 送信者に返信する。RTP 送信者は RTCP RR パケットによるフィードバックに従って、データの転送量を調整できる。マルチキャスト環境下では、衝突を避けるために、RTCP SR パケットの全受信者は RTCP RR パケットの送信タイミングを調整する。

RTCP RR パケットのフォーマットを図 2.4 に示し、注目すべきフィールドについて説明する。レポート作成者の SSRC フィールドの後に 0 以上のレポートブロックが続く。各レポートブロックは 7 つのフィールドに分かれ、RTP パケットの送信元の SSRC に対する受信品質が記述される。

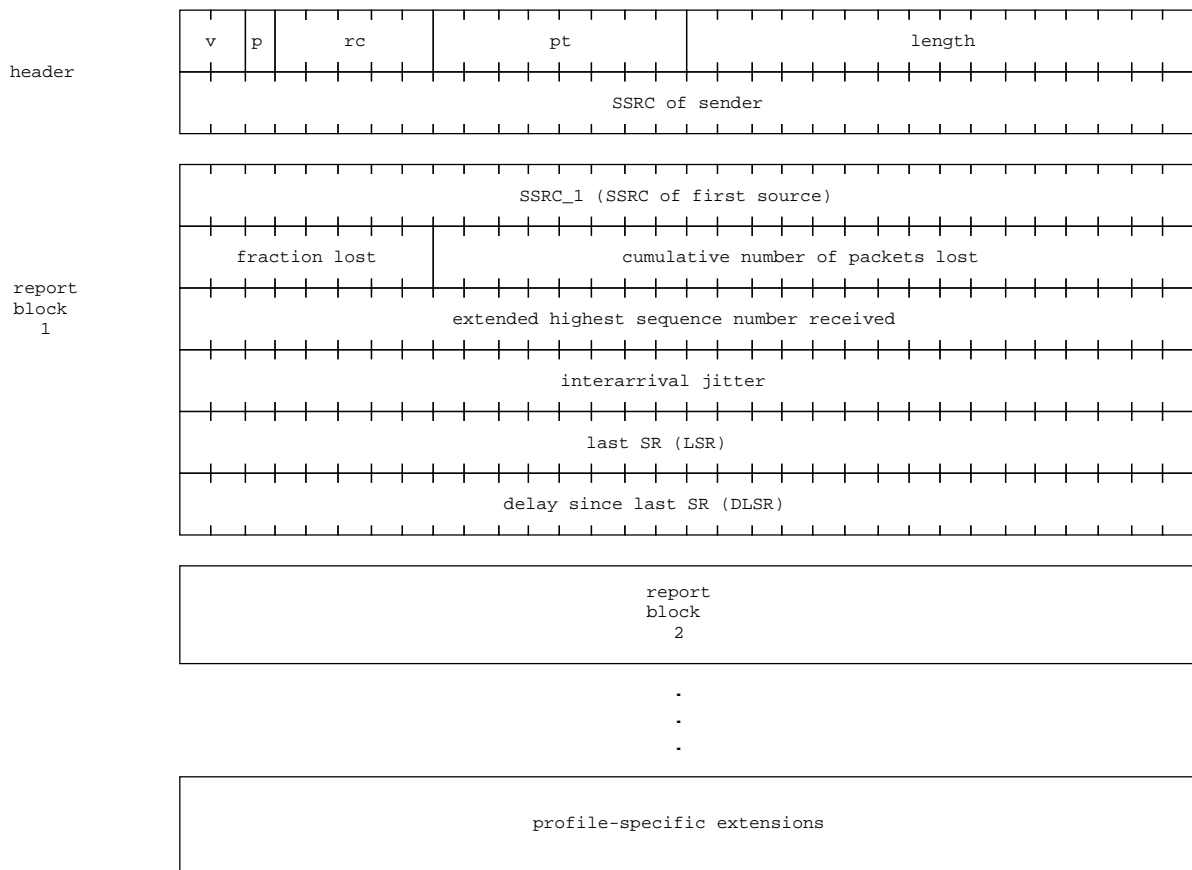


図 2.4: RTCP RR パケットヘッダ

- レポート作成者の SSRC
RTP パケットの送信元を示す識別子が記述される。
- レポート対象者の SSRC
レポートブロックが対象にしている参加者を識別するために用いる。

- 最新送信レポートのタイムスタンプ (lsr)
RTCP SR パケットの送信時間が記述される。32bit の符号なし整数で表現され、受信した RTCP SR パケットの NTP タイムスタンプフィールドに記述されている 64bit 値のうち、中央の 32bit を使用する。
- 最新送信レポート経過時間 (dlsr)
RTCP SR パケットを受信してから RTCP RR パケットを送信するまでの経過時間が記述される。

RTCP SR と RTCP RR を利用することでわかる受信品質として RTT があり、その算出方法について説明する。図 2.5 に RTT の計算方法を示す。RTT の算出には、NTP タイムスタンプフィールドを利用する。送信者は、RTCP SR パケットの送信時間を、RTCP SR パケットの LSR (Last SR) フィールドに記述する。受信者は、RTCP SR パケットを受信してから RTCP RR パケットを返信するまでの経過時間 (b) を RTCP RR パケットの DLSR (Delay since last SR) フィールドに記述する。従って、エンドポイントにおける処理時間は一切含まれない。受信者は、RTCP SR の LSR フィールドの値を RTCP RR の LSR フィールドにコピーし、RTCP RR パケットを送信する。RTCP RR パケットを受信した送信者は、RTCP SR を送信した時間と RTCP RR を受信した時間の差 (a) を計算する。図 2.5 の (a) から (b) を引いた値が RTT となる。

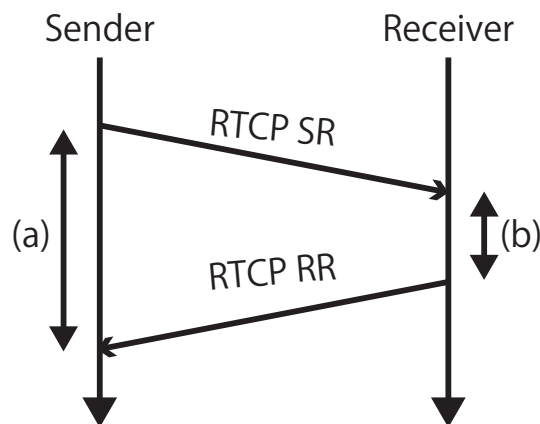


図 2.5: RTCP による RTT の取得

2.3.3 RTCP 送信タイミングルール

RTP を用いた通信では、受信者による受信品質のレポートや送信者によるメディア情報の提供や同期のために RTCP が用いられる。受信者数の増加にともない RTCP によるトラフィックも増加するため、セッションに対する RTCP トラフィックの総量が規定されている。本節では輻輳を回避するための RTCP 送信タイミングルールについて述べる。

RTP は少数から数千の参加者が同時にデータのやり取りを行えるように設計されている。多数の参加者が参加するセッションでは、音声や映像による RTP トラフィックは特定の送信者から集中的に発信される事が多い。多数の受信者に対する RTP トラフィック配信にマルチキャストが利用される場合、RTP によるデータトラフィックは受信者の増加に対して線形的に増加しない。一方で、多数の参加者は RTP トラフィックを発信せずに状態通知を行う RTCP によるコントロールトラフィックのみを送信するため、セッション参加者数が増加すると RTCP によるコントロールトラフィックが爆発的に増加してしまう可能性がある。このようなコントロールトラフィックの増加を防ぐために、各セッション参加者が自律分散的にセッションの規模を予測し、RTCP トラフィック量を抑制する仕組みが規定されている。

RTP セッションでは、RTP データトラフィックと比較して RTCP によるコントロールトラフィックは小さく、特定の割合に抑えられる。通常は RTCP トラフィックは RTP セッション全体の 5%以下に固定されている。RTCP トラフィックの占める割合は RTP のプロファイル毎に規定されているため、RTCP トラフィックが 5%以上に設定されることもある。

RTCP トラフィックの抑制は、RTCP SR 及び RTCP RR をマルチキャストで送受信し、各ノードが RTCP パケットの送信タイミングを調整することによって実現される。各受信者は、他の受信者が送信した RTCP RR の受信頻度やタイミングからセッション規模を予測する。

以下に、RTCP 送信インターバルを送信する方法を示す。

- 送信者は少なくともコントロールトラフィックの $1/4$ が配分される
- 送信者である場合、レポーティングインターバルは送信者の数に RTCP パケットの平均サイズを掛け、それを RTCP 帯域幅理想値の 25%で割った値がセットされる
- 受信者である場合、受信者の数に RTCP パケットの平均サイズを掛け、それを RTCP 帯域幅理想値の 75%で割った値がセットされる

図 2.6 に、RTCP 送信インターバルの仮想式を示す。

送信者が存在しないか、もしくは送信者が全体の $1/4$ よりも多い場合、RTCP パケットの平均サイズに参加者総数を掛け、RTCP 帯域幅理想値で割ることで、レポーティングインターバルが計算される。図 2.7 にレポーティングインターバルの仮想式を示す。

RTCP では、このように RTCP パケット送信者が送信タイミングを調整することによって、コントロールトラフィックが爆発的に増加し、輻輳が発生することを防いでいる。

```
if ((送信者 > 0) and (送信者 < (参加者の総数の 25%))) {  
    if (自分が送信者) {  
        インターバル = RTCP の平均サイズ * 送信者 / (RTCP 帯域幅の 25%);  
    } else {  
        インターバル = RTCP の平均サイズ * 受信者 / (RTCP 帯域幅の 75%);  
    }  
}
```

図 2.6: インターバル仮想式 1

```
if ((送信者 = 0) or (送信者 > (参加者の総数の 25%))) {  
    インターバル = RTCP の平均サイズ * 参加者の総数 / RTCP 帯域幅;  
}
```

図 2.7: インターバル仮想式 2

第3章 既存技術の問題点

本章では、マルチキャスト利用時におけるネットワーク品質把握の問題点を分析する。

3.1 マルチキャスト通信品質把握

受信者のデータ受信品質を把握するには、送信者が受信者からデータ受信品質に関する情報を収集する必要がある。しかし、多数の受信者を有するマルチキャストは、ユニキャストに比べ情報を収集することが困難である。

マルチキャスト通信では、受信者数や受信品質が動的に変化する。そのため、マルチキャスト送信者が、正確かつ確実に受信者数や受信品質を把握するのは困難である。

現状のインターネットで多く利用されている通信形態であるユニキャストは、TCP や UDP を用いた 1 対 1 型の通信である。TCP はコネクション型のプロトコルであり、送信者は送信したパケットについて、パケット到達確認応答 (Ack) を待ち、Ack 返答がないパケットを再送する。そのため、TCP を用いた通信の場合は、Ack パケットによる確認応答を利用して受信品質のデータを取得することができる。

1 対多の通信においても、前章で説明した RTCP RR によるコントロールメッセージを利用し、各受信者から受信品質に関するデータの取得が可能である。しかし、この場合は多数の受信者からの返信データが送信者に集中し、ネットワーク負荷を増大させてしまう。このネットワークに対する負荷を軽減するために、返信タイミングをランダムにするか、定期的なずらすことで返信を分散させるという手法が提案されているが、この方法は情報の集約に時間を要する。

3.2 フィードバックインプロージョン

マルチキャストは、受信ノードが存在するリンクに対し、ルータがデータを複製する事でリンク上を流れるデータ数を減らし、帯域幅を節約する。そのため 1 対多、多対多のコミュニケーションを目的とした通信においてユニキャストより有効な方法とされるが、UDP を用いた通信のため信頼性や通信品質を保証しない。マルチキャストでは通信品質の状況を知るために、送信ノードは受信ノードから情報のフィードバックを収集するか、パケットの配送をコントロールする必要がある。

信頼性の保証や送信ノードによるマルチキャストパケット総送信量のコントロールメッセージの調整、メンバーシップの監視のためにフィードバックは重要である。しかし、マルチキャスト通信において、多数の受信ノードからのフィードバックを監視するには限界

3.2. フィードバックインプロージョン

があり、全受信ノードのメンバーシップを把握するためにアプリケーションレベルでの実装が必要である。送信側ではフィードバックを要求するアプリケーション、受信側では定期的に受信状況を送信ノードに報告するアプリケーションの実装が要求される。

マルチキャスト送信ノードは全受信ノードにフィードバックを要請する。受信ノード数が非常に多い場合、各受信ノードが同じタイミングで一斉にデータを送信すると、送信ノードに接近するにつれフィードバックが集中し、ネットワークの輻輳が発生する。この状態をフィードバックインプロージョンと呼ぶ。フィードバックインプロージョンの概要図を図 3.1 に示す。

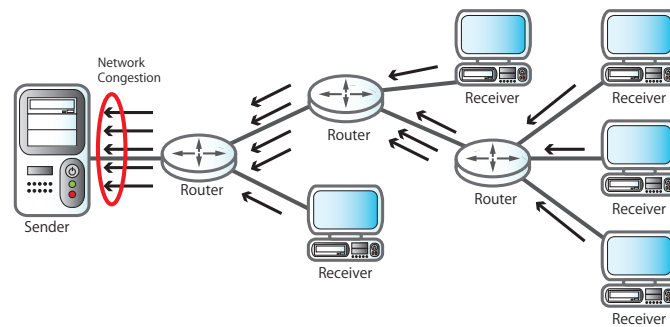


図 3.1: フィードバックインプロージョンの概要図

例えばマルチキャストでストリーミング映像を配信する場合を考える。最大帯域幅を 256Kbps、受信ノード数を 10,000 とする。全受信ノードが同時にマルチキャストグループアドレスに join し、全受信ノードがマルチキャストパケット送信ノードにフィードバックとして Ack を返信する。受信ノードからのフィードバックにそれぞれ 90byte 必要だとすると、フィードバックのトラフィックの合計は 7.2Mbps にもなる。この時、送信ノードとマルチキャスト配送木における送信ノードから最初のルータへのリンクに向かってフィードバックは集中し、フィードバックインプロージョンを誘発する。これは、送信ノード付近のネットワークにとって望ましくない結果となる。

RFC3550[4] では、RTP はマルチキャストを前提に設計されている。RTP をコントロールするための RTCP のトラフィックの上限はセッション全体の 5% とされる。一つのセッションにおける RTCP のトラフィック量を制限することでフィードバックインプロージョンの回避や、受信ノード間のフィードバック送信間隔の調整が可能となる。RTCP の転送間隔は参加者数が増加するに従って直接的に増加するため、非常に大きなマルチキャストセッションを監視する場合には、正確なセッションサイズを伝達することができない。

フィードバックインプロージョンの問題は、リライアブルマルチキャスト提案時から発見されていたが、10 年以上経った現在でも広範囲に渡り受信ノードのいるマルチキャスト通信において回避できない問題である。受信ノードからのフィードバックを必要とするようなリアルタイムのマルチキャストコミュニケーションを行う際に発生する。フィードバックインプロージョンの解決には、主に階層化によるアプローチと分散によるアプローチの 2 種類が提案されている。

3.2.1 階層化による問題回避

フィードバックインプロージョンを減少させるためには、送信元へのオーバーヘッドを減らす必要がある。階層化によるアプローチでは、マルチキャストグループ内の配送木に注目し、フィードバック情報を集約する中間ノードを配置した階層的な木構造を構築して問題解決を図る。この手法の概念図を図 3.2 に示す。

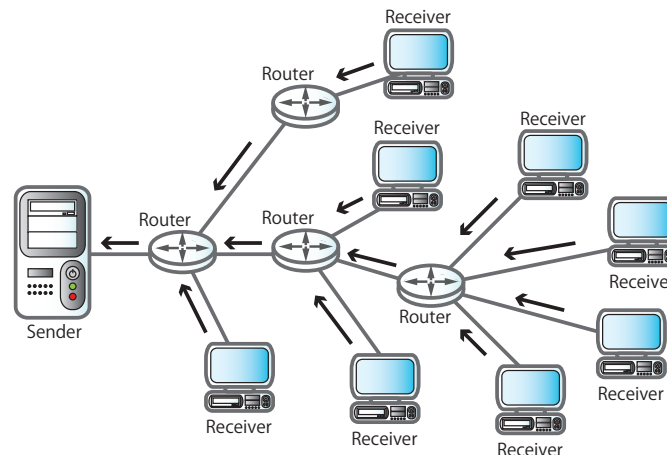


図 3.2: 階層的なアプローチの概念図

中間ノードの選出を組織ベースで行うと、ネットワークの構成要素ごとにフィードバック情報を集約できるように中間ノードを配置する。中間ノードでフィードバック情報を集約することで、より早くセッションサイズを計測できる。このため、送信ノードでのフィードバックインプロージョンを防ぎ、オーバーヘッドがない。しかし、組織ベースで中間ノードを選出すると、フィードバック木構造やパフォーマンスに依存した配置となり、ネットワーク全体の構成要素を見直す必要が生じる。全てのネットワーク上のノードに改変を加える必要があり、実装と運用のコストが高くなる。

一方、中間ノードの選出を配送木構造ベースで行う方法がある。この方法では、RTT や TTL などのフィードバック情報をもとに、フィードバック伝送用の配送木を動的に組み立て、代表の中間ノードを選出する。中間ノードは、自身の下位層にいる受信ノードからのフィードバックを一定時間待ち続ける。一定時間内に受信した自身の下位層にいる受信ノードからフィードバックを集約し、自身の上流にある他の代表ルータがマルチキャスト送信ノードに送信する。この時、自身の下位層にいる全受信ノードからのフィードバックを受け取らなくてよい。リライアブルマルチキャストにおいてフィードバックを最小化して、処理速度を速くすることに主眼をおいた手法のためである。従って、中間ノードはセッションの規模を推測することができない。階層化のアプローチでは、標準化や仕様の統一を行った後、全ネットワークルータへ新機能を実装する必要があり、既存のインターネットで実現するのは困難である。特に、インターネット上の全ネットワークルータの機能を利用する必要がある点において、ネットワーク所有者は自身のネットワークの情報開示を回避するためにルータの機能を容易に提供することはないため、実現は困難である。

3.2.2 分散による問題回避

フィードバックインプロージョンが発生する要因のひとつに、マルチキャストグループに多数のメンバが参加するという規模性の問題がある。フィードバックが全受信ノードから同時に届くのを回避するためには、フィードバックの送信時間を制御するか、フィードバックを送信する受信ノードを選択するという2つの方法がある。

時間制御手法

時間制御手法では、タイマを利用してフィードバックの送信タイミングを制御する。フィードバックは、マルチキャストアドレスに参加する全受信ノードに対して送信される。受信ノードは、他の受信ノードからフィードバックメッセージを受信すると、即座に自身のフィードバックの送信を中断し、タイマを作動させる。一般的にタイマはランダムか一定時間に設定する。タイマが満期になり、他の受信者からのフィードバックを確認できない場合、初めて自分のフィードバックが送信可能となる。この手法は、ネットワークの中間ノードでの処理や、現行のマルチキャストルーティングプロトコルの改変を必要としない。アプリケーションの実装に修正を加えるだけで実行可能であるため、簡易で柔軟な手法といえる。

しかし、時間制御手法はアプリケーションの遅延を誘発するという問題がある。Marconi Net の実験 [1] では、フィードバック取得の所要時間が長い事が紹介されている。所要時間が長いほど、フィードバックを取得している間に他セッションのトラフィックによるネットワーク状況の変化や経路変更が生じ、取得した各受信ノードの情報を比較しても、状況を把握できない可能性がある。

RTCP は時間制御手法を採用しているため、コントロールトラフィックを受け取るまでの遅延問題が発生する。このことから、RTCP を利用する場合、フィードバックインプロージョンを防ぎつつ、可能な限り迅速にフィードバックを受け取れる仕組みが必要となる。

サンプリング手法

サンプリング手法では、送信ノードが受信ノードに対し、ポーリング要求を送信してサンプリングを行う。確率因数を使用したポーリングが、参加受信ノードにマルチキャストされる。受信ノードがポーリング要求を受け取ると、要求されるパラメータを保持している受信ノードのみフィードバックを送信する。この方法は、時間制御アプローチと異なり、送信ノードがフィードバックを集中的に管理する方法である。しかし、この手法は、最適な確率因子を決定するという研究課題を有している。

3.3 問題点のまとめ

本章では、マルチキャスト通信における品質把握の問題点について述べた。受信者数が非常に多い場合、送信者が受信者のデータ受信品質を把握を試みると、受信者から送信されたデータが送信者に集中し、フィードバックインプロージョンが発生する。フィード

バックインプロージョンの回避手段は、階層化による回避と分散による回避の二種類がある。特に後者は時間制御による回避とサンプリングによる回避が含まれる。各回避手段には、それぞれ研究課題が残されている。

次章では、本章で述べた問題点を踏まえ、マルチキャスト環境下で効率的に受信者のデータ受信品質の傾向を把握する手法を提案する。

第4章 アプローチ

本章では，3章で述べたマルチキャスト通信利用時のネットワーク品質把握における問題点を解決し，効率的にネットワーク状況を把握するための手法を提案する．ネットワーク状態把握に利用する手法として，パケットペア理論と AMES について説明する．そして，提案手法を述べる．

4.1 RTP/RTCP による品質把握

2.3 節でも述べた様に，リアルタイム通信において，マルチメディアを転送するためのプロトコルである RTP と，RTP の制御機能として作用する RTCP を使用するのが一般的である．RTP/RTCP を利用して取得できるネットワーク状態の指標として，パケット欠損率や RTT がある．

しかし，RTP/RTCP の利用で取得できるのは，パケット欠損率と RTT に留まり，ネットワークの通信品質状態を把握するには不十分である．

4.2 パケットペア理論

パケットペア理論は，中継ノードが公平待ち行列を利用していることを前提として，パス上のボトルネックリンク帯域を推測する手法である．[13] この手法では，送信ノードから受信ノードまでの伝送遅延時間とパケットサイズに着目して帯域推測を行う．

図 4.1 にパケットペア理論の概念図を示す．

パケットペアでは，送信ノードはペアとなる 2 つの計測パケットを連続して送信する．公平待ち行列法では，コネクション毎にキューが割り当てられるため，ペアとなっているパケットはペアを崩すことなく一緒にキューイングされる．キューイングされたパケットは，ボトルネックリンクを通過する際，パケットの間隔が拡大される．図 4.2 のように，ペアとなるパケットのうち初めにキューイングされたパケット (A1) がボトルネックリンクを通過し，それより大きい帯域幅を持つリンクに送出された時，ペアパケットのうち後にキューイングされたパケット (A2) は未だボトルネックリンクを通過中である．ボトルネックリンク通過中，パケットの伝送速度は最も遅くなる．そのため，A1 のパケットがボトルネックリンクを通過した後，A2 のパケットがボトルネックリンクを通過し終えるまで，二つのパケットの伝送速度に大きな差が生じる．パケット間隔の拡大は，ペアとなるパケットの伝送速度の差から発生する．

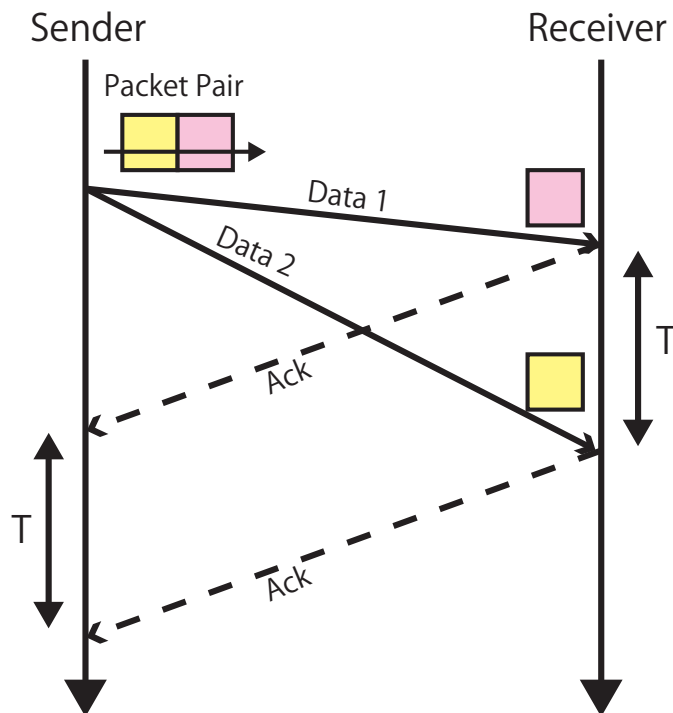


図 4.1: パケットペア理論の概念図

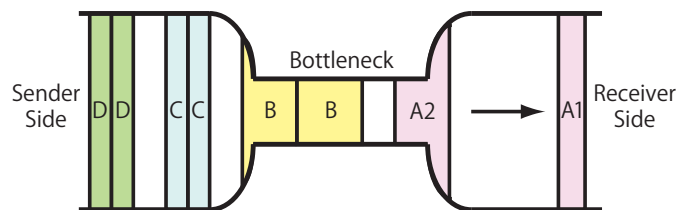


図 4.2: パケットペアのボトルネックリンク通過時の概念図

受信ノードは計測パケットを受信すると同時に送信ノードに対して Ack を送信する。Ack パケットの大きさは、データパケットに対して十分に小さいため、ボトルネックリンクにおいて拡大しないことを前提とする。送信ノードは受信ノードから受け取った Ack の RTT の差を計算し、ボトルネックリンクの帯域を推測することができる。

送信ノードが送信したデータの packetsize と RTT の時間差 T から、式 4.1 によってボトルネックリンクの帯域を算出する。

$$\text{ボトルネックリンクの帯域 (byte/sec)} = \frac{\text{データパケットサイズ (byte)}}{T(\text{sec})} \quad (4.1)$$

しかし、実際のネットワーク機器において、公平待ち行列法を採用している中継ノードの実装は少なく、FIFO(First In First Out) 型のキューイング機構を用いた実装が主流である。そのため、ペアをなす 2 つのパケットが連続的にキューイングされず、他のトラ

フィックの packets がペアとなる 2 つの packets の間にキューイングされる可能性がある。また、計測 packets の損失による計測の失敗や、受信ノードから送信ノードへの伝送遅延時間の揺らぎなど、計測結果に誤差を生む要因が多い。

このような問題を解決する packets ペア理論の応用例として、bprobe[12] というアプリケーションがある。bprobe では、表 4.1 の様にして、packets ペアが持つ問題点を解決している。

表 4.1: packets ペアが持つ問題点と解決手法

問題点	解決手法
データ packets のサイズが小さい場合、ボトルネックリンクで packets の間隔が拡大されない	転送する packets サイズを変化させる
他の packets の割り込みにより packets 間隔が拡大する	連続的に転送する packets 数を増やす
計測 packets の損失、破損	複数回計測を行う
受信ノードからの Ack packets 側の通信経路の輻輳	複数回計測を行う

解析結果によると、広域でボトルネックリンクの帯域が大きな場合に若干精度が低下したが、主に $\pm 20\%$ の範囲に 80% から 90% 程度の計測値が存在することが報告されている。以上のことから、さまざまな packets サイズを持った計測サンプル数を増加させることで計測精度を高めることができ、中継ノードが公平待ち行列法を採用していない場合でも、packets ペアによる帯域推測が有効であると言える。RTP/RTCP を利用したネットワーク通信品質把握では、RTT 値と packets 欠損値しか取得できず不十分であると考えた。本研究では、packets ペア理論を用いて、ボトルネックリンクにおける帯域幅を測定する。

4.3 AMES: Adaptive Method for Estimating Multicast Group Size

AMES[2] は、大規模マルチキャストネットワークにおける、ポーリングによる受信者数推測手法である。この手法は、フィードバックインプロージョンの抑制、受信者情報の計測における情報収束時間の短縮、高い受信者数推測性能の維持の 3 点に着目している。

AMES では、データ送信者がマルチキャストグループメンバに対し、ポーリング要求を送信する。ポーリング要求を受信したグループメンバのうち、確率変数 p を用いてフィードバックを返信する受信者が選出される。ランダムに選出された受信者からフィードバックを受信した送信者は、推測パラメータに基づいて全受信人数を推測する。また、送信者は、フィードバックからスループットや RTT を算出し、新たな推測パラメータを選別す

る．次のポーリング要求を送信する際に利用する p の値やフィードバック受信の待ち時間 T を導き出す．

この推測パラメータは，受信者数やネットワークの状態に従って動的に変化する．推測パラメータは，様々なネットワーク環境を想定したシミュレータを用いて最適な値を導き出し，実通信環境として PlanetLab[7] を用いて検証した結果，取得した値である．そのため，受信者数の推測機能を高い精度で維持することができる．AMES では，推測した受信者数と RTT の平均値・標準偏差を利用して，ポーリングを行う間隔を制御し，フィードバックインプロージョンを抑制させることができる．また，他の既存受信者数予測手法に比べ，情報収集における収束時間が短縮されるため，マルチキャストグループへのメンバの動的な参加・離脱に対応できる上，サンプリングを利用して総受信者数を推測するので，マルチキャストグループのメンバが多数の場合にも対応できるという利点がある．

RTCP を利用して受信者での受信品質に関する情報を取得しても，3.2 節で述べたように，大規模マルチキャスト環境での各受信者からのフィードバックは，フィードバックインプロージョンを誘発する．そのため，本研究では受信者のデータ受信品質のフィードバックを取得する際，AMES を用いてフィードバックインプロージョンを抑制する．

4.4 提案手法

本研究では，マルチキャスト通信において通信品質を把握するため，リアルタイム通信プロトコルである RTP/RTCP に加え，パケットペア理論を利用し，RTT，パケット欠損率，ボトルネックリンクの帯域幅を計測する機構を提案する．このとき，サンプリング手法として AMES を用いることで，フィードバックインプロージョンを回避する．また，マルチキャスト受信者数の規模に応じてサンプリング受信者数を変化させることで，大規模マルチキャストセッションにおける効率的な通信品質把握を実現する．取得したサンプリングデータから統計を取り，分布を調べることで，ネットワークコンディションの傾向を把握する．

第5章 設計

本章では、本研究で提案する機構の設計について述べる。本機構の動作をモジュールごとにまとめ、説明する。

5.1 システム概要

本機構は、1台の送信ノードと複数の受信ノードを想定し、マルチキャスト通信を用いる。ネットワークの品質状態に関する情報を提供する受信ノードと、受信ノードの情報を管理・処理する送信ノードによって構成される。ポーリングクエリを基に返信する受信ノードを選別し、選出した受信ノードのデータ受信品質に関する情報を取得する。送信ノードはこれらの情報をサンプリングデータとして収集し、統計をとる。統計からネットワークコンディションを推測する。

本システムの全体像を図 5.1 に示す。

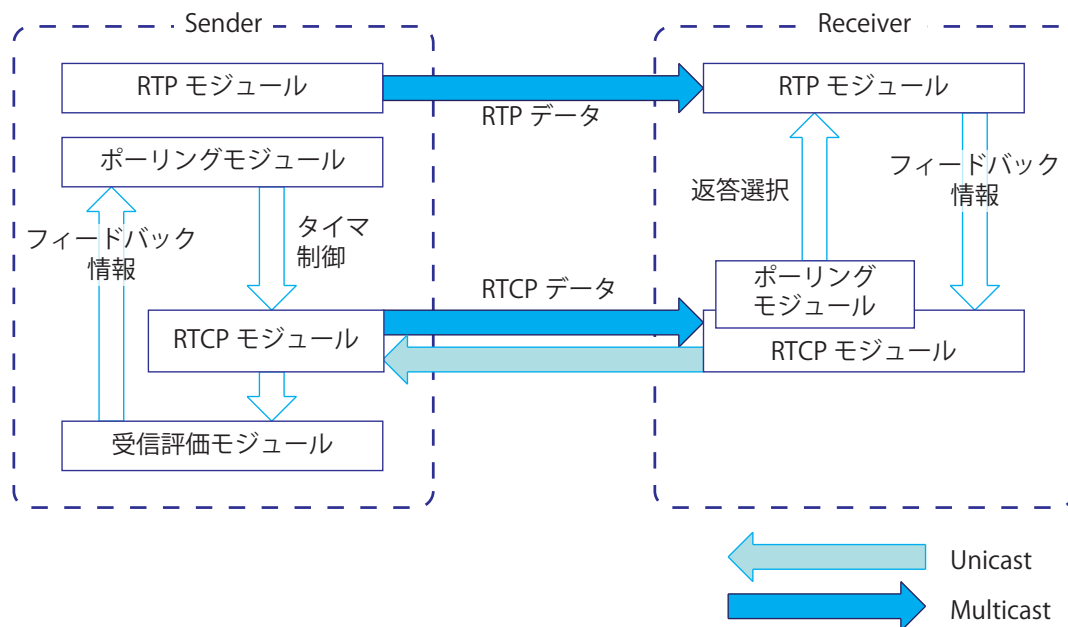


図 5.1: 設計概略図

5.2 モジュール構成

本機構は、次の 4 つのモジュールにより構成される。

- ポーリングモジュール
- RTP モジュール
- RTCP モジュール
- 受信評価モジュール

5.2.1 ポーリングモジュール

本モジュールでは、ポーリングクエリを利用して、返答する受信ノードを選別する。また、ポーリングクエリを送信するタイミングを制御する。

モジュールの構成

本モジュールでは、RTCP パケットの送信に対して制御を行う。実行する制御は、RTCP パケット返答制御と RTCP パケット送信タイミング制御の 2 種類である。

- 返答制御
マルチキャストグループメンバに対してポーリングクエリを送信するために、送信ノードは返答確率 p を設定し、RTCP モジュールに渡す。ポーリングクエリとして RTCP SR パケットを受信した受信ノードは、自身が返答するノードであるか判断する。返答ノードとして選出された受信ノードは、RTP モジュールに対し、ネットワーク状況に関する情報を要求する。このとき、要求する情報は、パケットロス率とボトルネックリンクの帯域幅である。
- 送信タイミング制御
RTCP RR パケットを受信した送信ノードは、受信評価モジュールで導き出したフィードバック情報のうち RTT を本モジュールに渡す。本モジュールへと渡されたフィードバック情報から、受信人数を予測する。予測した受信人数と RTT の平均値・標準偏差から、次の RTCP パケットの送信のための確率変数 p を算出する。また、次回送信までのインターバルを計算し、次の送信時間まで待機する。インターバルが終了すると、次の RTCP パケット送信のために、RTCP モジュールを起動させる。この時、新たに定めた確率変数 p を RTCP モジュールへ渡す。

時間制御

RTCP パケットの送信タイミングは、AMES を利用して導き出された値を使用する。

返答ノードの選別方法

AMES のアルゴリズムを用いる．返答確率 p と受信ノード自身が生成したランダムな数値 my_p を比較し， $p > my_p$ が成立した場合，受信ノードは返信を行う． $p > my_p$ の時，受信ノードは RTP モジュールに対し，ネットワークに関する情報の返答要求を行う．

5.2.2 RTP モジュール

本モジュールでは，RTP を利用してデータ配信を行う．データを受信したノードは，RTP ヘッダを用いてパケットロス率とボトルネックリンクの帯域幅を算出する．

モジュールの構成

本モジュールでは，送信ノードがマルチキャストを用いてグループメンバに対し RTP パケットを送信する．RTP ヘッダのシーケンス番号 (*seq*) フィールドを利用して，偶数番号のパケット 1 つと奇数番号のパケット 1 つでペアを作る．ペアを成す偶奇の組み合わせは，偶数番が奇数番より 1 だけ小さい値とする．たとえば，*seq2* と *seq3* でひとつのペア，*seq 54* と *seq 55* でひとつのペアを成す．ペアを成している 2 つのパケットは，時間を空けることなく連続して送信される．

パケットが受信ノードに到達すると，到達時間を確認する．そして，到達パケット数と送信ノードが送信した総パケット数からボトルネックリンク帯域幅を算出する．ポーリングモジュールから返答要求を受けた場合，これらの値からパケットロス率を導き，ボトルネックリンクの帯域幅と共にネットワーク状況の情報として RTCP モジュールへ渡す．

本モジュールの流れを図 5.2 に示す．

RTP パケット作成

本モジュールでは RTP パケット作成時に，シーケンス番号とパケット長を定める．

シーケンス番号は，本機構が起動した際に初期値を設定し，その後は 1 ずつ番号が加算されていく．初期値はランダムな値を用いるが，偶数となるようにする．

RTP パケットは，RTP ヘッダ部とデータ部から構成される．ヘッダ部は RTP ヘッダフォーマットにより定められる．データ部は，シーケンス番号の偶奇によりデータ長を変える．シーケンス番号が偶数のパケットは，データ長を 1400Byte とする．シーケンス番号が奇数のパケットは，データ長を 4Byte ~ 1400Byte のランダムな Byte 数とする．なお，データ長の上限 1400Byte は，現状のインターネットで用いられている Ethernet において，MTU(Maximum Transmission Unit) が 1500Byte 程度とされているため，RTP ヘッダとデータ長の合計が常に MTU を超えない値として定めた．

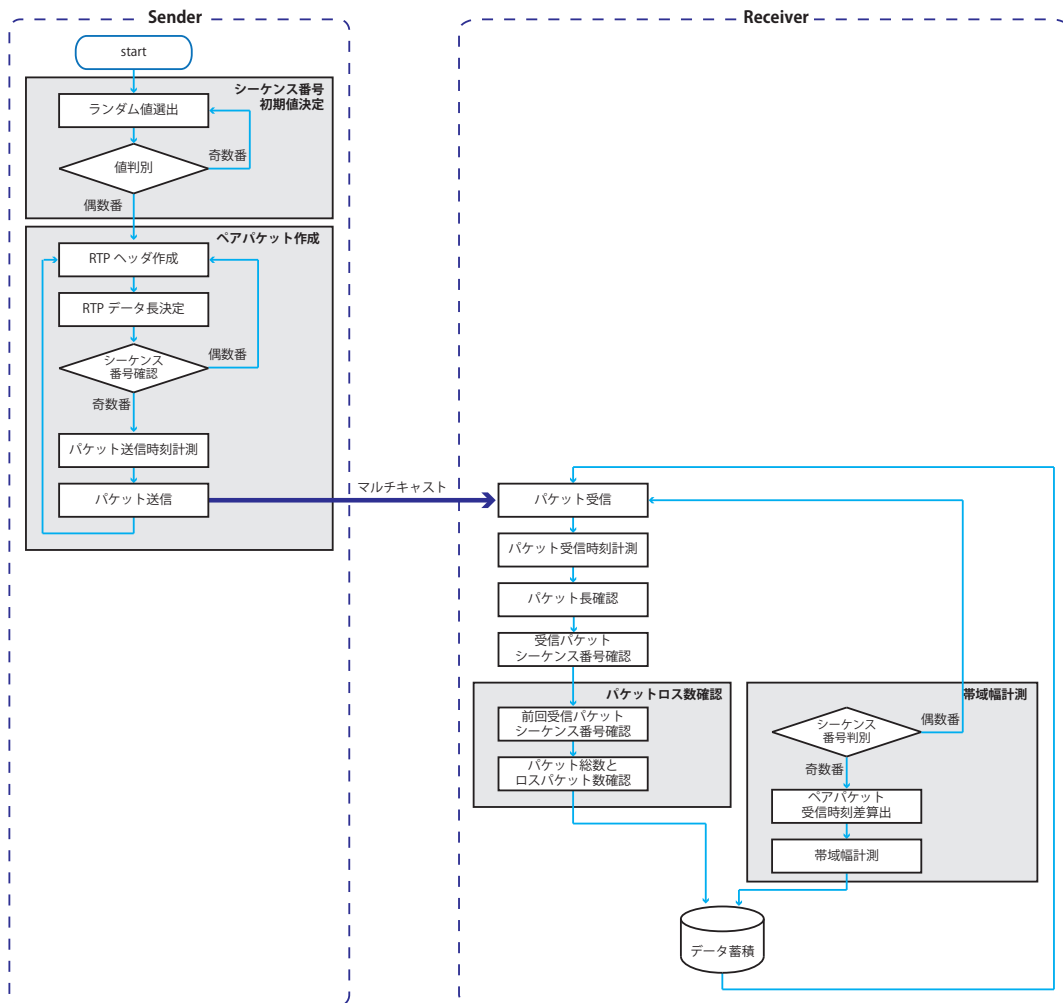


図 5.2: RTP モジュールの概要

収集する情報

本モジュールでは、受信した RTP パケットを利用して、パケットロス率とボトルネックリンクの帯域幅を算出する。

まず、パケットロス率の算出について述べる。RTP ヘッダのシーケンス番号フィールドを利用して、受信ノードで実際に受信した RTP パケットの総数と、送信側で送信された RTP パケットの総数を集計する。これらの値から、パケットロス率を計算する。

次に、ボトルネックリンク帯域幅の算出について述べる。4.2 節で述べたパケットペア理論を利用して計算する。RTP モジュールでは、RTP パケットが受信ノードに到達すると、到達時間を確認する。RTP ヘッダのシーケンス番号からペアと成るパケットを判別し、到達時間の差を計算する。また、ペアと成るパケットの内、奇数のシーケンス番号を持つパケットのパケットサイズを測定する。この、到達時間の差と RTP パケットサイズを式 4.1 に代入してボトルネックリンクの帯域幅を算出する。

5.2.3 RTCP モジュール

本モジュールは、RTCP パケットを利用し、受信ノードのネットワーク品質状態に関する情報を取得する。

モジュールの構成

本モジュールでは、送信ノードがマルチキャストで RTCP SR パケットを送信する。送信のタイミングは、ポーリングモジュールによって制御されている。RTCP SR パケットを受信した受信ノードは、自身が返信するノードであるか判断するために、ポーリングモジュールを起動する。ポーリングモジュールから返答要求がされると、RTP モジュールからパケットロス率とボトルネックリンクの帯域幅の情報が渡される。これらの情報を RTCP RR パケットとして送信ノードへ送信する。RTCP RR パケットを受信した送信ノードは、収集したデータを受信評価モジュールへ渡す。

本モジュールの流れを図 5.3 に示す。

収集する情報

本モジュールでは、RTP モジュールから渡された情報以外に各受信ノードの RTT を収集する。RTT を収集するため、4 種類の時刻情報を収集する。RTCP SR パケットが送信された時刻、各受信ノードに対して RTCP SR パケットが到達した時刻と RTCP RR パケットが送信された時刻、各受信ノードが送信した RTCP RR パケットが送信ノードに到達した時刻の情報である。

5.2.4 受信評価モジュール

本モジュールでは、RTCP モジュールから取得したデータからネットワーク状況把握のための情報を算出する。RTT、パケットロス率、ボトルネックリンクの帯域幅を取得し、統計的な処理を行う。導き出した情報のうち、RTT をポーリングモジュールへ渡す。

5.3 設計のまとめ

本章では、第 4 章で述べた提案に基づき、マルチキャスト環境においてネットワークの状況を効率的に把握する機構の設計について述べた。本機構は 4 つのモジュールから構成され、各モジュールの動作とモジュール毎の関係を詳細に説明した。

次章では本章で述べた設計を基に行った実装について述べる。

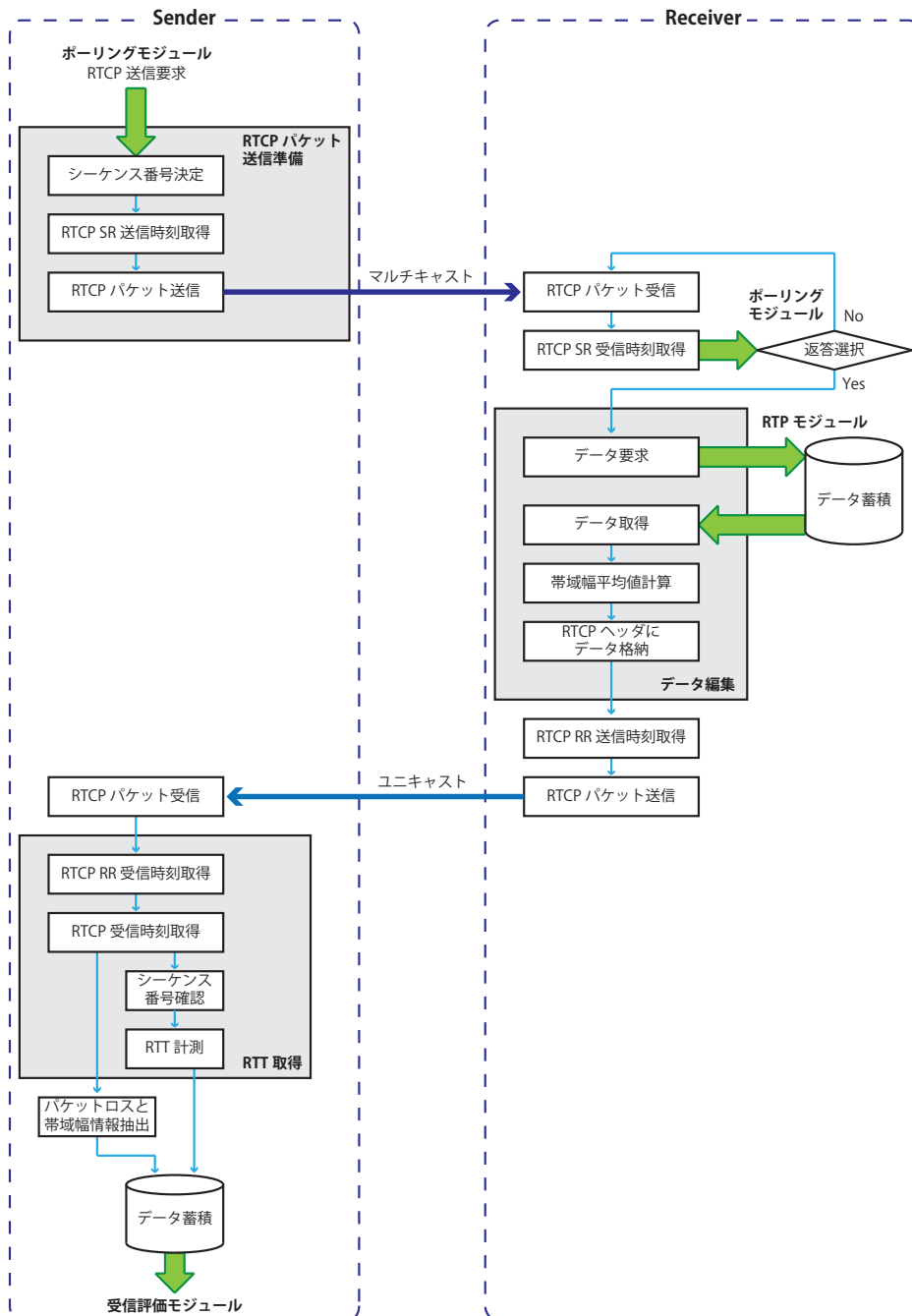


図 5.3: RTCP モジュールの概要

第6章 実装

本章では，第5章で述べた設計を基に行った実装について述べる．実装概要について説明した後，送信ノード部と受信ノード部に分けて詳細に説明する．

6.1 実装環境

表 6.1 に，本機構の実装環境を示す．

表 6.1: 実装環境

Operating System	FreeBSD 7.0-RELEASE-p3
使用言語	C 言語
コンパイラ	gcc-4.2.1

本機構は，RFC3550，RFC3551 で定義されている RTP と RTCP のメッセージフォーマットに基づいて，実装を行った．RTCP メッセージに関して，図 6.2 に示すように，アプリケーション拡張機能として定義される RTCP APP フォーマットに基づき，本機構で用いる RTCP メッセージフォーマットを定義した．

6.2 実装概要

本節では，本機構における処理の流れを述べる．

1. 送信ノードは，RTP を利用してデータ配信を続ける
2. RTP パケットを受信したノードは，データを蓄積する
3. 一定時間後，送信ノードは，受信ノードが蓄積したデータを収集するために RTCP SR パケットをマルチキャストで送信する
4. RTCP SR パケットを受信した受信ノードは，自身が返答するノードであるか確認する
5. 返信応答を行う受信ノードに選出されると，蓄積していたデータを RTCP RR パケットとして送信ノードに送信する

6. RTCP RR を受信した送信ノードは，受信データからネットワークに関する情報を取得し，受信評価を行う
7. 受信評価の結果から，次に RTCP SR パケットを送信するタイミングを決め，処理 3 に戻る

6.3 送信ノード部の実装

本節では，送信ノードにおける実装について述べる．

送信ノード部は，5.2 節で述べた，ポーリングモジュール，RTP モジュール，RTCP モジュール，受信評価モジュールの 4 つのモジュールによって構成される．

図 6.1 に，送信ノード部の実装概要を示す．

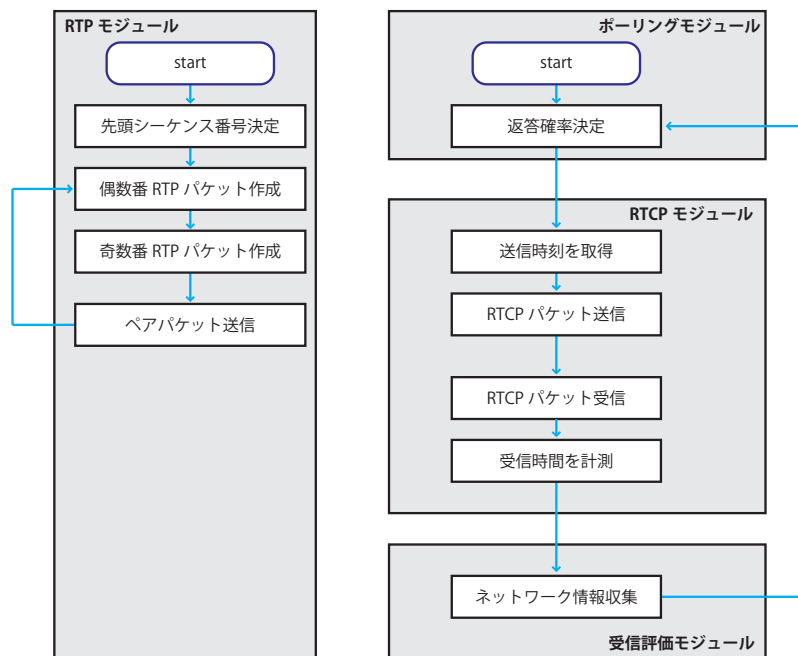


図 6.1: 送信ノード部の実装概要

ポーリングモジュール

ポーリングモジュールでは，対象とするマルチキャストグループのメンバに対し，どれくらいの割合のメンバが返答するかを決定する．送信者は，システム起動時に，最初のポーリング要求に関して，返答確率 p を設定する．返答確率 p は 1 ~ 100 の整数で設定される．2 回目以降のポーリング要求送信の際に用いる返答確率 p は，RTCP RR で収集したフィードバックの総量から導き出したマルチキャストグループ規模と，フィードバック収集の所要時間から決定する．

RTP モジュール

RTP モジュールでは、RTP を利用してデータを送信する。

各 RTP パケットには、シーケンス番号が設定されている。シーケンス番号の初期値は、rand 関数で 0 ~ 65535 の整数を発生させて決定する。RTP パケットヘッダのフォーマットによって、シーケンス番号の値は 16bit 整数値で表現されるためである。また、パケットペア理論を利用するため、シーケンス番号が偶数番とその偶数番に 1 足した奇数番でひとつのペアとして扱う必要がある。そのため、シーケンス番号の初期値が奇数の場合には、再度 rand 関数によって整数を発生させ、偶数値が発生されるまで待つ。

RTP パケット長は、シーケンス番号の偶奇により異なるように設定する。パケットペア理論を利用する際、さまざまなサイズのパケットを送信することで、帯域幅計測の精度を高められるからである。シーケンス番号が偶数番のパケットは、常にデータ長が 1400Byte になるように設定する。シーケンス番号が奇数番のパケットの場合、データ長は rand 関数で 4 ~ 1400 の整数を発生させて決定する。

RTCP モジュール

RTCP モジュールでは、RTCP パケットを利用して、受信者から受信品質に関する情報として、RTT とパケットロス数、及び帯域幅を取得する。

アプリケーションで機能拡張 (APP) を定義できるように用意されている RTCP APP のヘッダを利用する。本機構では、RTCP SR フォーマットを基本にして APP フォーマットを作成し、パケットロスと帯域幅を格納するフィールドを追加した。図 6.2 に、本機構で使用した APP パケットのフォーマットを示す。APP-SR(sender report) で示される部分が送信者から送信される RTCP パケットの構造体である。

受信品質把握のための情報のうち、RTT の取得方法について説明する。

RTT は、図 6.2 に示された `rtcp_t` 構造体の RTCP APP-SR ヘッダ内 `send_sec`、`send_usec`、及び `rtcp_rr_t` 構造体の `sec`、`usec` フィールドを利用して計測する。RTT の計算式を、式 6.1 に示す。ただし、a:RTCP SR 送信時刻、b:RTCP SR 受信時刻、c:RTCP RR 送信時刻、d:RTCP RR 受信時刻とする。

$$\begin{aligned} RTT &= (d - a) - (c - b) \\ &= d - (a + c - b) \end{aligned} \tag{6.1}$$

本実装では、式 6.1 のうち、`send_sec` フィールドと `send_usec` フィールドを用いて a の値を、`sec` フィールドと `usec` フィールドを用いて `a + c - b` の値を示した。従って、RTCP RR パケットを受信した送信ノードは、パケットの受信時刻:d と、`sec` フィールド及び `usec` フィールドの値を用いて RTT を計算することができる。

受信評価モジュール

受信評価モジュールでは、RTCP モジュールで取得した、RTT とパケットロス、及び帯域幅といったネットワーク品質情報を管理する。

```

typedef struct {
    unsigned int version:2; /* protocol version */
    unsigned int p:1; /* padding flag */
    unsigned int count:5; /* varies by packet type */
    unsigned int pt:8; /* RTCP packet type */
    u_int16 length; /* pkt len in words, w/o this word */
} rtcp_common_t;

typedef struct {
    u_int32 packetloss; /* number of packetloss */
    unsigned int fraction:8; /* fraction lost since last SR/RR */
    int lost:24; /* cumul. no. pkts lost (signed!) */
    u_int32 totalpacket; /* number of total packets */
    u_int32 bandwidth; /* bandwidth */
    u_int32 sec; /* time arrive RTCP SR to send RTCP RR (sec) */
    u_int32 usec; /* time arrive RTCP SR to send RTCP RR (usec)*/
} rtcp_rr_t;

typedef struct {
    rtcp_common_t common ; /* common header */
    union {
        /* APP-SR(sender report) */
        struct {
            u_int32 ssrc; /* sender generating this report */
            u_int32 send_sec; /* send timestamp (sec) */
            u_int32 send_usec; /* send timestamp (usec) */
            u_int32 p; /* reply probability */
            rtcp_rr_t rr[1]; /* variable-length list */
        } app_sr;

        /* APP-RR(reception report) */
        struct {
            u_int32 ssrc; /* sender generating this report */
            rtcp_rr_t rr[1]; /* variable-length list */
        } app_rr;
    } r;
} rtcp_t;

```

図 6.2: RTCP APP ヘッダ

6.4 受信ノード部の実装

本節では、受信ノードにおける実装について述べる。受信ノード部は、5.2節で述べた、ポーリングモジュール、RTPモジュール、RTCPモジュールの3つのモジュールによって構成される。

図6.3に、受信ノード部の実装概要を示す。

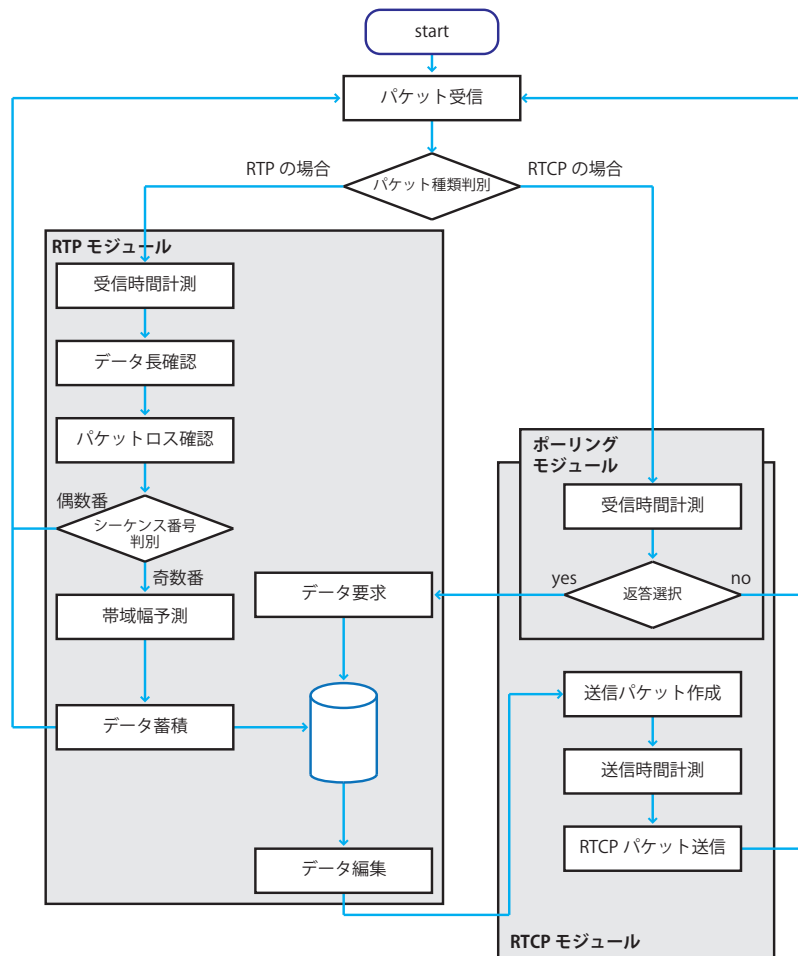


図 6.3: 受信ノード部の実装概要

ポーリングモジュール

ポーリングモジュールでは、ポーリングクエリとしてRTCP SRを受信した受信者が、RTCP RRの返答選択を行う。受信者は、ポーリングクエリを受信すると、rand関数で0~100の整数を発生させて、返答変数 my_p を決定する。一方で、送信者が要求する返答確率 p は百分率で示される。 p と my_p を比較し、 $p > my_p$ が成立した場合、RTPモジュールへデータ要求を行い、不成立の場合は、パケット受信待ち状態に戻る。

RTCP モジュール

RTCP モジュールでは、RTP モジュールからパケットロスと帯域幅の情報を取得し、RTCP RR パケットとして送信者に送信する。取得したデータは、図 6.2 の `rtp_rr_r` 構造体のうち、`packetloss` フィールド、`totalpacket` フィールド、`bandwidth` フィールドに格納される。また、RTCP RR パケット送信時に送信時刻を確認し、式 6.1 のうち $a + b - c$ の値を `sec` と `usec` フィールドに格納する。

RTP モジュール

RTP モジュールでは、RTP を利用してデータを受信する。

RTP パケットを受信すると、受信時刻を取得し、RTP データ部は音声データとして再生される。そして、パケットロスと帯域幅をデータとして蓄積する。これらのデータの算出方法について述べる。

パケットロス数は、RTP ヘッダのシーケンス番号フィールドを利用して計算する。RTP モジュールでは、最後に受信した RTP パケットのシーケンス番号を `last_seq` 関数に格納する。`last_seq` の初期値は -1 とする。一方、RTP ヘッダのシーケンス番号フィールドに格納された値を `seq` とする。RTP パケットの受信者は、`seq` と `last_seq` の値の差分を計算して、1 より大きい場合、差分から 1 引いた数をパケットロス数とする。そしてパケットロス数の計算が終了すると、`last_seq` に `seq` の値を代入して、次のパケット受信に備える。なお、`last_seq = -1` の場合は、パケットロス数の計算を行わず、`seq` の値を代入する。同様に、送信者が送信した RTP パケット数の合計も計算する。`seq` と `last_seq` の差分だけ、`last_seq` のパケットを受信してから、最新のパケット受信時まで、送信者は RTP パケットを送信している。

帯域幅は、RTP ヘッダのシーケンス番号を確認して、偶数番 (`even_seq`) と偶数番に 1 足した奇数番 (`odd_seq`) でひとつのペアを作り計算する。ペアの番号 (`pair_num`) は、偶数番を 2 で割った値とする。`even_seq` の RTP パケットを受信した場合、受信時刻を記録して、次の RTP パケットの受信待ち状態に戻る。`odd_seq` の RTP パケットを受信した場合、受信時刻を記録して、`odd_seq - 1` 番の `even_seq` 受信時刻を参照し、時間差 (`time_lag`) を計算する。また、`odd_seq` のパケット長を `odd_recv_lenByte` とする。このとき、帯域幅は、 $odd_recv_len \div time_lag$ で計算される。

本モジュールでは、RTCP モジュールからデータ要求があった場合に、蓄積していたデータを RTCP モジュールへ渡す。RTCP モジュールへ渡すデータは、RTP モジュールで蓄積していたデータのうち、パケットロス数、受信すべき総パケット数、帯域幅の平均値の 3 つのデータである。

6.5 送信データ

本節では、RTP パケットによって配信されるデータについて述べる。

本機構は、RTP を使用した音声配信アプリケーションである。送信側マイクで入力した音声データを受信側スピーカーから出力する放送型アプリケーションとして実装を行った。

6.5. 送信データ

本実装で利用した音声データフォーマットはPCM(Pulse Code Modulation : パルス符号変調)である。RTP上にダミーデータを乗せる方式ではなく、CBR(Constant Bit Rate)となるPCM音声データを利用することによって、安定したデータ生成と配信を実現した。本実装で利用した音声データフォーマットを以下に示す。

- PCM 16bit
- 2チャンネル
- 44.1kHz

本実装では音声入力を行うために/dev/dsp0.0を利用した。図6.4ファイルディスクリプタに対する設定コードを示す。

```
#define AUDIODEV "/dev/dsp0.0"

if ((rtp_fd = open (AUDIODEV, O_RDONLY)) < 0) {
    perror ("open");
    return 1;
}

opt = 16;
if (ioctl (rtp_fd, SOUND_PCM_WRITE_BITS, &opt) != 0) {
    perror ("ioctl bits");
}

opt = 44100;
if (ioctl (rtp_fd, SOUND_PCM_WRITE_RATE, &opt) != 0) {
    perror ("ioctl bits");
}

opt = 2;
if (ioctl (rtp_fd, SOUND_PCM_WRITE_CHANNELS, &opt) != 0) {
    perror ("ioctl bits");
}
}
```

図 6.4: 音声入力設定コード

6.6 実装のまとめ

本章では、送信部と受信部の具体的な実装について述べた。各モジュールの実装について詳細に示し、送信部・受信部毎に各モジュールの関係を整理した。

第7章 評価

本章では，本研究で実装した機構の評価について述べる．

7.1 評価概要

本システムでは，送信ノードが，マルチキャストグループに参加するメンバのうち選別された各受信ノードからフィードバックを取得する．本実験は，送信ノードが受信したフィードバックが，実際に利用したマルチキャストネットワークにおける受信ノードのデータ受信品質を反映している事を示すためのものである．

評価内容として，以下の二種類の実験を挙げる．

- 実験環境に関する考察
- 受信品質把握手法の評価

本システムの正当性を検証するために，様々な実験ネットワークを構築し，実験パターンを変化させて結果を取得する．

7.2 評価環境

7.2.1 評価ネットワークの概要

評価は，図 7.1 に示す環境を用いて行う．

本評価環境は，1 台の送信ノードと，VMware Server を用いて論理的に仮想化されている複数の受信ノードによって構成されている．送信ノードと受信ノードの間には，Dummynet を用いたネットワーク制御ホストがあり，ノード間の通信環境を制御する．物理的な接続は，図 7.1 に示す形態となるが，各ノード間の接続は，VMware の仮想ネットワーク技術を用いることにより物理的な接続とは異なる形態をとる．

7.2.2 機器環境

計測を行った計算機のハードウェア環境を，表 7.1，表 7.2，表 7.2.2 に示す．

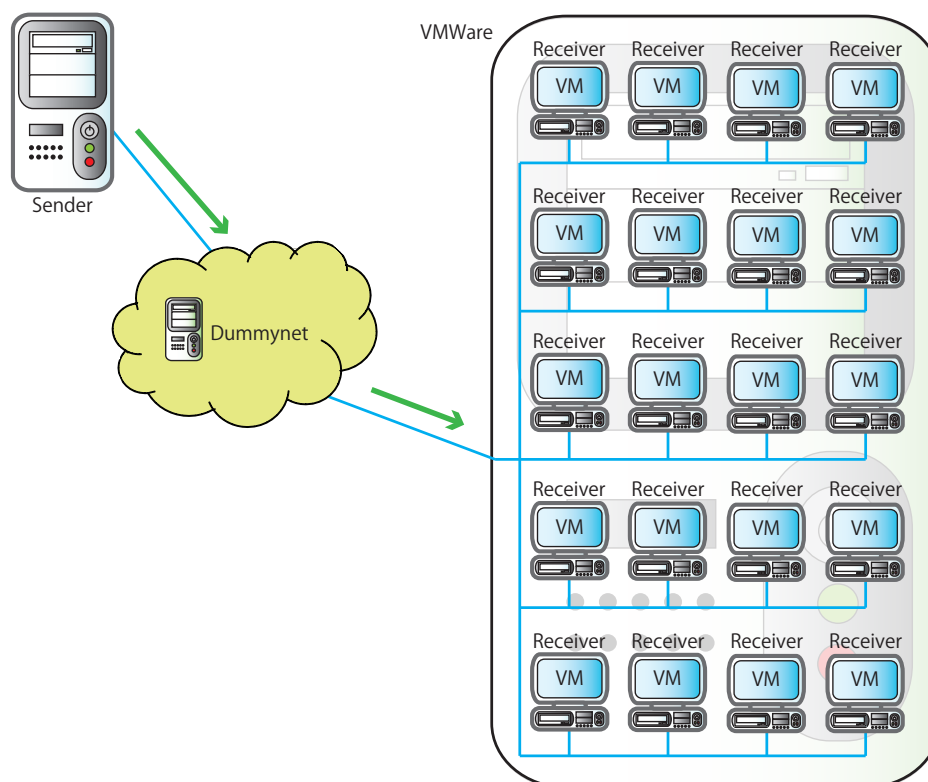


図 7.1: 評価ネットワーク物理トポロジ

表 7.1: VM ホスト計算機環境

CPU	Intel Xeon E5430 2.66GHz
メモリ	4GB
NIC	Broadcom NetXtreme II BCM5708 1000Base-T
OS	CentOS 5 Kernel 2.6.18-92

表 7.2: dummysnet 計算機環境

CPU	AMD Athlon 64 Processor 3500+
メモリ	2GB
NIC	Intel PRO/1000 1000Base-T
OS	FreeBSD 7.0-RELEASE

表 7.3: 送信者計算機環境

CPU	Intel Celeron 1GHz
メモリ	256MB
NIC	RealTek 8130 100Base-T
OS	FreeBSD 7.0-RELEASE

7.3 実験: パケットペア精度測定

実験環境に関する考察として、パケットペアを用いた帯域幅の計測精度を評価する。送信ノードと受信ノードをひとつのセグメント上に配置したネットワークにて評価実験を行った(図7.2)。

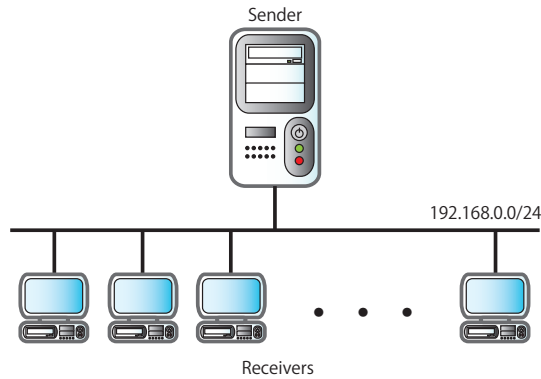


図 7.2: ネットワークトポロジ

7.3.1 計測結果

パケットペアを用いて取得した帯域幅の計測結果から、10Mbps を階級としたヒストグラムを図 7.3 に示す。図 7.3 より、100Mbps 以下に計測結果が集中しているのがわかる。そのため、計測結果のうち 100Mbps 以下を有効数字とする。

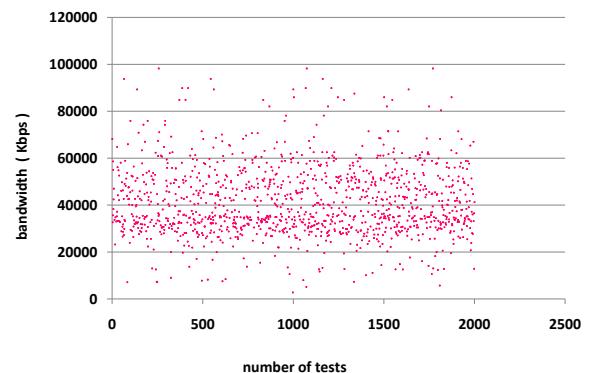
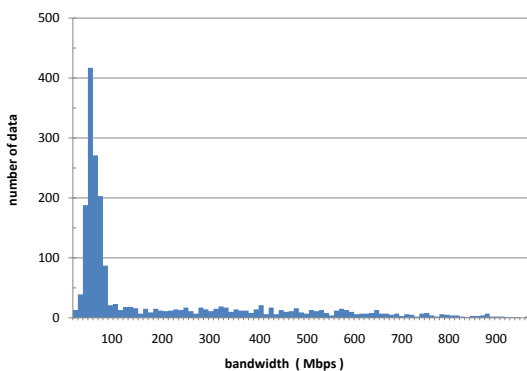


図 7.3: パケットペアによる帯域幅計測結果のヒストグラム

図 7.4: パケットペアによる帯域幅計測結果

図 7.4 に、100Mbps より大きい計測結果を除外した帯域幅の計測結果を示す。図 7.4 を観察すると、20Mbps から 60Mbps に計測結果が集中している。

7.3.2 考察

本実験において、送信ノードの NIC および Ethernet ケーブルは 100Base-T であり、ポトルネックになっていると考えられる。100Base-T の実効速度は通常 70Mbps 前後であるため、計測結果は有効である。従って、実ネットワーク上において、本システムを利用して帯域幅を計測できるといえる。

7.4 実験 1: ダミーネットを含めたパケットペア精度測定

仮想ネットワークによって構成された疑似ネットワークにおける、パケットペアを用いた帯域幅の計測精度を評価する。図 7.5 に示すネットワークにて評価実験を行った。

評価に用いたネットワークは、送信ノードと受信ノードをひとつのセグメント上に配置し、送信ノードが接続されている回線の帯域幅をダミーネットにて制御する。ダミーネットの帯域幅を 500Kbps、1Mbps、2Mbps、5Mbps、10Mbps と順に変化させ、各帯域幅に対し複数回計測を行った。

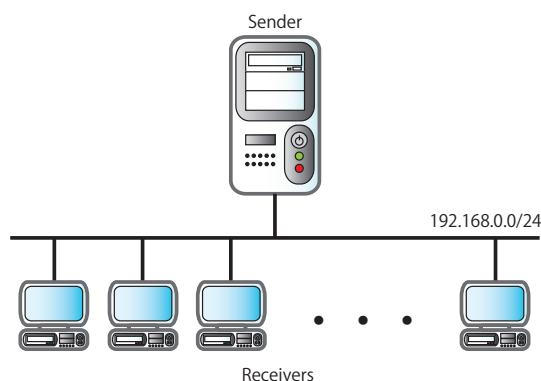


図 7.5: 実験 1 におけるネットワークトポロジ

7.4.1 帯域幅 500Kbps

表 7.4 に、ダミーネットで制御した帯域幅が 500Kbps の時、送信ノードが 40,000 パケットを送信した場合、受信ノードで取得したデータの一例を挙げる。

同一条件下で 40,000 パケットの送信を 1 回として複数回計測を行った結果、ペアパケットの計測回数は常に 12 回前後であった。総パケット数が 20 を超えるとパケット欠損率が 50% 以上に上昇し、最終的にパケット欠損率が約 75% を示した。

7.4. 実験 1: ダミーネットを含めたパケットペア精度測定

表 7.4: 帯域幅 500Kbps の場合のパケットロスの変化

ペアパケット計測回数 (回)	1	2	3	4	5	6	7	8	9	10	11
パケットロス数	0	0	0	0	0	0	0	0	0	100	29774
総パケット数	2	4	6	8	10	12	14	16	18	160	39998
パケットロス率 (%)	0	0	0	0	0	0	0	0	0	62.5	74.9387

本システムでは、シーケンス番号が偶数番とそれより 1 大きい奇数番をひとつのペアとして帯域幅を計測しているため、ペアとなる両方のパケットが受信ノードに到達しないと帯域幅を計測できない。そのため、実効帯域が 500Kbps の場合、パケット欠損数が非常に多いためパケットペアの計測数が少なく、帯域幅を計測することは困難である。

7.4.2 帯域幅 1Mbps

図 7.6 に示す様に、20,000 ペアのパケットのうち受信ノードが受信したペアパケットは約 100 ペアである。この約 100 回の計測回数のうち、10 回目前後でパケットロス率が急激に増加し、20 回目以降のパケットロス率はおよそ 55% で安定する。20,000 ペアのパケットを複数回送信した際、パケットペアから計測された帯域幅を図 7.7 に示す。取得した帯域幅のデータは、計測回数 15 回前後までは 500Kbps 以下を示し、その後はおよそ 1Mbps を示している。

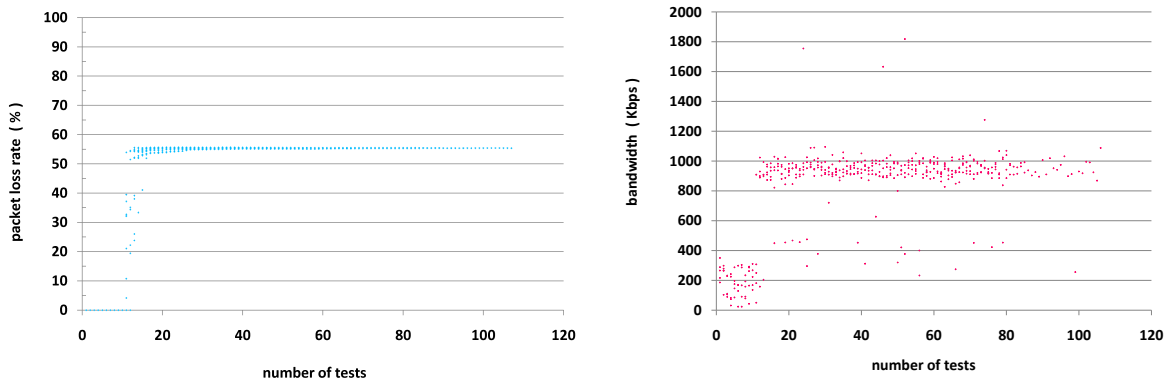


図 7.6: 帯域幅 1Mbps の場合のパケットロス率の変化 図 7.7: パケットペアによる帯域幅計測結果 (ボトルネックリンク 1Mbps)

図 7.6 と図 7.7 を観察すると、計測回数がおおよそ 15 回を超えると、パケットロス率と計測帯域幅の両方の値が安定する。パケットペアの計測開始時は、ダミーネットを通過する

パケットの量が増加傾向にあっても、想定した帯域幅の制御に対して処理すべきパケット量が不足している。しかし、一定の時間を過ぎると制御に対して処理すべきパケット量が過剰になり、ダミーネット内でパケット量がシェーピングされ、想定した範囲にパケットの間隔が拡大する。以上の要因から、ダミーネットの制御により、1Mbps の帯域幅が計測可能である。

7.4.3 帯域幅 2Mbps

ダミーネットの帯域幅を 2Mbps とし、20,000 ペアのパケットを複数回送信した。

送信した際に、受信ノードが受信したパケットペアは約 2,700 ペアである。各パケットペアから計測された帯域幅を図 7.8 に示す。パケットペアから計測された帯域幅は 2 極化し、500Kbps と 2Mbps 付近に集中している。

図 7.9 にパケットロス率の変化を示す。ペアパケットの計測開始と同時にパケットロス率は急激に上昇し、計測回数がおおよそ 100 回を超えるとパケットロス率が約 45% で安定する。

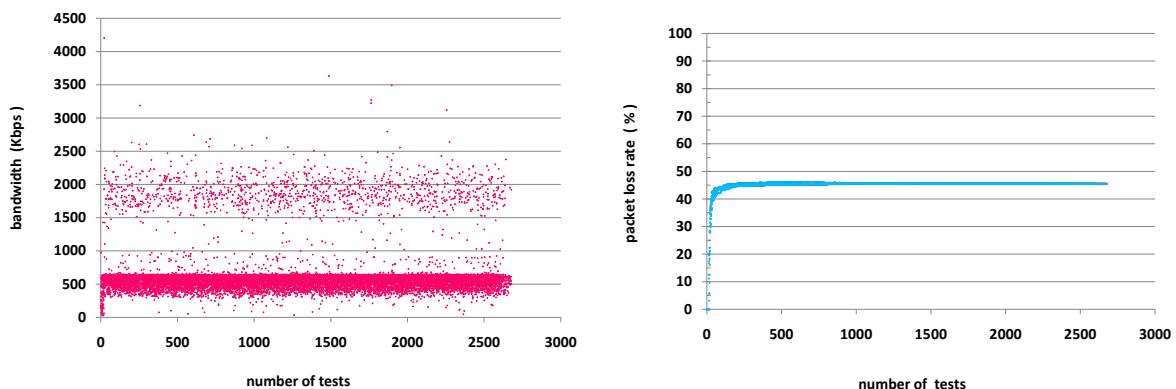


図 7.8: パケットペアによる帯域幅計測結果 (ボトルネックリンク 2Mbps) 図 7.9: 帯域幅 2Mbps の場合のパケットロス率の変化

図 7.9 と図 7.8 を観察すると、計測開始直後はパケットロス率も計測帯域幅も値が上昇するが、一定回数計測後は計測結果が安定する。帯域幅 1Mbps の場合と同様に、パケットロスの発生から、ダミーネットにおいて輻輳が発生しており、ボトルネックリンクとして機能しているといえる。しかし、帯域幅が 2Mbps の場合は帯域幅 1Mbps の場合と異なり、パケットペアによる計測結果が 1Mbps と 2Mbps の 2 種類の帯域に集中した。これは、ダミーネットによる帯域制御では、パケットペア理論で想定したパケットペアの間隔より、間隔の拡大する場合が多いことが要因といえる。パケット同士の間隔の拡大発生は、送信ノード、ダミーネット、受信ノードの全ての箇所において考えられる。そのため、本システムの使用には、パケット間隔の拡大発生箇所の特定において課題がある。

7.4. 実験 1: ダミーネットを含めたパケットペア精度測定

実験結果では、外れ値を除外した計測結果のうち上位 30%が 1.5Mbps 以上を示した。そのため、実験 3 以降では上位 30%を有効数字として帯域幅計測に用いる。

7.4.4 帯域幅 5Mbps

ダミーネットの帯域幅を 5Mbps とし、4,000 ペアのパケットを複数回送信した。受信ノードがペアとして受信したパケットペアの計測回数は約 2,000 回であった。それぞれのパケットペアから計測した帯域幅を図 7.10 に示す。

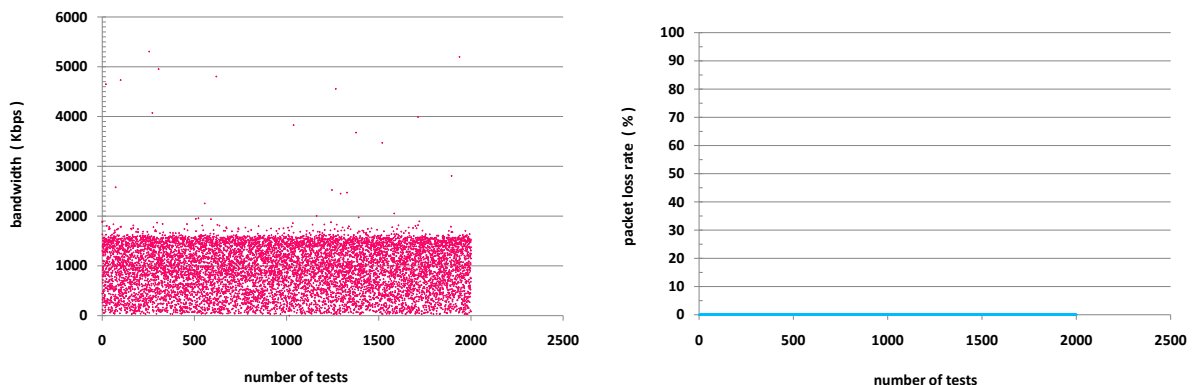


図 7.10: パケットペアによる帯域幅計測結果 図 7.11: 帯域幅 5Mbps の場合のパケットロスの変化 (ボトルネックリンク 5Mbps)

計測された帯域幅は 0Kbps から約 1.8Mbps までを示し、特定の帯域幅にパケットペアの計測結果が集中することはなかった。また、取得したデータからパケットロス率を求めたところ、常に 0%を示した (図 7.11)。

計測した帯域幅が特定の値に集中しないのは、パケットがダミーネットを通過する際、ペアとなるパケット同士の間隔が想定した値と異なる値に拡大していることが要因であると考えられる。パケットロスが発生しないことから、ダミーネット通過時に輻輳が発生していないことがわかる。ダミーネットによる帯域制御は実ネットワーク上でのボトルネックリンクにおけるパケットの処理とは異なる点に注目する。送信ノードから連続して送信されたパケットは、実ネットワーク上におけるボトルネックリンク通過時、図 7.12 の A に示すように、ひとつひとつのパケットの間隔が拡大される。しかし、図 7.12 の B に示す様に、ダミーネットによる帯域制御では、ダミーネットを通過するトラフィックがボトルネックリンクを想定してシェーピングされ、複数のパケットを一つの塊として塊同士の間隔を調整する。この処理の差異から、ダミーネットで帯域幅を 5Mbps にした場合、パケット同士が拡大される間隔がパケットペアの理論上想定される間隔と異なっていると考えられる。また、7.4.3 節で考察した様に、パケットが経由する機器においてパケット同士の間隔が拡大する可能性もある。

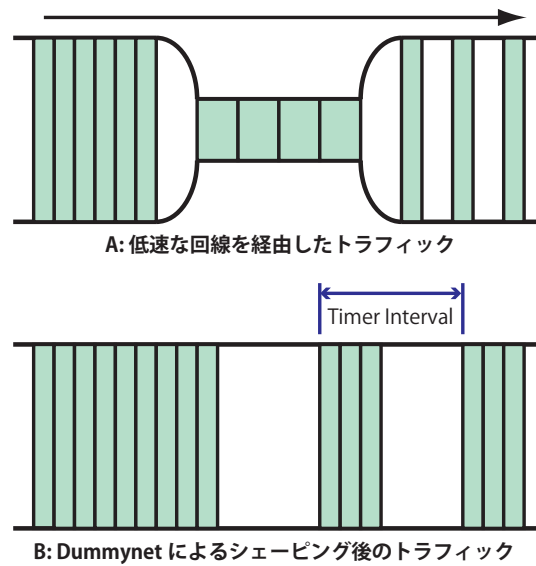


図 7.12: ダミーネット帯域制御の概念図

本システムにおいて、ダミーネットによりボトルネックリンクの帯域幅を 5Mbps に制御して計測した場合、計測値が特定の値に集中しないという課題がある。そのため、本実験環境において、帯域幅を 5Mbps に制御して実験することは困難である。

7.4.5 帯域幅 10Mbps

ダミーネットの帯域幅を 10Mbps とし、4,000 ペアの packets を複数回送信した。受信ノードが受信したペアパケットの計測回数は約 2,000 回であった。それぞれのパケットペアから計測した帯域幅を図 7.13 に示す。

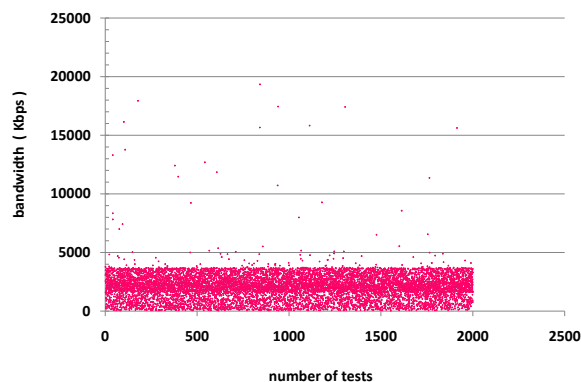


図 7.13: パケットペアによる帯域幅計測結果 (ボトルネックリンク 10Mbps)

7.5. 実験 2: 2 台のダミーネットを含めたネットワークにおけるパケットペア精度測定

計測された帯域幅は 0Kbps から約 4Mbps までを示し、特定の帯域幅にパケットペアの計測結果が集中することはなかった。また、取得したデータからパケットロス率を求めたところ、常に 0% を示した。

ダミーネットにより帯域幅を 5Mbps に設定した時と同様に、ダミーネット通過時に廃棄されるパケットがないため、様々な計測値が発生したと考えられる。図 7.12 B における timer interval の値は、ダミーネットで想定する帯域幅毎に設定され、10Mbps の場合の値は 5Mbps の場合の半分の値である。そのため、計測値が 5Mbps の時の上限値の 2 倍となったと考えられる。

5Mbps の場合と同様に、計測値が特定の値で安定しないため、パケットペア理論に基づく帯域幅算出に関して課題がある。そのため、本実験環境において、帯域幅を 10Mbps に制御して実験することは困難である。

7.5 実験 2: 2 台のダミーネットを含めたネットワークにおけるパケットペア精度測定

図 7.14 に示すネットワークにて、パケットペアを用いた帯域幅の計測精度を評価する。実験 2 に用いるネットワークは実験 1 とは異なり、A と B の 2 つのダミーネットにより、ネットワークのうち 2 か所で帯域幅が制御される。実験 1 の結果を踏まえ、最も想定した値に近い値として、1Mbps と 2Mbps の制御を利用する。制御する 2 つの帯域幅の組み合わせを変えて 4 種類の実験を行った。帯域幅の組み合わせとボトルネックリンク帯域幅の理論値を表 7.5 に示す。

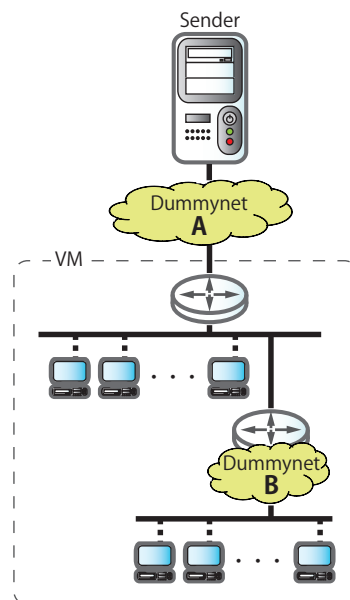


図 7.14: 実験 2 におけるネットワークトポロジ

表 7.5: 帯域幅の組み合わせ

	A	B	ボトルネックリンク帯域幅理論値
case1	1Mbps	500Kbps	500Kbps
case2	1Mbps	2Mbps	1Mbps
case3	2Mbps	500Kbps	500Kbps
case4	2Mbps	1Mbps	1Mbps

7.5.1 計測結果

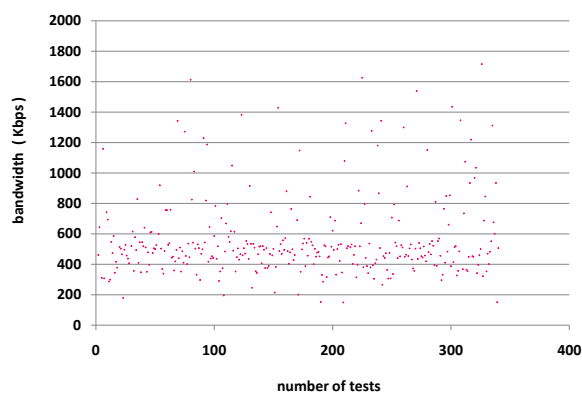


図 7.15: case1 の計測帯域幅

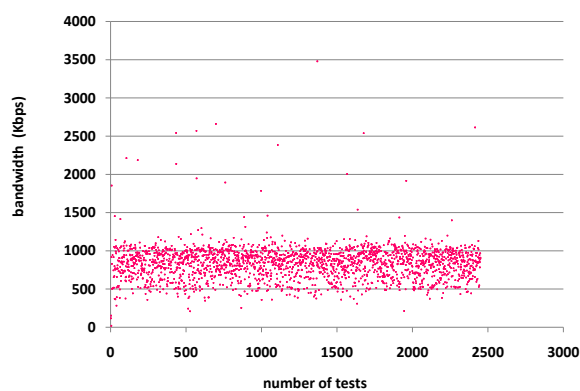


図 7.16: case2 の計測帯域幅

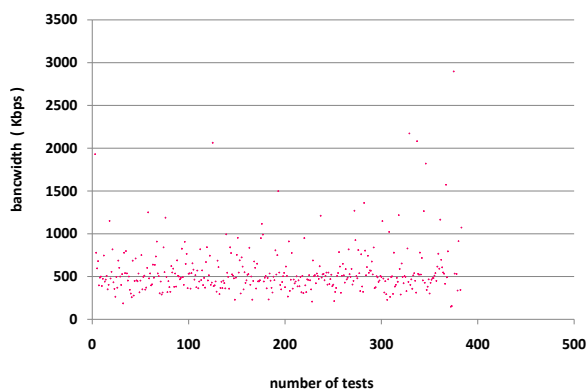


図 7.17: case3 の計測帯域幅



図 7.18: case4 の計測帯域幅

7.5. 実験 2: 2 台のダミーネットを含めたネットワークにおけるパケットペア精度測定

図 7.15, 図 7.16, 図 7.17, 図 7.18 に各 case において計測した帯域幅のうち, 外れ値を除外した結果を示す. ただし, case4 については上位 10% を外れ値とした.

また, 図 7.19, 図 7.20, 図 7.21, 図 7.22 に各 case におけるパケットロス率の変化を示す.

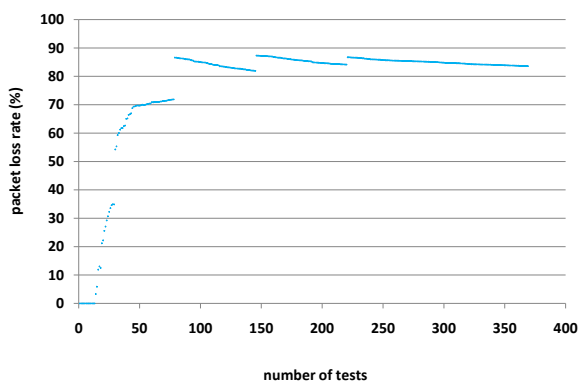


図 7.19: case1 のパケットロス率の変化

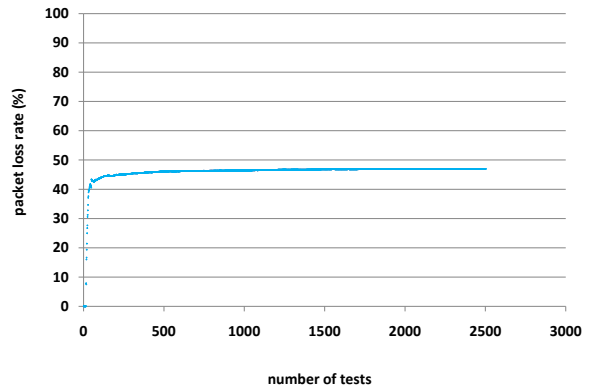


図 7.20: case2 のパケットロス率の変化

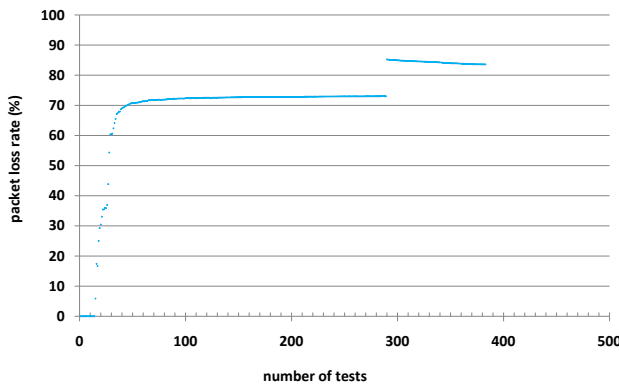


図 7.21: case3 のパケットロス率の変化

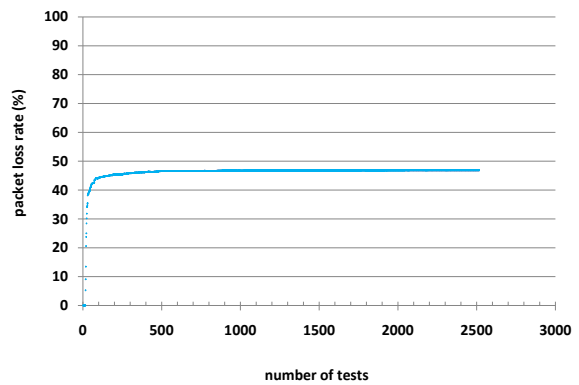


図 7.22: case4 のパケットロス率の変化

図 7.15, 図 7.16, 図 7.17, 図 7.18 より, 計測結果によるボトルネックリンクの帯域幅は, case1 では 400Kbps から 600Kbps, case2 では 500Kbps から 1Mbps, case3 では 300Kbps から 500Kbps, case4 では 500Kbps から 1Mbps を示す.

図 7.19, 図 7.21 を観察すると, ダミーネット B で制御された帯域幅が 500Kbps の場合, パケットロス率の変化は一定の値に収束していくものの, パケットロスが周期的に発生する. 図 7.20, 図 7.22 を観察すると, ダミーネットにより制御される帯域幅の組み合わせが 1Mbps と 2Mbps の場合は, パケットロス率が 45% 強に収束する.

7.5.2 考察

ダミーネットを2つ用いた本実験では、どの case においても、計測されたボトルネックリンクの帯域幅が理論値に近い値として示される。case1 と case3 に関して、実験1では、帯域幅を 500Kbps に制御すると計測不能であったが、実験2では、図 7.14 のダミーネットAを介したパケットがダミーネットBによって 500Kbps に制御されている。そのため、ダミーネットA通過時にパケット同士の間隔が拡大しており、ダミーネットB通過時には帯域制御が 500Kbps であっても制御処理が可能であり、帯域幅計測及びパケットロス率の計測が可能であったと考えられる。case2 と case4 に関して、計測された帯域幅に大きな差異はない。これは実験1において、ダミーネットによる帯域幅の制御が 2Mbps の場合に計測された帯域幅が 1Mbps と 2Mbps に二極化し、特に 1Mbps を示す場合が多かったことが要因と考えられる。

実験1の結果に基づくと、ダミーネットによる帯域制御が 1Mbps の場合のパケットロス率は約 55%、2Mbps の場合のパケットロス率は 45%である。従って、1Mbps と 2Mbps の2つのボトルネックリンクがある場合、理論上のパケットロス率は約 75%である。しかし計測されたパケットロス率は、case2 と case4 の双方の場合で約 45%を示した。これは、時間単位でのパケット処理数を調整することで、ダミーネットの帯域が制御されていることが要因と考えられる。帯域制御が 1Mbps の場合も 2Mbps の場合も、一つのダミーネットをパケットが通過する場合のパケットロス率は同じであるため、同程度のパケット数の廃棄により帯域制御がされると考えられる。

7.6 実験3: ダウンストリームをボトルネックリンクとするネットワークにおける品質把握

図 7.23 に示す VMware を用いた仮想ネットワークにおいて、ネットワーク状況把握の性能評価を行う。AからFの各ルータはダミーネットの機能も備えており、実験に利用するネットワークは、ダミーネットによって図 7.23 内の赤線のように帯域幅が制御される。帯域幅の制御は実験1と実験2の結果から、本実験環境下において有効と考える値の組み合わせを利用した。ダミーネットによって制御された各受信ノードのボトルネックリンク帯域幅の理論値を表 7.6 に示す。本実験により、計測値と理論値の差異を調べる。

本環境下において、受信ノードの返答確率を 100%、80%、50%、30%と順に変化させて実験する。また、返答確率を 10%に固定し、返答回数を 1回、3回、5回、8回、10回変化させ、サンプリングデータの遷移を調べる。

7.6.1 計測結果

各返答確率において、送信ノードが RTCP RR を 10回受信した場合の計測結果を図 7.24、図 7.25、図 7.26、図 7.27 に示す。また、返答確率が 100%の場合、1回の RTCP に

7.6. 実験 3: ダウンストリームをボトルネックリンクとするネットワークにおける品質把握

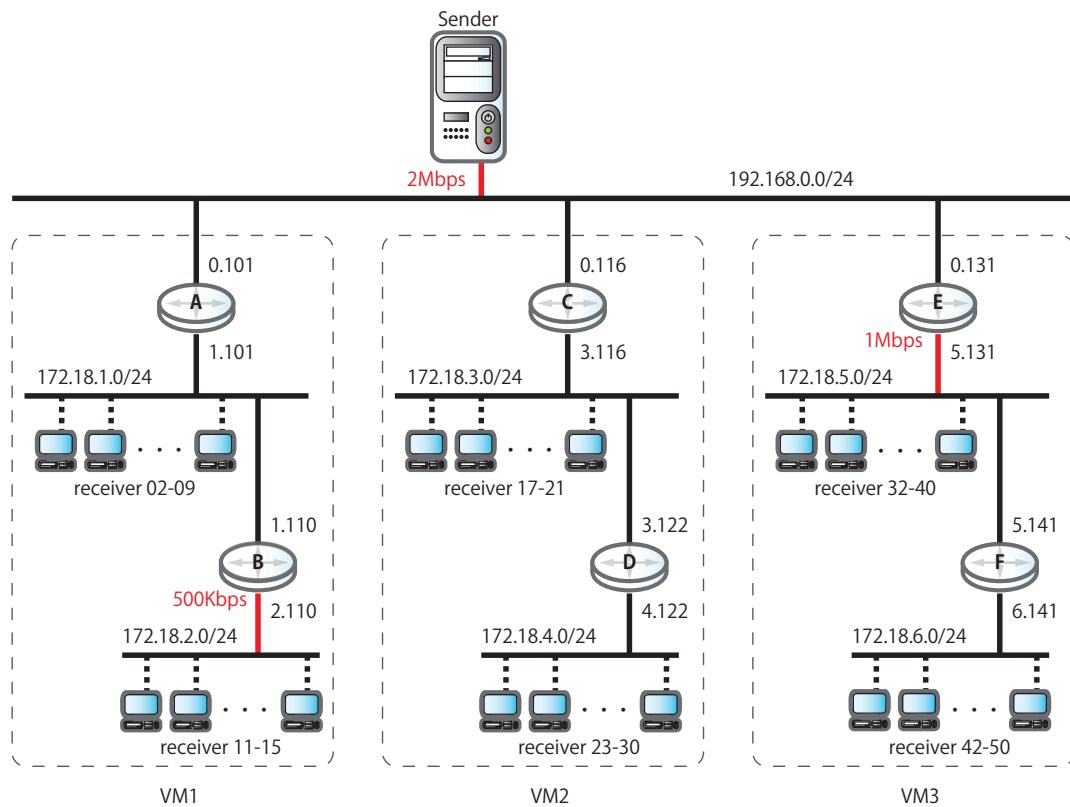


図 7.23: 実験 3 におけるネットワーク物理トポロジ

表 7.6: 実験 3 におけるボトルネックリンク帯域幅の理論値

受信ノード番号	帯域幅
receiver02 ~ receiver09	2Mbps
receiver11 ~ receiver15	500Kbps
receiver17 ~ receiver21	2Mbps
receiver23 ~ receiver30	2Mbps
receiver32 ~ receiver40	1Mbps
receiver42 ~ receiver50	1Mbps

より取得したデータの結果を図 7.28 に示す。そして、返答確率が 10% の場合、RTCP によるデータ取得の回数を 1 回、3 回、5 回、8 回、10 回と変化させた結果を図 7.29、図 7.30、図 7.31、図 7.32、図 7.33 に示す。

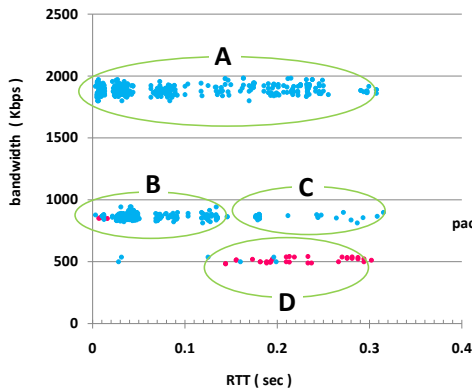


図 7.24: 返答確率 100%

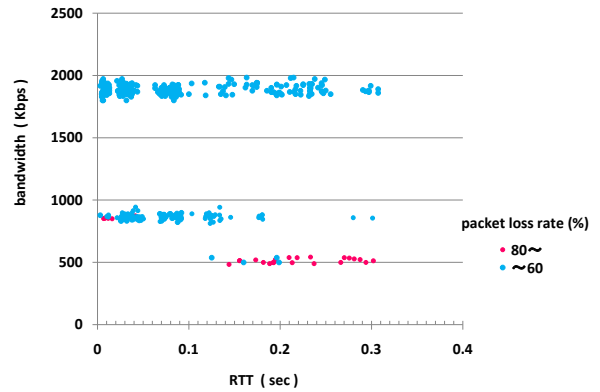


図 7.25: 返答確率 80%

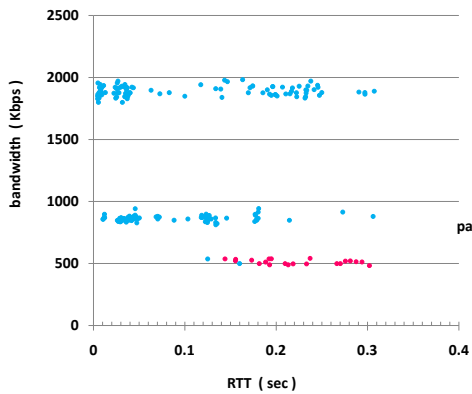


図 7.26: 返答確率 50%

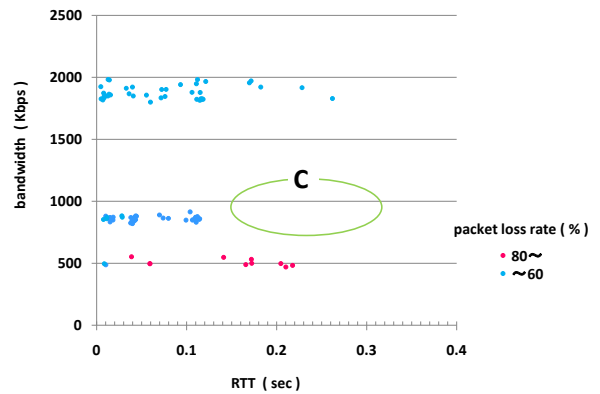


図 7.27: 返答確率 30%

7.6.2 考察

図 7.24, 図 7.25, 図 7.26, 図 7.27 より, どの返答確率の場合も, 受信ノードが存在するネットワーク上にボトルネックリンクとして約 500Kbps, 1Mbps, 2Mbps の 3 種類の帯域が存在することが分かる. 各図を観察すると, ボトルネックリンクが 2Mbps の帯域に存在するノードが一番多く, 500Kbps と 1Mbps の帯域に存在するノードは 1Mbps の帯域の方が多い. 図 7.24 に注目すると, 計測された帯域幅によって RTT が異なり, A は D の約 2 倍拡大し, B と C を合わせると A と同等な範囲に, C と D は同等な範囲に存在している. しかし図 7.27 では, 図 7.24 の C に該当する範囲に受信ノードが存在しない. 返答確率が 100% の場合の結果が, マルチキャスト通信が利用されている範囲のネットワーク品質を完全に反映しているとする, 返答確率が低い程, 品質データにばらつきが生じるが, ネットワークの概要を把握することは可能であるといえる.

7.6. 実験 3: ダウンストリームをボトルネックリンクとするネットワークにおける品質把握

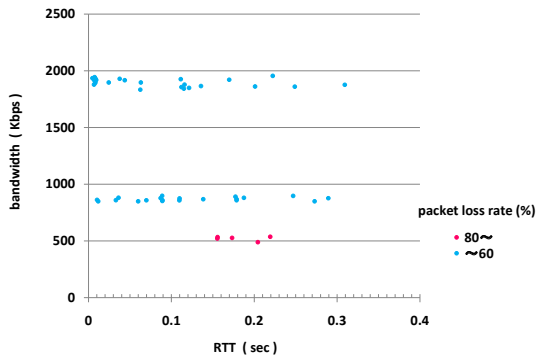


図 7.28: 返答回数 1 回 (返答確率 100%)

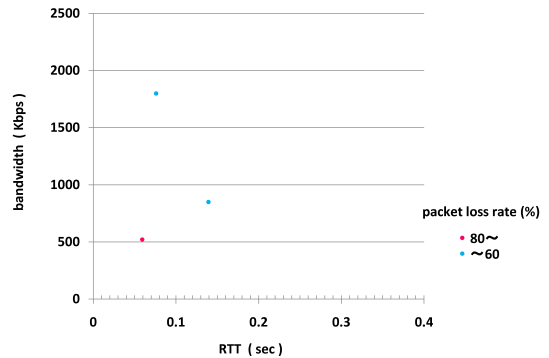


図 7.29: 返答回数 1 回 (返答確率 10%)

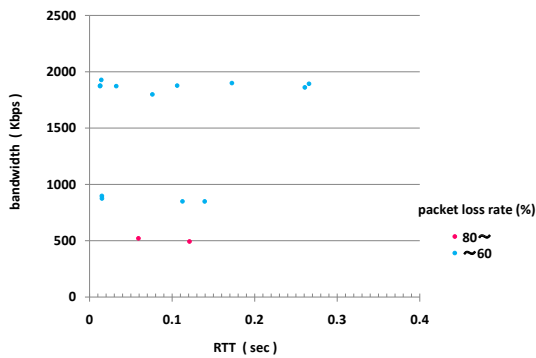


図 7.30: 返答回数 3 回 (返答確率 10%)

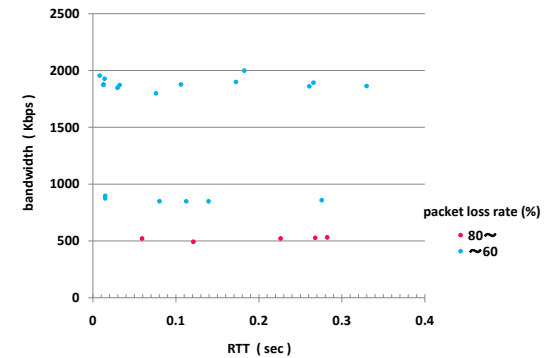


図 7.31: 返答回数 5 回 (返答確率 10%)

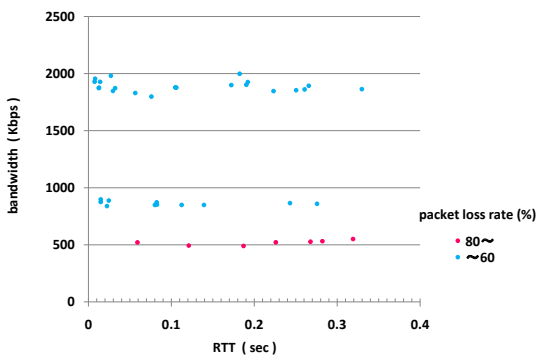


図 7.32: 返答回数 8 回 (返答確率 10%)

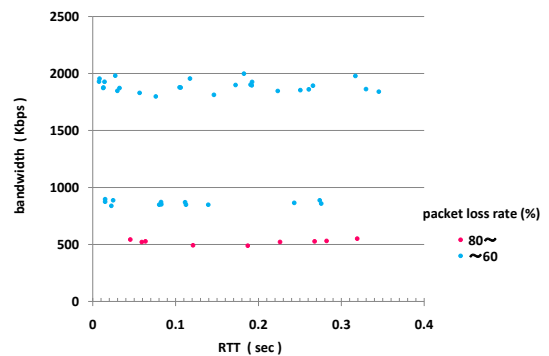


図 7.33: 返答回数 10 回 (返答確率 10%)

本実験では、帯域幅の計測値が約 1Mbps を示すのは、1 つのダミーネットにより 1Mbps に制御されている場合と、2 つのダミーネットを用いて 1Mbps と 2Mbps に帯域を制御している場合が考えられる。また、帯域幅の計測値が 500Kbps を示すのは、2 つのダミーネットにより制御され、受信ノードに近いダミーネットが 500Kbps の場合である。以上のことから、実験に利用したネットワークには複数のダミーネットが存在していることが考えられる。本実験環境では、ダミーネットによる帯域制御により、パケットロスが特定の割合で発生する。そのため、ネットワーク品質把握の指標としてパケットロス率の値を利用することができない。

実験 3 のうち返答確率 100% において取得したパケットロス率、帯域幅、RTT の値に対して、最低値、平均値、分散を求めた結果を表に示す。

表 7.7: パケットロス率、帯域幅、RTT の最低値、平均、分散 (RTCP RR 10 回)

	パケットロス率 (%)	帯域幅 (Mbps)	RTT(sec)
最低値	94.77	0.482	0.00303
平均値	49.97	1.418	0.08843
分散	127.04	0.2579	0.006375

表 7.8: パケットロス率、帯域幅、RTT の最低値、平均、分散 (RTCP RR 5 回)

	パケットロス率 (%)	帯域幅 (Mbps)	RTT(sec)
最低値	92.27	0.489	0.00436
平均値	50.63	1.3	0.102967
分散	138.31	0.3079	0.00903

表 7.9: パケットロス率、帯域幅、RTT の最低値、平均、分散 (RTCP RR 3 回)

	パケットロス率 (%)	帯域幅 (Mbps)	RTT(sec)
最低値	86.32	0.496	0.00462
平均値	51.24	1.389	0.0909
分散	129.49	0.3018	0.00743

表 7.10: パケットロス率、帯域幅、RTT の最低値、平均、分散 (RTCP RR 1 回)

	パケットロス率 (%)	帯域幅 (Mbps)	RTT(sec)
最低値	84.62	0.498	0.00462
平均値	50.26	1.318	0.138975
分散	121.84.	0.3334	0.008637

7.7. 実験 4: アップストリームをボトルネックリンクとするネットワークにおける品質把握

RTCP RR の回数が 10 回の場合を表 7.7, 5 回の場合を表 7.8 に, 3 回の場合を表 7.9 に, 1 回の場合を表 7.10 に示す.

ネットワークの構成要素を固定し, 受信ノードの返答確率 100%として実験を行った場合でも, 計測毎に取得する品質データは変化し, 各品質データの最低値, 平均値, 分散に揺らぎが生じることがわかる. この揺らぎは, 時間の経過に従ったネットワークの状態遷移を示す. 返答確率 10%で計測を行った場合, 理論上は 10 回の計測で全受信ノードからフィードバックを取得することができる. 図 7.29 と図 7.30, 図 7.31, 図 7.32, 図 7.33 を比較すると, 返答確率が 10%の場合でも, RTCP によるデータの取得回数を増加させる程, 返答確率が 100%の場合の結果と近似することがわかる. 従って, 返答確率を低く設定してフィードバックを取得しても, 計測回数を増加させることで計測精度を高めることができ, ネットワーク品質把握が有効であるといえる.

7.7 実験 4: アップストリームをボトルネックリンクとするネットワークにおける品質把握

図 7.34 に示す VMware を用いた仮想ネットワークにおいて, ネットワーク状況把握の性能評価を行う. 利用するネットワークについて, ルータ B とルータ E はダミーネットの機能も備えており, ダミーネットによって図 7.34 内の赤線の様に帯域幅が制御されている. ダミーネットによって制御された各受信ノードのボトルネックリンク帯域幅の理論値を表 7.12 に示す. 本実験では, 計測値と理論値の差異を調べる.

表 7.11: 実験 4 におけるボトルネックリンク帯域幅の理論値

受信ノード番号	帯域幅
receiver02 ~ receiver09	1Mbps
receiver11 ~ receiver15	500Kbps
receiver17 ~ receiver21	1Mbps
receiver23 ~ receiver30	1Mbps
receiver32 ~ receiver40	1Mbps
receiver42 ~ receiver50	1Mbps

表 7.12:

本環境下において, 受信ノードの返答確率を 100%, 80%, 50%, 30%と順に変化させて実験する. また, 計測回数を変化させ, サンプルングデータの遷移を調べる.

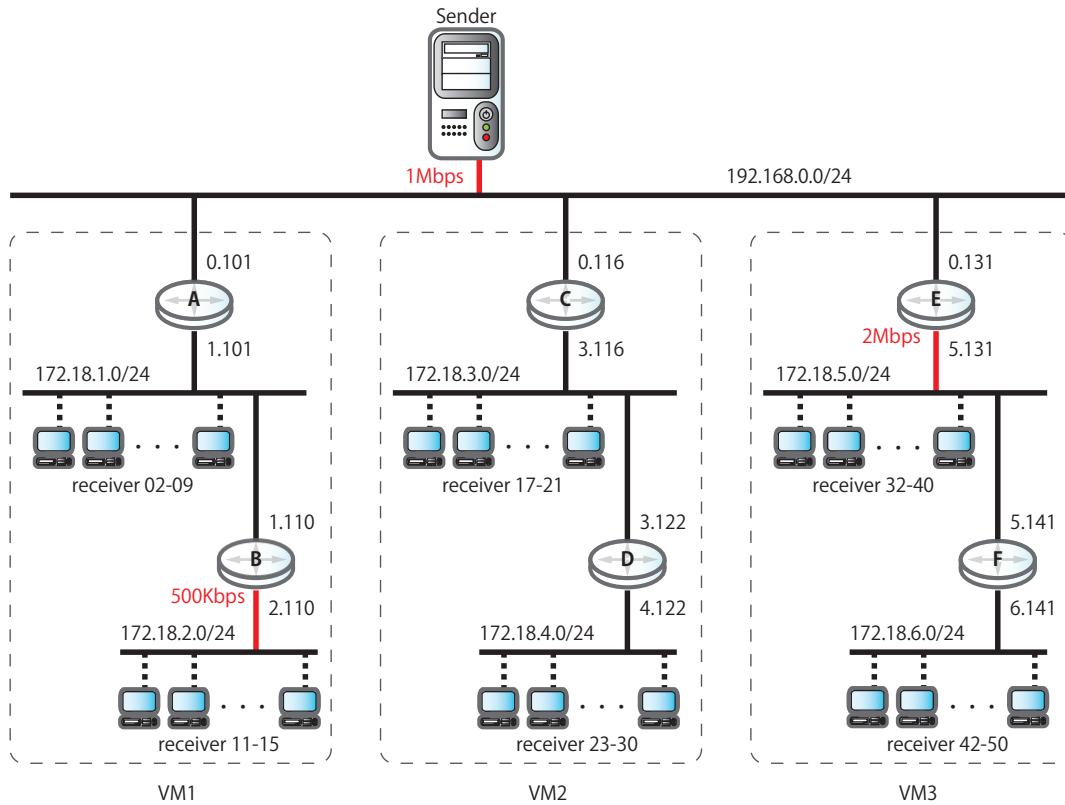


図 7.34: 実験 4 におけるネットワーク物理トポロジ

7.7.1 計測結果

各返答確率において、送信ノードが RTCP RR を 10 回受信した場合の計測結果を図 7.35, 図 7.36, 図 7.37, 図 7.38 に示す。また、返答確率が 100% の場合、1 回の RTCP SR 送信により取得したデータの結果を図 7.39 に示す。そして、返答確率が 10% の場合、RTCP によるデータ取得の回数を 1 回、3 回、5 回、8 回、10 回と変化させた結果を図 7.40, 図 7.41, 図 7.42, 図 7.43, 図 7.44 に示す。

7.7.2 考察

図 7.35, 図 7.36, 図 7.37, 図 7.38 を観察すると、どの返答確率の場合も、帯域幅 1Mbps 付近に多くの受信ノードが集中し、残りのノードは帯域幅 500Kbps のネットワークに存在する。ボトルネックリンク帯域幅 1Mbps のネットワークに属する受信ノードは送信ノードからネットワークの距離として近い位置に存在し、500Kbps のネットワークに属する受信ノードは 1Mbps 帯域幅の場合より遠い位置にも存在する。実験 1・2 の結果より、本実験において計測帯域幅が 500Kbps を示すのは、ネットワーク上にダミーネットがふたつ存在し、受信ノードに近いダミーネットが 500Kbps に制御されている場合である。

7.7. 実験 4: アップストリームをボトルネックリンクとするネットワークにおける品質把握

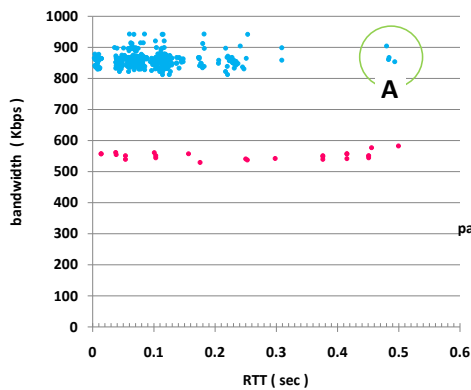


図 7.35: 返答確率 100%

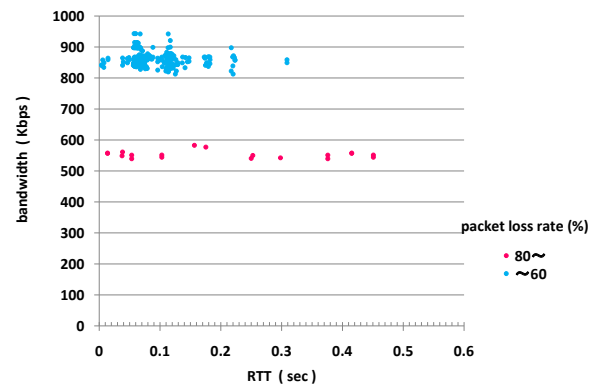


図 7.36: 返答確率 80%

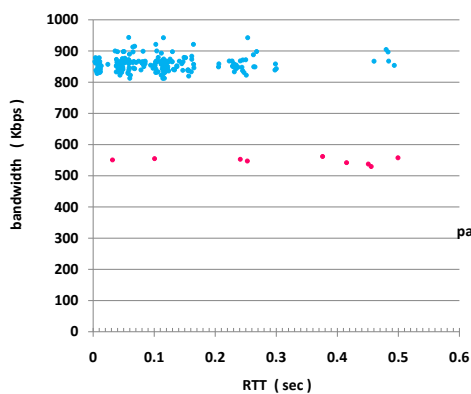


図 7.37: 返答確率 50%

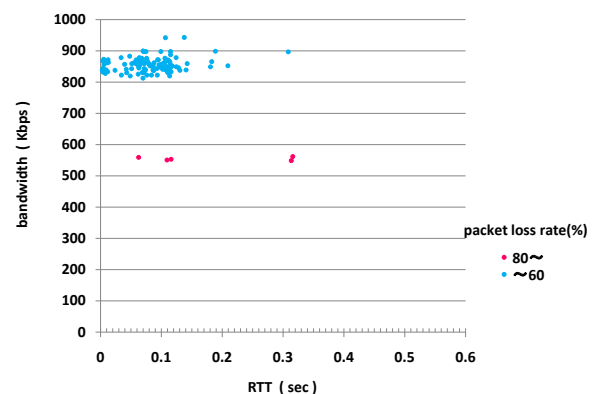


図 7.38: 返答確率 30%

方, 1Mbps を示すのは, 1Mbps に制御されたダミーネットがひとつだけ存在するネットワークか, ダミーネットが 2 つ存在し, 1Mbps と 2Mbps の帯域制御がされている場合である。

ここで図 7.35 内の A の範囲に注目すると, 図 7.37 では同じ範囲に計測値が存在するが, 図 7.36 と図 7.38 の場合は存在しない。このように計測値の発生頻度が低い場合, 非常に小さなネットワークが存在すると考えられる。しかし, 必ずしもネットワーク品質として送信ノードがデータ取得するわけではないため, 推測は困難である。また, 実験 3 と同様に, ダミーネットによる帯域制御では特定の割合でパケットロスが発生するため, ネットワーク品質の指標としてパケットロス率を利用することは困難である。

実験 3 と同様に返答確率 10% で計測を行った場合, 図 7.39 と図 7.40, 図 7.41, 図 7.42, 図 7.43, 図 7.44 を比較すると, RTCP によるデータ取得回数を増加させることで計測精度を高め, 100% の返答確率の場合と類似した結果を取得することができる。さらに, 計

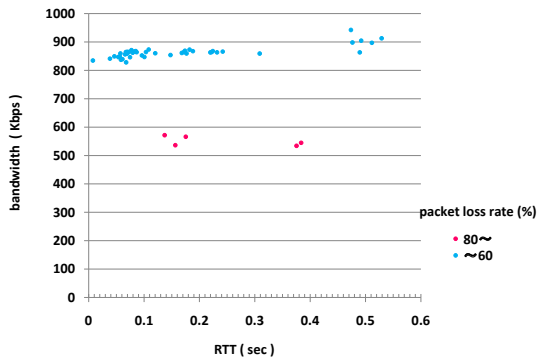


図 7.39: 返答回数 1 回 (返答確率 100%)

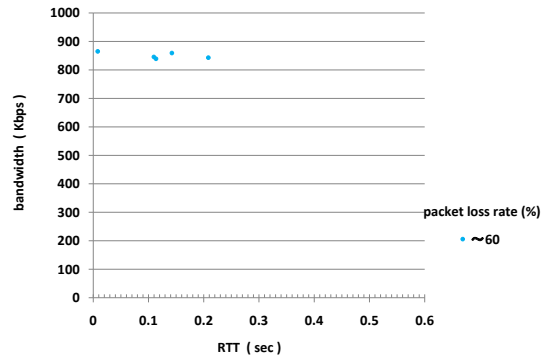


図 7.40: 返答回数 1 回 (返答確率 10%)

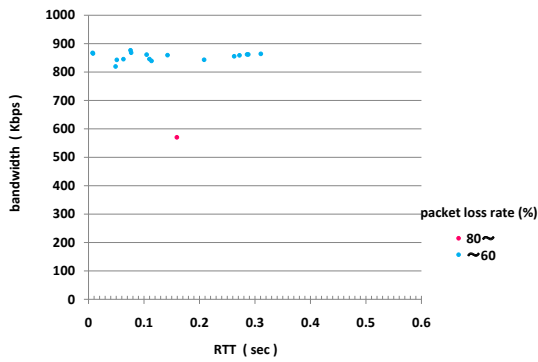


図 7.41: 返答回数 3 回 (返答確率 10%)

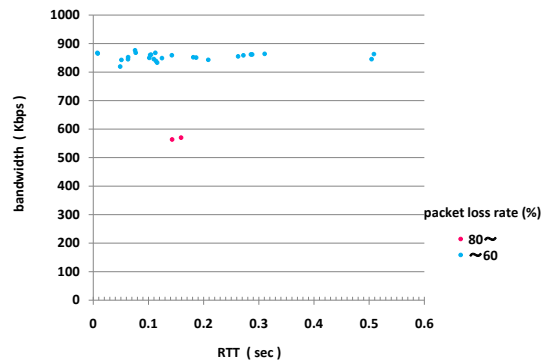


図 7.42: 返答回数 5 回 (返答確率 10%)

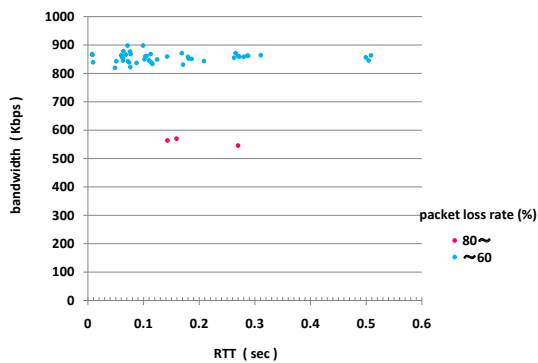


図 7.43: 返答回数 8 回 (返答確率 10%)

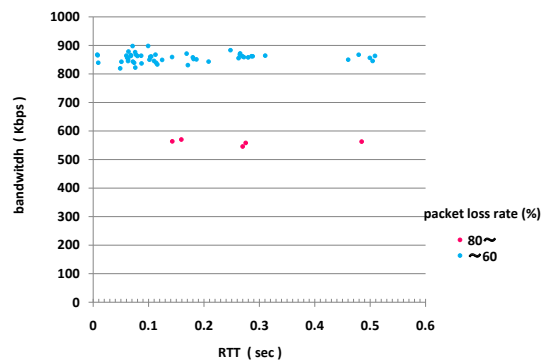


図 7.44: 返答回数 10 回 (返答確率 10%)

7.8. 実験 5: 遅延を有するネットワークにおける品質把握

測を複数回行うことで、時間の経過に従ったネットワークの状態遷移を反映することができ、リアルタイム性を伴った品質把握が可能であるといえる。

7.8 実験 5: 遅延を有するネットワークにおける品質把握

図 7.45 に示す VMware を用いた仮想ネットワークにおいて、ネットワーク状況把握の性能評価を行う。利用するネットワークは、図 7.45 内の赤線部をボトルネックリンクと想定し、送信ノードから受信ノードの方向にパケットが通過した場合、ダミーネットにより遅延を誘発する。B のダミーネットでは 1,000msec の遅延、E のダミーネットでは 5,000msec の遅延が発生するように制御した。また、全ダミーネットでは 1Mbps の帯域制御を行った。

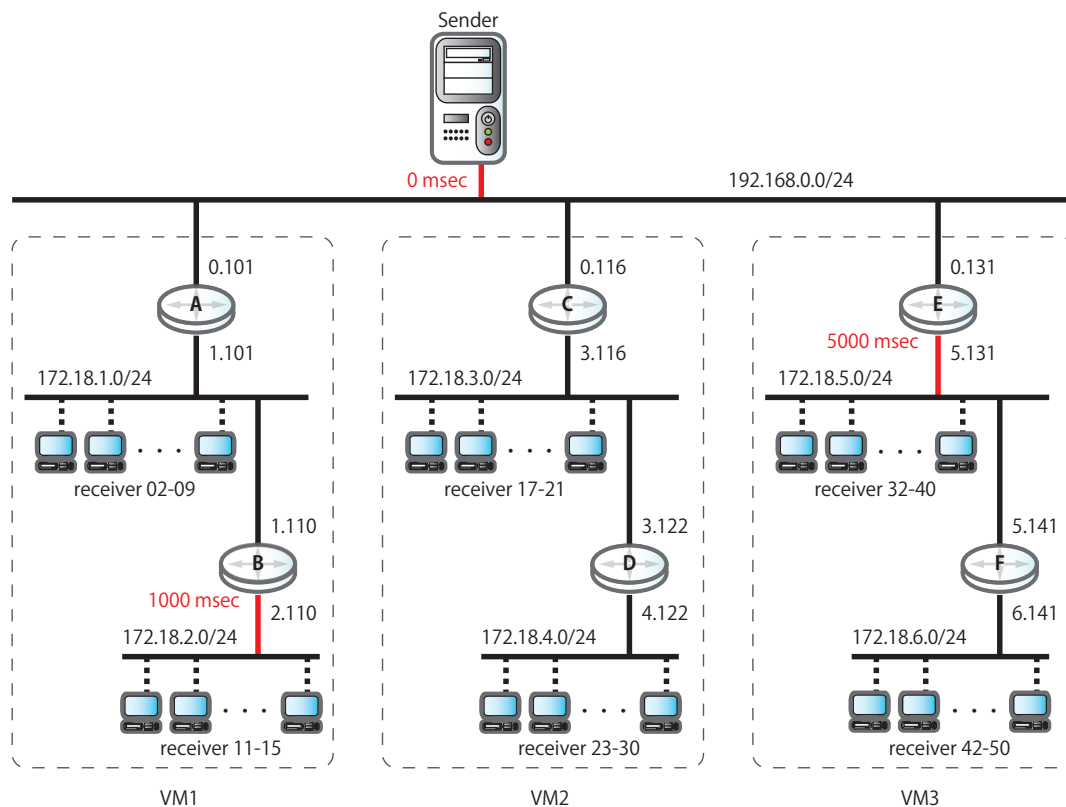


図 7.45: 実験 5 におけるネットワーク物理トポロジ

本環境下において、受信ノードの返答確率を 100%、80%、50%、30%と順に変化させて実験する。また、計測回数を変化させ、サンプリングデータの遷移を調べる。

7.8.1 計測結果

各返答確率において、送信ノードが RTCP RR を 10 回受信した場合の計測結果を図 7.46、図 7.47、図 7.48、図 7.49 に示す。

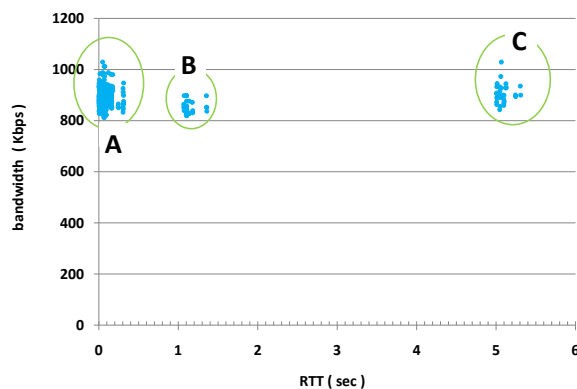


図 7.46: 返答確率 100%

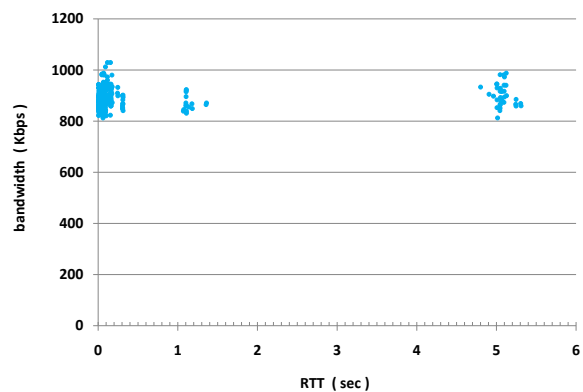


図 7.47: 返答確率 80%

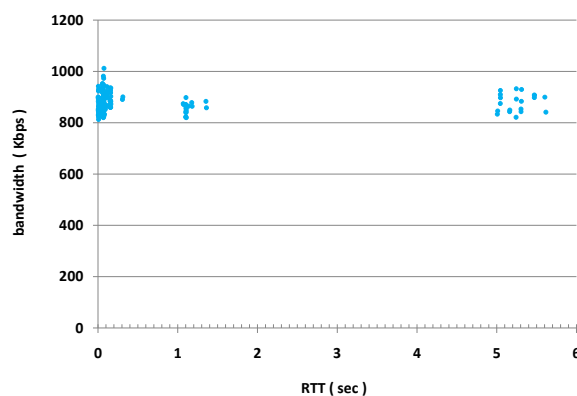


図 7.48: 返答確率 50%

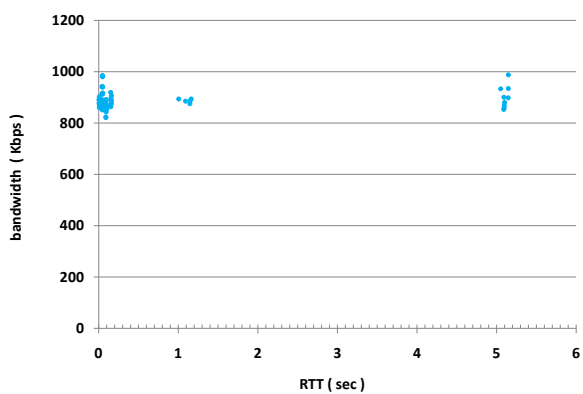


図 7.49: 返答確率 30%

7.8.2 考察

図 7.46, 図 7.47, 図 7.48, 図 7.49 を観察すると, どの返答確率の場合も 800Kbps から 1Mbps の帯域幅に計測値が存在し, 図 7.46 の A, B, C のように 0.5sec 以下, 1sec 前後, 5sec 前後の 3 種類の範囲で計測値が密集している. 本実験におけるダミーネットの制御では, 送信ノードから受信ノードへの方向のリンクに対し, 遅延なし, 1000msec の遅延, 5000msec の遅延の 3 種類の設定をした. 以上のことから, 本システムにおいて, 計測結果から取得した RTT の値はネットワーク品質把握のためのデータとして有効であるといえる.

表 7.13 に, 実験 5 において返答確率 100% の場合, ネットワーク品質として取得した RTT の最低値, 平均値, 分散の推移を示す. 表 7.13 から, 同一環境下において RTT の測定を行っても, 測定毎にデータ値が異なることがわかる. 実験 3 や実験 4 と同様に計測回

7.9. 実験のまとめと考察

表 7.13: 実験 5 において取得した RTT の最低値，平均値，分散

計測回数	1	2	3	4	5	6	7	8	9	10
最低値	0.0054	0.003	0.0065	0.0034	0.0047	0.0062	0.0038	0.0063	0.0028	0.0037
平均値	0.443	0.5190	0.3389	0.4577	0.4264	0.3092	0.3441	0.3389	0.8075	0.449
分散	1.6411	1.7589	1.2031	1.5793	1.3780	1.5304	1.028	1.022	1.500	1.466

数を重ねることで，マルチキャスト通信が対象とするネットワーク内に存在する受信ノードの RTT の遷移を，時間の経過に従って把握することができる．そのため，本システムを用いることで，リアルタイム性を伴った品質把握が可能であるといえる．

7.9 実験のまとめと考察

本システムでは，サンプリング手法を用いて帯域幅，パケットロス率，RTT を取得し，マルチキャスト受信者の受信品質の傾向を把握した．評価実験では，様々な実験ネットワークを構築し，返答確率を変化させて各受信ノードからのフィードバックを取得した．

返答確率が 10% といった低い場合でも，サンプリングの計測回数を増やすことで，全受信ノードが返答する場合と同等の精度でマルチキャスト通信が対象とする通信範囲における通信品質の傾向が把握可能であることを示した．

本システムでは，返答確率や計測回数を柔軟に変化させることが可能である．そのため，一度に全受信ノードからフィードバックを取得する場合と比較して，ネットワーク及び送信ノードに与える負荷を軽減し，フィードバックインプロージョンを抑制できる．

一回のフィードバックからは，その瞬間のネットワークの品質状況しか把握できない．本システムを用いることで，フィードバックを断続的に取得し，計測した瞬間のデータのみでなく，時間の経過とともに遷移するネットワークの状況を反映したデータ受信品質を把握することが可能となった．

第8章 結論

本章では，本研究のまとめと今後の課題を述べる．

8.1 まとめ

本研究では，IP マルチキャスト通信におけるネットワークの品質を把握するため，リアルタイム通信プロトコルである RTP/RTCP とパケットペア理論を利用し，RTT，パケットロス率，送信ノードから受信ノードまでの経路中にあるボトルネックリンク帯域幅を推測するシステムを設計，実装した．マルチキャストグループの規模に応じてランダムに選出した受信者からフィードバックを受け取る AMES と呼ばれるサンプリング手法を利用し，多数の受信者により誘発されるフィードバックインプロージョンの問題を回避しながら，通信品質に関する情報の取得が可能になった．本システムは，マルチキャスト受信者数の規模に応じてサンプリングする受信者数を調整するため，大規模なマルチキャストネットワークにおいても利用可能である．

本研究では，システムの正当性を評価するため，様々な実験ネットワークを構築し，実験パターンを変化させて評価実験を行った．評価実験により，以下の四点を示した．

- 規模性
サンプリング手法を用いた場合でも，多数のマルチキャスト受信者におけるデータ受信品質の傾向を把握することが可能である．
- スケーラビリティ
サンプリング手法を用いることで，フィードバック数の分散が可能となり，ネットワークや送信者の負荷を軽減する．
- 柔軟性
本システムは，返答確率や計測回数を柔軟に変化させることが可能であるため，ネットワークの負荷状況やオペレーションの要求などに応じて，推測精度をコントロールすることが可能となる．
- 連続性
断続的なサンプリングを行うことで，時間の経過にともなう受信者のデータ受信品質の遷移が把握可能となる．

8.2 今後の課題

本研究では、ダミーネットにより実効帯域を制御し、VMware を利用して多数の受信者を想定して評価実験を行った。ダミーネットによるボトルネックリンクの制御は、実ネットワークにおける輻輳発生メカニズムとは異なることに加え、本システムがユーザランドにおける実装であること、更に VMware 上で多数のゲストを稼働させたという要因から、データ処理に限界が生じ、想定したデータの値と実測値の差異が大きい。パケットペア理論を用いた際、ダミーネットと VMware を使用すると、帯域制御に対して制限がかかる。そのため、本システムを実ネットワーク上で稼働させ、本研究と同様の実験を行う必要がある。

また、VMware を用いることで多数の受信者を想定したが、本実験の計算機環境では VM ゲストの設定が最大 50 台に留まった。数百、数千、数万人の受信者を想定した規模性に関して、本システムの対応を検証する必要がある。

最後に、推測精度を向上するための研究を課題として挙げる。本研究では、サンプリング手法を用いることで、フィードバックを分散しつつ、多数のマルチキャスト受信者におけるデータ受信品質の傾向を把握することができた。評価実験では返答確率と計測回数を変化させることで、傾向の把握が可能であることを示したが、把握するための効果的な確率と回数の組み合わせを導き出せるのではないかと考える。傾向を把握するための精度を向上させる具体案として、新しいパラメータの追加を挙げる。例えば、本システムで利用した返答確率 p に加え、パケットロス率に制限を加えることで、マルチキャストの全受信者から、特定の受信者に焦点を当てることが可能となる。焦点を当てた特定の受信者をグルーピングできれば、FEC (Forward Error Correction) による制御を行うといった対策が可能となり、レイヤードマルチキャストの際の指標となると考える。また、グルーピングされた特定の受信者から IP アドレスを取得することで、グループの属性を把握し、AS に対するピアリングのストラテジーを考慮することが可能になると考える。以上のことから、新しいパラメータに対する検討を今後の課題とする。

謝辞

本研究を進めるにあたり、御指導を頂きました、慶應義塾大学常任理事 村井純博士、同大学環境情報学部教授 徳田英幸博士、同学部教授中村修博士、同学部准教授楠本博之博士に深く感謝致します。

また、絶えず御指導と御助言を頂きました慶義塾大学大学院政策・メディア研究科准教授 朝枝仁博士、同大学SFC研究所上席所員 小川晃通博士、同大学大学院政策・メディア研究科後期博士課程 Achmad Basuki 氏に感謝致します。そして本研究を進めていく上で、様々な励ましと助言、お手伝いをいただきました、慶應義塾大学大学院政策・メディア研究科後期博士課程 三島和宏氏、久松剛氏、松園和久氏をはじめとする Mistral 研究グループのメンバに感謝致します。

大学入学直後から研究室に所属した私にとって、研究室生活が大学生活の全てと言っても過言ではありません。この時間を楽しいものにして下さった徳田・村井・楠本・中村・高汐・重近・バンミーター・植原・三次・中澤合同研究プロジェクトの皆様感謝致します。特に、同じ卒論生として、心強い相談相手になってくれた慶應義塾大学環境情報学部の黒宮佑介氏、立石幹人氏、峯木巖氏に感謝の意を表します。

最後に、今まであらゆる面で多大な助力を頂き、いつも私を支え励ましてくれた家族に心から深謝と敬愛を表し、謝辞と致します。

参考文献

- [1] A. Dutta, H. Schulzrinne, and Y. Yemini, “MarconiNet - an architecture for Internet radio and tv networks,” in Proc. NOSSDAV’99, Basking Ride, NJ, USA, Jun. 1999.
- [2] Achmad Basuki. Estimation Method of Multicast Group size for Large Networks. 修士論文, 慶応義塾大学大学院 政策・メディア研究科, 2007 年度.
- [3] B. Fenner, M. Handley, H. Holbrook, and I. Kouvelas, “Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised),” RFC4601, Aug. 2006. [Online]. Available: <http://www.ietf.org/rfc/rfc4601.txt>
- [4] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, “RTP: A Transport Protocol for Real-Time Applications,” RFC 3550 (Standard), Jul. 2003. [Online]. Available: <http://www.ietf.org/rfc/rfc3550.txt>
- [5] Colin Perkins, マスタリング TCP/IP RTP 編, オーム社 開発局, Apr.2004.
- [6] D.L. Mills, “Network Time Protocol (NTP),” RFC958, Sep. 1985. [Online]. Available: <http://www.ietf.org/rfc/rfc958.txt>
- [7] “An open platform for developing, deploying, and accessing planetary-scale services,” Dec 2007. [Online]. Available: <http://www.planet-lab.org/>
- [8] H. Asaeda, S. Suzuki, K. Kobayashi, and J. Murai, “Architecture for IP Multicast Deployment: Challenges and Practice,” IEICE Transactions on Communications, Vol.E89-B, No.4, pp.1044-1051, Apr. 2006.
- [9] J.Postel, “User Datagram Protocol,” RFC 768(STANDARD), Aug. 1980, [Online]. Available: <http://www.ietf.org/rfc/rfc768.txt>
- [10] “TRANSMISSION CONTROL PROTOCOL DARPA INTERNET PROGRAM PROTOCOL SPECIFICATION,” RFC 793(STANDARD), Sep. 1981, [Online]. Available: <http://www.ietf.org/rfc/rfc793.txt>
- [11] V. Jacobson, “pathchar - a tool to infer characteristics,” April 21, 1997. [Online] <ftp://ftp.ee.lbl.gov/pathchar/>,
- [12] Bob Carter, “Bprobe and cprobe - network probe tools home page,” [Online] <http://cs-people.bu.edu/carter/tools/Tools.html> as of Nov 2008.

- [13] M. Carter, R. Crovella, “Bottleneck link speed in packet-switched networks technical report,” Computer Science Department, Boston University, 1996.