

# 特別研究報告

## 題目

アトラクタ振動モデルを応用したエンド間遅延を安定させる  
送信レート制御手法の提案と評価

## 指導教員

村田 正幸 教授

## 報告者

脇 みどり

平成 23 年 2 月 16 日

大阪大学 基礎工学部 情報科学科

アトラクタ摂動モデルを応用したエンド間遅延を安定させる  
送信レート制御手法の提案と評価

脇 みどり

内容梗概

インターネットは、複数の利用者がネットワークの帯域を共有する、ベストエフォート型のネットワークであり、遅延やパケット棄却などに関するサービス品質（QoS：Quality of Service）は保証されていない。一方、テレビ会議やインターネット電話のようなマルチメディアアプリケーションは通信のリアルタイム性を必要とするため、遅延や遅延のゆらぎに関する QoS 要求が厳しい。しかしながら、ベストエフォートのネットワークにおいては、たとえ一定のレートでネットワークにパケットを送出したとしても、同じ経路を通る他のセッションの数やトラフィック量の変動など、ネットワークに内在する様々なゆらぎの影響を受けるため、エンド間の遅延は一定にはならない。そのためルータでバッファリング、スケジューリングすることでエンド間遅延の変動を抑える手法などが提案されている。しかしながら、ルータでのバッファ予約、スケジューリングのパラメータ設定などのために、トラフィック特性や遅延制約の申告、ネットワークの状態などにもとづく呼受付制御を行う必要があり、これらの条件に大きな変化が生じると遅延が大きく変動し、QoS 要求を満たせなくなる。また、機能や能力の異なる多様なノードが接続され、多様なアプリケーションから様々な変動するトラフィックが送出される環境においては、そもそも適切なネットワークの状態管理と正しいトラフィック特性の申告を前提とした呼受付制御が成り立たない。さらに、遅延の安定化のために新たなルータを多数配置するなど、ネットワーク内部の仕組みを大きく変更することは難しい。そこで本報告では、ゆらぎが内在するシステムにおける外力とシステムの応答との関係を表すアトラクタ摂動モデルを用いた、エンド間遅延を安定させる自律的なレート制御手法を提案する。提案手法では、送信側端末においてエンド間遅延とその分散から送信レートを決定することにより、ネットワークの負荷変動に対して所望のエンド間遅延を達成、維持することができる。シミュレーション評価により、バックグラウンドトラフィックのセッション数が固定のネットワークにおいては、目標とするエンド間遅延を平均二乗誤差 20 以下、変動係数 0.03 以下、ジッタ 15 ms 以下でおおよそ 8 割達成できることを示した。また、バックグラウンドトラフィックのセッション数が変化するネットワークにおいては、約 5 割で

平均二乗誤差 20 以下，変動係数 0.03 以下，ジッタ 15 ms 以下，約 6 割で平均二乗誤差 40 以下，変動係数 0.06 以下，ジッタ 25 ms 以下となり，所望のエンド間遅延をある程度達成，維持できることを示した．

主な用語

送信レート制御 遅延 アトラクタ摂動モデル

## 目次

1	はじめに	7
2	アトラクタ摂動モデルを用いたレート制御手法	10
2.1	対象とするアプリケーションとシステム	10
2.2	アトラクタ摂動モデル	10
2.3	アトラクタ摂動モデルを用いた送信レート制御手法の提案	11
2.3.1	提案手法の概要	12
2.3.2	アトラクタ摂動モデルを用いた送信レート制御手法	12
3	シミュレーション評価	15
3.1	評価モデル	15
3.2	アトラクタ摂動モデルの検証と定数 $b$ の設定	15
3.3	評価指標	17
3.3.1	平均二乗誤差	20
3.3.2	変動係数	20
3.3.3	遅延ジッタ	20
3.4	提案手法の性能評価	21
3.4.1	バックグラウンドの TCP セッション数が一定の場合の評価	21
3.4.2	バックグラウンドの TCP セッション数が変化する場合の評価	37
4	おわりに	46
	謝辞	47
	参考文献	48

## 目 次

1	アプリケーションとシステムのイメージ図 . . . . .	11
2	エンドホスト間での SR・RR の送受信 . . . . .	13
3	ノード数 100 の BA トポロジの例 . . . . .	16
4	850 回分のシミュレーションから求めた $b_i$ と 80%信頼区間 . . . . .	18
5	100 ノードのネットワークにおける定数 $b$ . . . . .	19
6	平均二乗誤差, 変動係数とジッタの評価結果 (TCP セッション数 20, 目標 RTT = TCP 平均 RTT) . . . . .	26
7	平均二乗誤差, 変動係数とジッタの評価結果 (TCP セッション数 30, 目標 RTT = TCP 平均 RTT) . . . . .	27
8	平均二乗誤差, 変動係数とジッタの評価結果 (TCP セッション数 40, 目標 RTT = TCP 平均 RTT) . . . . .	28
9	RTT と送信レートの時間変動 (TCP セッション数 20, 目標 RTT = TCP 平均 RTT, シミュレーション番号 5) . . . . .	29
10	RTT と送信レートの時間変動 (TCP セッション数 20, 目標 RTT = TCP 平均 RTT, シミュレーション番号 25) . . . . .	30
11	RTT と送信レートの時間変動 (TCP セッション数 30, 目標 RTT = TCP 平均 RTT, シミュレーション番号 33) . . . . .	31
12	RTT と送信レートの時間変動 (TCP セッション数 40, 目標 RTT = TCP 平均 RTT, シミュレーション番号 87) . . . . .	32
13	平均二乗誤差, 変動係数とジッタの評価結果 (TCP セッション数 20, 目標 RTT = TCP 平均 RTT の 98%) . . . . .	33
14	平均二乗誤差, 変動係数とジッタの評価結果 (TCP セッション数 30, 目標 RTT = TCP 平均 RTT の 98%) . . . . .	34
15	平均二乗誤差, 変動係数とジッタの評価結果 (TCP セッション数 40, 目標 RTT = TCP 平均 RTT の 98%) . . . . .	35
16	RTT と送信レートの時間変動 (TCP セッション数 30, 目標 RTT = TCP 平均 RTT の 98%, シミュレーション番号 1) . . . . .	36
17	平均二乗誤差, 変動係数とジッタの評価結果 (TCP セッション数 20 から 25 に変化) . . . . .	39
18	平均二乗誤差, 変動係数とジッタの評価結果 (TCP セッション数 25 から 30 に変化) . . . . .	40

19	RTT と送信レートの時間変動 (TCP セッション数 20 から 25, シミュレーション番号 19) . . . . .	41
20	RTT と送信レートの時間変動 (TCP セッション数 20 から 25, シミュレーション番号 21) . . . . .	42
21	RTT と送信レートの時間変動 (TCP セッション数 20 から 25, シミュレーション番号 58) . . . . .	43
22	RTT と送信レートの時間変動 (TCP セッション数 25 から 30, シミュレーション番号 37) . . . . .	44
23	RTT と送信レートの時間変動 (TCP セッション数 25 から 30, シミュレーション番号 8) . . . . .	45

## 表目次

1	パラメータ設定 . . . . .	21
---	-------------------	----

## 1 はじめに

インターネットは、自らの制御規律に従って動作する複数の利用者がネットワーク帯域を共有するベストエフォート型のネットワークであり、遅延やパケット棄却などに関する サービス品質 (QoS: Quality of Service) は保証されていない。そのため、送信側端末から一定の送信レートでパケットを送出しても、ネットワークの状態が変化すると、パケットの転送遅延や棄却率もまた変化する。パケットのエンド間遅延や棄却率に影響を与えるネットワークの状態変化としては、同じ経路を通るセッションの数やトラフィック量の変動、ルーティングテーブルの更新によるネットワーク層での経路の変更、また、無線ネットワークにおいてはリンク品質の変動などが考えられる。

ベストエフォート型のネットワークにおいて信頼性のある通信を行うため、多くのアプリケーションではトランスポート層プロトコルとして TCP (Transmission Control Protocol) [1] を用いている。TCP は、正しく受信できたセグメントに対して受信側端末から返信される ACK にもとづいたフロー制御、輻輳制御を行うとともに、送出するセグメントにシーケンス番号を付加することで、セグメントの棄却を検知し、再送制御を行い、通信の信頼性を高めている。一方、これらの制御の結果、エンド間遅延が不安定になるため、テレビ会議やインターネット電話のようなリアルタイム性を必要とするマルチメディアアプリケーションには不向きである。そのため、H.323 によるテレビ会議やインターネット電話ではトランスポート層プロトコルとして、UDP (User Datagram Protocol) [2] を用いる RTP (Realtime Transport Protocol) [3] を利用している [4, 5]。UDP は、TCP のような輻輳制御や再送制御を行わないため、通信の信頼性は保証されないが、送信側端末の望む送信レートでデータグラムを送出できる。しかしながら、前述の通り、ある一定の送信レートで継続的にデータグラムを送出しても、ネットワークの状態変化の影響を受け、エンド間遅延は一定にならない。

そこで、通信品質保証のないインターネットにおいて、エンド間遅延の安定した高品質なマルチメディア通信を行うため、RTP を用いて得られるパケット棄却率や遅延ゆらぎの観測値からネットワークの状態を推定して、送信レートを調整する手法 [6, 7] や、通信経路上のルータや中間ノードにおいて、遅延や遅延ゆらぎに対するアプリケーションの要求を考慮して、パケット送出手のスケジューリングを行うことで遅延ゆらぎを抑制する手法 [8, 9, 10, 11, 12, 13] などが提案されている。例えば文献 [14] では、ルータでのパケットスケジューリングに jitter-EDD と呼ばれるアルゴリズムを用いることを提案している。jitter-EDD では、ルータごとに定められたパケットの送出期限を考慮してパケット送出手のスケジューリングを行い、前段のルータにおけるパケット送出期限が到来するまで送信処理を行わず、パケットをバッファに留めておくことによって遅延のゆらぎを抑える。しかしながら、jitter-EDD によりエン



ド間遅延を一定に保つためには、理想的には経路上の全てのルータに jitter-EDD を導入しなければならず、また、ルータ間での高精度な時刻同期が必要となる。また、文献 [11] では Jitter Regulator と呼ばれる機器を用いることで、複数のセッションの遅延ゆらぎを同時に解消する手法を提案している。しかしながら、構内網の出入り口など、受信側端末に近接した位置に Jitter Regulator を置く必要があり、そのため、任意の送受信端末間での遅延ゆらぎ抑制を可能にするものではない。ネットワークがますます大規模化していく現状において、新たな機能を有するルータや機器を多数配置することはネットワークをよりいっそう複雑にし、制御や管理の困難さを増すだけでなく、ネットワークの適応性、頑健性の低下を招き、予期しない状態変化に対してネットワークが脆弱になってしまう [15]。したがって、ネットワークの機能に依存することなく、送受信端末における自律的な機構によって遅延のゆらぎを制御できることが望ましい。

このような遅延ゆらぎの除去、抑制のための制御手法が様々に提案されているように、人工的なシステムにおいて、ノイズは有害なものと思われ、フィルタやノイズキャンセラなどによってノイズを除去する努力がなされている。一方、自然界においては、生体は常に内在、外在するゆらぎにさらされており、ノイズやゆらぎは生体活動と不可分のものである [16]。例えば、分子モータなどのナノスケールの構造体では、物質伝達や筋運動において熱ゆらぎが主要な役割を果たしている。また、同じ環境におかれた全く同じ遺伝子配列の細胞間でも遺伝子発現量や基質濃度が異なるが、このことが生体の環境適応性や進化過程における多様性発現の源になっている。

そこで、ノイズを除去するのではなく、積極的に活用することにより、環境変動に対する適応性、頑丈性の高いネットワークやネットワーク制御を実現するアプローチが近年注目されている [17]。動作環境の状態を観測し、その結果に応じて定められた手続きを実施する現在の決定論的制御では、プロトコル設計やシステム構築の段階で遅延などの観測値の時間変動を予測、規定し、その想定範囲にあわせた制御ルールや機構をあらかじめ通信機器に導入しておくことになる。そのため、通信機器やアプリケーションが予期せぬ挙動を示すと、システム全体の極端な機能低下や動作停止が引き起こされてしまう。通信機器やアプリケーションの多様性やトラヒックの時間変動に対処するために、あらゆる起こりうる事象を網羅してノイズやゆらぎを除去する決定論的制御は不十分かつ非現実的であることから、むしろ、効果的にゆらぎを活用することが今後のネットワーク制御において重要である [18]。

システムに内在するゆらぎと制御の関係に関して、近年、生物システムを対象にアトラクタ摂動という概念が提唱されている [19]。アトラクタ摂動モデルでは、ゆらぎのあるシステムへの外力  $a$  を微量  $\Delta a$  だけ変化させたときのシステムの観測値  $x$  の変化量が、外力の変化量  $\Delta a$  と観測値の分散  $\sigma_a^2$  の積に比例することが示されている。すなわち、アトラクタ摂動モデルにもとづく、ゆらぎを持つシステムにおいて、観測値  $x$  の分散から制御  $a$  の変化

量  $\Delta a$  に対する観測値  $x$  の変化量，すなわち制御効果を推定でき，また，観測値  $x$  をある量だけ変化させるために必要な制御  $a$  の変化量  $\Delta a$  を観測値  $x$  の分散から推定することができることとなる．文献 [18] では，このようなアトラクタ摂動モデルのネットワーク制御への応用例として，無線アクセス網のマルチホーミング環境における網選択や，トランスポート層における輻輳制御などのアイデアが示されている．

本報告では，アトラクタ摂動モデルにおいて，外力を送信レート，送受信端末で測定するエンド間遅延を観測値とすることにより，所望のエンド間遅延を達成，維持するための送信レートを決定するレート制御手法を提案する．これにより，ネットワーク内部に特殊なルータや機器を設置することなく，また，ネットワークの動作原理や構造の情報を必要としない，送受信端末による自律的かつ簡便な制御が実現できる．さらに，アトラクタ摂動モデルが成立する環境においては，送信レートを段階的に上げ下げして望むエンド間遅延を達成できるレートを発見的に探索したりせずとも，通信状態の変化に対して適切な送信レートを簡単な式の計算により直接的に決定することができる．

本報告の構成は以下の通りである．まず，2.2 章でアトラクタ摂動モデルの概要について述べる．次に，2 章においてアトラクタ摂動モデルを応用したレート制御手法を提案する．次に，3 章において，シミュレーション評価により，提案手法の有効性を示す．最後に，4 章で本報告のまとめと今後の課題について述べる．

## 2 アトラクタ摂動モデルを用いたレート制御手法

本章では、提案手法の対象とするアプリケーションとシステムについて述べ、アトラクタ摂動モデルの送信レート制御への適用法、およびアトラクタ摂動モデルに基づく送信レートの更新方法について述べる。

### 2.1 対象とするアプリケーションとシステム

提案手法は、リアルタイム性のある通信のためにエンド間遅延に関する QoS を要求するアプリケーションに対して、ネットワーク内部の制御や構造に関する情報を送受信端末が取得することなく、エンド間遅延を安定させることを目的としたレート制御手法である。具体的には、ネットワークとしては、ネットワーク層プロトコルとして IP を用いた、UDP データグラムを任意の送信レートで送出できるベストエフォート型のものを対象とするが、ネットワーク機器やその機能に対して特定の条件は設定せず、QoS 制御機能のあるルータが混在してもよい。また、ネットワークには、固定端末だけでなく、移動端末も接続されるものとする。ルータや端末などのノードの数やトポロジについても制約は設けない。

一方、アプリケーションとしては、テレビ会議やインターネット電話のように、ある程度の時間にわたって RTP/UDP と RTCP (RTP Control Protocol) [3]/UDP を用いた定常的な通信を行い、エンド間遅延に関する QoS を要求するものを対象とする。例えば、ITU-T Y.1541 勧告においては、双方向性が高いマルチメディアアプリケーション向けのクラス 0、クラス 1 について、UNI (User Network Interface) 間遅延として 100 ミリ秒と 400 ミリ秒、遅延ジッタとしていずれのクラスでも 50 ミリ秒が要求されている [20]。また、提案手法では、送信レートを制御することによってエンド間遅延を安定させるため、アプリケーションには設定された送信レートに応じて音声や映像の符号化レートを調整できる機能が求められる [21]。ただし、レート調整可能な範囲に最低レート  $a_{min}$  Mbps から最大レート  $a_{max}$  Mbps の制約があるものとする。また、受信側端末において、ゆらぎ吸収バッファを用いることによりエンド間遅延のゆらぎがアプリケーション品質に与える影響を抑制するアプリケーションを想定する。

### 2.2 アトラクタ摂動モデル

アトラクタ摂動モデルは、バクテリアの進化過程における蛍光タンパク質の量とその表現型の違いとの相関から導出されたモデルであり、内在するゆらぎの影響を受けるシステムにおいて、外力に対する応答の変化量が、システムの変動の大きさに比例することを表している。代謝反応などの生物システムにおけるあらゆる挙動は熱ゆらぎなどの様々なゆらぎの影

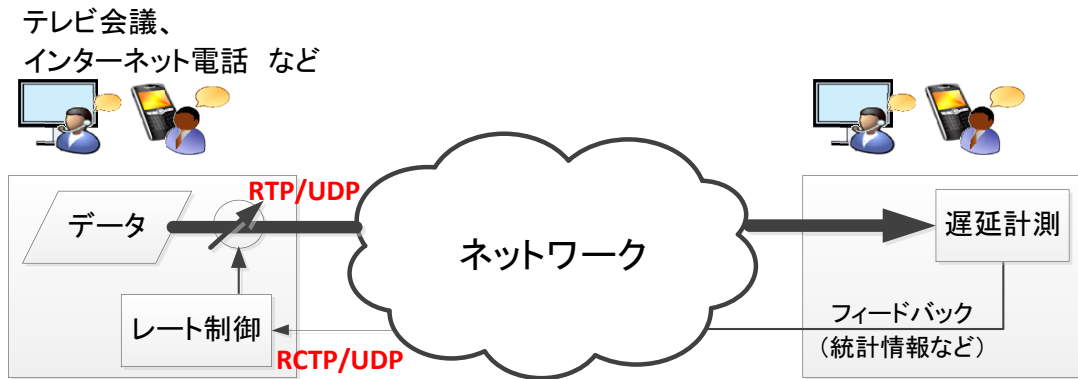


図 1: アプリケーションとシステムのイメージ図

響を受けており、一定の環境においても細胞内の代謝物濃度は時々刻々と変化し、細胞間の代謝物濃度も様々に異なる。文献 [19] では、このような内在するゆらぎが、環境変化に対する生物システムの適応性をもたらしていることを明らかにしている。

文献 [19] の実験においては、細胞間で蛍光タンパク質の量の分散が大きい世代ほど、遺伝子表現型の変化に対して世代の平均蛍光タンパク質量が大きく変化することが示されている。これにもとづき、アトラクタ摂動モデルでは、システムに対して外力  $a$  を加えた際の観測値  $x$  の変化量の関係を次式で与える。

$$\langle x \rangle_{a+\Delta a} - \langle x \rangle_a = b\Delta a\sigma_a^2 \quad (1)$$

式 (1) において、 $\langle x \rangle_a$  は外力の大きさが  $a$  の時の観測値  $x$  の平均値、 $\sigma_a^2$  は分散である。また、 $\Delta a$  は外力の微小変化量、 $b$  は外力の大きさ  $a$  に無関係な定数である。上記の生物実験においては、遺伝子表現型の変化が外力、蛍光タンパク質の量が観測値に対応している。すなわち式 (1) は、摂動  $\Delta a$  に対するシステムの変化量がシステムに内在するゆらぎの大きさに比例することを意味する。これにより、外力  $a$  と観測値  $x$  の関係を直接知ることなく、観測値  $x$  の分散から外力  $\Delta a$  の効果を推定し、その結果に基づく効果的な制御が可能となる。

### 2.3 アトラクタ摂動モデルを用いた送信レート制御手法の提案

本節では、2章で述べたアトラクタ摂動モデルを用いた UDP トラフィックの送信レート制御について、その動作概要と具体的な機構を述べる。

### 2.3.1 提案手法の概要

アトラクタ摂動モデルは，外力の変化量  $\Delta a$  に対する観測値  $x$  の変化量が，観測値の分散  $\sigma_a^2$ ，すなわちシステムに内在するゆらぎの大きさに比例することを表している．そこで，アトラクタ摂動モデルにおけるシステムをネットワーク，外力を送信レート，観測値をエンド間遅延と対応づけることにより，エンド間遅延を安定させるレート制御を実現する．

提案手法では，ある一定間隔でレート制御を行う．レート制御のタイミングにおける送信レート  $a$  に対して，それまでの通信におけるエンド間遅延の平均値を  $\langle x \rangle_a$ ，分散を  $\sigma_a^2$  としたとき，安定化の目標とするエンド間遅延を  $\langle x \rangle_{a+\Delta a}$  とし，送信レートの変更量  $\Delta a$  を， $\Delta a = (\langle x \rangle_{a+\Delta a} - \langle x \rangle_a) / (b\sigma_a^2)$  により導出する．なお，本報告で提案するレート制御手法は，RTP/UDP を用いてデータを送信するアプリケーションを対象としており，送信側端末が，RTCP の Sender Report メッセージを受信側端末に送信することによって返信される Receiver Report メッセージから，RTT (Round Trip Time) を計算し，エンド間遅延とする．以後，エンド間遅延を RTT，所望のエンド間遅延を目標 RTT とよぶ．アトラクタ摂動モデルによれば，目標 RTT に対して現在の RTT が大きいとき，目標 RTT を達成するためには，分散  $\sigma_a^2$  の大きい状態では送信レートを少しだけ下げればよく，一方，分散  $\sigma_a^2$  が小さい状態では送信レートを大きく下げなければならない．前者は，RTT の分散が大きいことからネットワークの負荷状態が中程度であると考えられ，送信レートのわずかな変更により，エンド間遅延を容易に下げられることを，一方，後者の場合には，ネットワークが高負荷状態にあると考えられるため，大きく送信レートを下げなければ RTT を下げられないということを，それぞれ示唆している．

### 2.3.2 アトラクタ摂動モデルを用いた送信レート制御手法

送信側端末は，アプリケーションの設定した最低レートである  $a_{min}$  Mbps を送信レートとして通信を開始する．送信側端末は，送信レート  $a$  Mbps と定数  $N$  ( $N > 0$ ) の積以上の RTP パケットを送信するごとに，受信側端末に向けて RTCP の Sender Report パケット (以降，SR パケットと表記する) を送信する．なお， $i$  番目に送信した SR パケットのヘッダには送信時刻  $S_i$  が timestamp フィールドに記録される．定数  $N$  により，送信側端末は送信レートに比例した数の RTP パケットごと，すなわち一定時間ごとに SR パケットを送信することになる．なお，送信側端末でタイマを用いて SR パケットの送信間隔を制御してもよい．

$i$  番目の SR パケットを受信した受信側端末は，Receiver Report パケット (以降，RR パケット) を生成し，送信側端末へ送信する．RR パケットの Last SR timestamp (LSR) フィールドには受信した SR パケットのタイムスタンプ  $S_i$ ，Delay since Last SR (DLSR) フィー

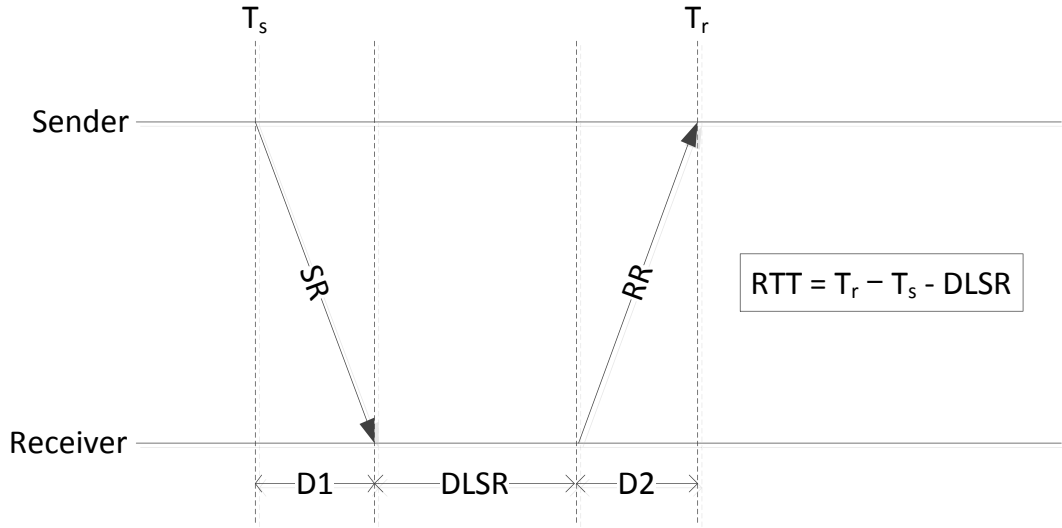


図 2: エンドホスト間での SR・RR の送受信

ルドには SR パケットを受信してから RR パケットを送信するまでに要した時間  $D_i$  がそれぞれ格納される。

RR パケットを受信した送信側端末は，受信時刻  $R_i$  とヘッダに含まれる情報から，次式を用いて  $i$  番目の SR パケットによって計測された RTT  $t_i$  を得る。

$$t_i = R_i - S_i - D_i \quad (2)$$

送信側端末は，通信開始から  $K$  個目 ( $K > 0$ ) の RR パケットを受信すると，最初の送信レート制御を行う。以降，RR パケットを受信するごとに，前回の送信レート制御から  $I$   $s$  ( $I > 0$ ) 以上経過していれば，送信レートの制御を行う。送信側端末が  $i$  番目 ( $i \geq K$ ) の SR パケットに対する RR パケットを受信したものとする。送信レートの制御においては，まず，受信した RR パケットから算出されるものも含めた過去  $K$  個分の RTT の観測値  $t_k$  ( $i - K + 1 \leq k \leq i$ ) から RTT の相加平均  $\bar{t}_i$  と分散  $\sigma_i$  を次式でそれぞれ求める。

$$\bar{t}_i = \frac{1}{K} \sum_{k=i-K+1}^i t_k \quad (3)$$

$$\sigma_i = \frac{1}{K} \sum_{k=i-K+1}^i (t_k - \bar{t}_i)^2 \quad (4)$$

次に、アプリケーションの要求する RTT  $\hat{T}$  s と定数  $b$  ( $b > 0$ ) から、次式により送信レートの変更量  $\Delta a$  を決定する。なお、急激なレート変更を抑制するため、送信レートの変更量には上限  $\Delta_{max}$  Mbps と下限  $\Delta_{min}$  Mbps を設ける。

$$\Delta a = \min(\Delta_{max}, \max(\Delta_{min}, \frac{\hat{T} - \bar{t}_i}{b\sigma_i})) \quad (5)$$

さらに、アプリケーションの設定する送信レートの範囲にしたがって、新たな送信レート  $\acute{a}$  Mbps を次式で決定し、送信レートを  $\acute{a}$  Mbps に変更する。

$$\acute{a} = \min(a_{max}, \max(a_{min}, a + \Delta a)) \quad (6)$$

### 3 シミュレーション評価

本章では、シミュレーション評価により、提案手法の有効性を示す。シミュレーションは ns-2 [22] を用いて行った。

#### 3.1 評価モデル

100 ノードからなるランダムな BA トポロジのネットワークを用いる。BA トポロジは BA (Barabási-Albert) モデルにより生成されるネットワークトポロジである [23]。BA モデルでは、2 台のノードが 1 本のリンクで接続された初期トポロジに対して、ノードを順次追加していくことによってネットワークを構成する。追加されたノードがノード数  $n$  のネットワーク内のあるノード  $i$  ( $1 \leq i \leq n$ ) に接続される確率を、ノードの次数  $d_i$  に比例した  $d_i / \sum_{j=1}^n d_j$  で与えることにより、次数分布が  $p(k) \propto k^{-\gamma}$  に従うスケールフリー性を持つトポロジとなる。その結果、図 3 に示されるような、多数の隣接ノードを持つ少数のハブノードと、少数の隣接ノードを持つ多数のノードが生まれる。現在のインターネットトポロジはスケールフリー性を持つことが近年明らかになっている [24]。シミュレーションでは、全てのリンクが全二重であり、帯域を 10 Mbps、伝搬遅延を 10 ms とした。また、ノードは FIFO、DropTail で管理されたバッファサイズが 100 パケットのバッファを有している。

提案手法のセッションを、ランダムに選んだノード間で設定する。図 3 の場合には、赤色で示されたノードが送受信端末を表している。提案手法で送信されるデータパケットのサイズは、RTP、UDP、IP ヘッダを含めて 1028 byte、また RTCP の SR、RR のパケットサイズはそれぞれ IP ヘッダを含めて 64 byte、72 byte である。また、バックグラウンドトラヒックとして、任意の本数の TCP セッションを、セッション間および提案手法のセッションと重複しないランダムに選ばれたノード間に設定した。TCP セッションのアプリケーションは FTP とし、継続的にファイル転送を行う。TCP セッションで送信されるデータパケット、ACK パケットのサイズは IP ヘッダを含めてそれぞれ 1040 byte、40 byte である。

#### 3.2 アトラクタ摂動モデルの検証と定数 $b$ の設定

3.1 節で定義したシミュレーションモデルにおいて、外力としての UDP トラヒックの送信レートと、観測値としての RTT との相関の有無を確認するとともに、相関が認められ、アトラクタ摂動モデルが成立する場合には、定数  $b$  として適当な値を決定する。

具体的には、あるランダムに生成されたトポロジ、セッションの組み合わせ  $j$  ( $1 \leq j \leq W$ ) に対して、バックグラウンドトラヒックとして 49 本の TCP セッションを設定し、提案手法の代わりに送信レートを  $i$  Mbps に固定した CBR トラヒックを流すシミュレーションを、送



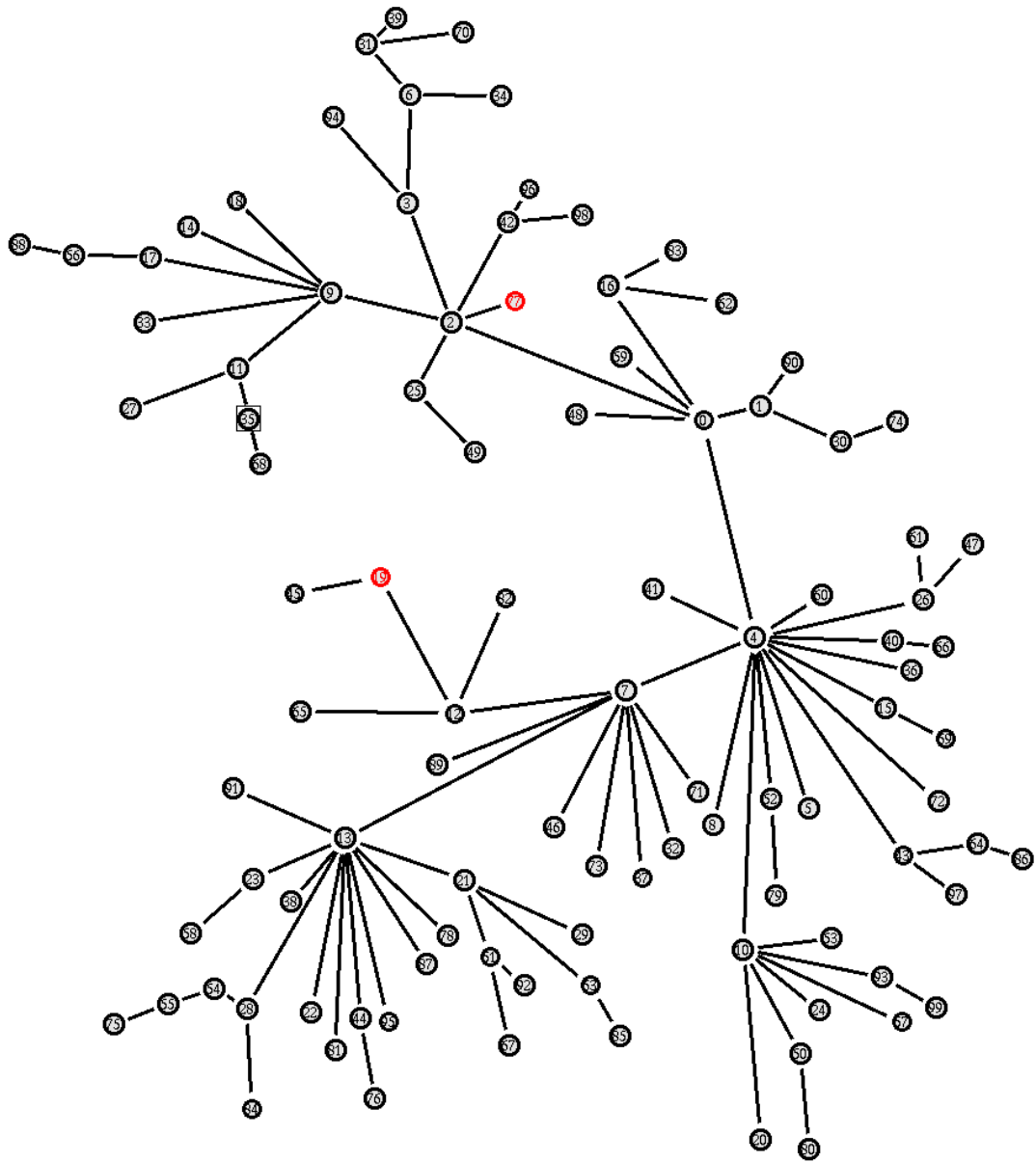


図 3: ノード数 100 の BA トポロジの例

信レートを 1 Mbps から 10 Mbps に 1 Mbps ずつ増加させて実施する。まず、シミュレーションによって得られた送信レート  $i$  Mbps における RTT の平均  $\langle x \rangle_i^j$  と、RTT の分散  $\sigma_i^j$  から、次式を用いて定数  $b_i^j$  を求める。

$$b_i^j = \frac{\langle x \rangle_{i+1}^j - \langle x \rangle_i^j}{\sigma_i^j} \quad (7)$$

次に、トポロジ、セッションの組み合わせについて  $b_i^j$  の平均値  $b_i$  を次式で求める。

$$b_i = \frac{1}{n} \sum_{k=1}^W b_i^k \quad (8)$$

このようにして求めた  $b_i$  が、トポロジやセッション設定、CBR の送信レートによらず、ある一定の値であれば、2.2 章で述べたアトラクタ摂動モデルが成立しているとみなすことができる。

17 通りのランダムな BA トポロジに対してそれぞれセッションの組み合わせをランダムに 50 通りに変え ( $W = 850$ )、得られた結果から求めた  $b_i$  ( $1 \leq i \leq 9$ ) を図 4 に示す。図に示すとおり、 $b_i^j$  のばらつきが大きく、 $b_i$  が変動していることがわかる。シミュレーション結果を確認したところ、極端な外れ値が含まれていたため、送信レートあたり 850 回のシミュレーション結果から、 $-100000 < b_i^j < 100000$  が成り立つ 763 回の結果を用いた場合の平均値と 80%信頼区間、 $-10000 < b_i^j < 10000$  が成り立つ 732 回の結果を用いた場合の平均値と 95%信頼区間をそれぞれ図 5(a)、図 5(b) に示す。ばらつきはあるものの、86%以上のシミュレーションにおいて、定数  $b_i$  がおよそ 100 であることから、送信レートを外力、RTT を観測値としたアトラクタ摂動モデルがおよそ成立しているものとする。以降では、提案手法における定数  $b$  を 100 として、シミュレーションを行う。なお、ある送信レート  $i$  Mbps において外れ値となるシミュレーションにおいては、他の送信レートでも極端に大きな正または負の外れ値になる傾向が高いことから、 $b$  の値はトポロジとセッションの組み合わせに依存すること考えられる。外れ値が生じるのは、CBR セッションの通る経路を競合する TCP セッションがほとんどない組み合わせであり、遅延の変動が小さく、分散が 0 に近くなるためであると考えられる。したがって、本シミュレーション条件では、UDP セッションから見てネットワークがある程度以上の負荷状態にあるときにのみ、アトラクタ摂動モデルが成立し、アトラクタ摂動モデルを用いたレート制御が利用可能なことがわかる。

### 3.3 評価指標

本節では、提案するレート制御により観測される平均 RTT の目標 RTT への達成度、安定度合いを評価するための評価指標について述べる。

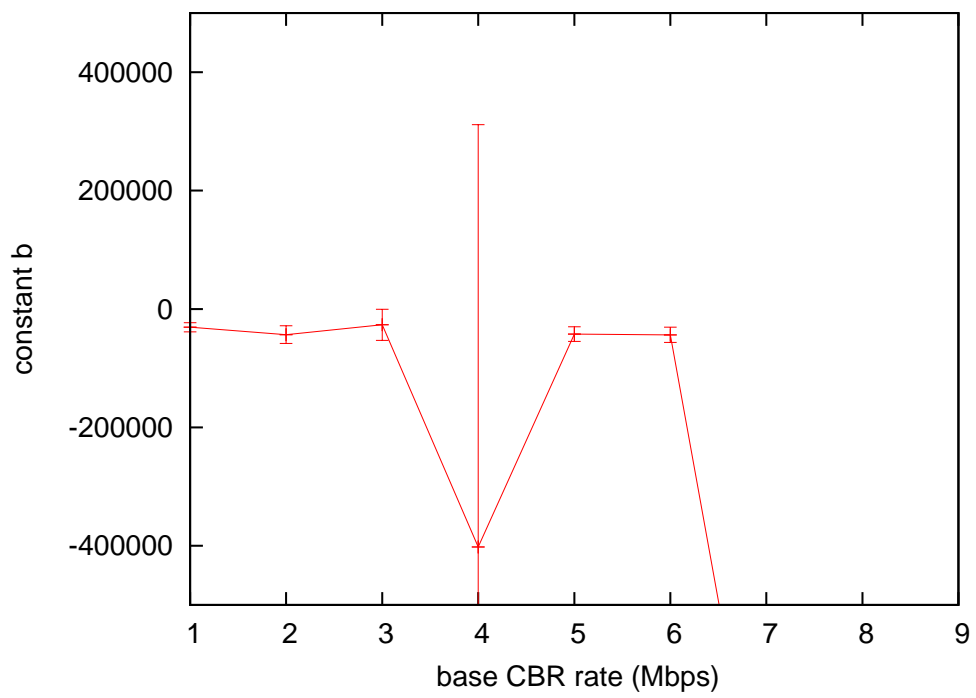
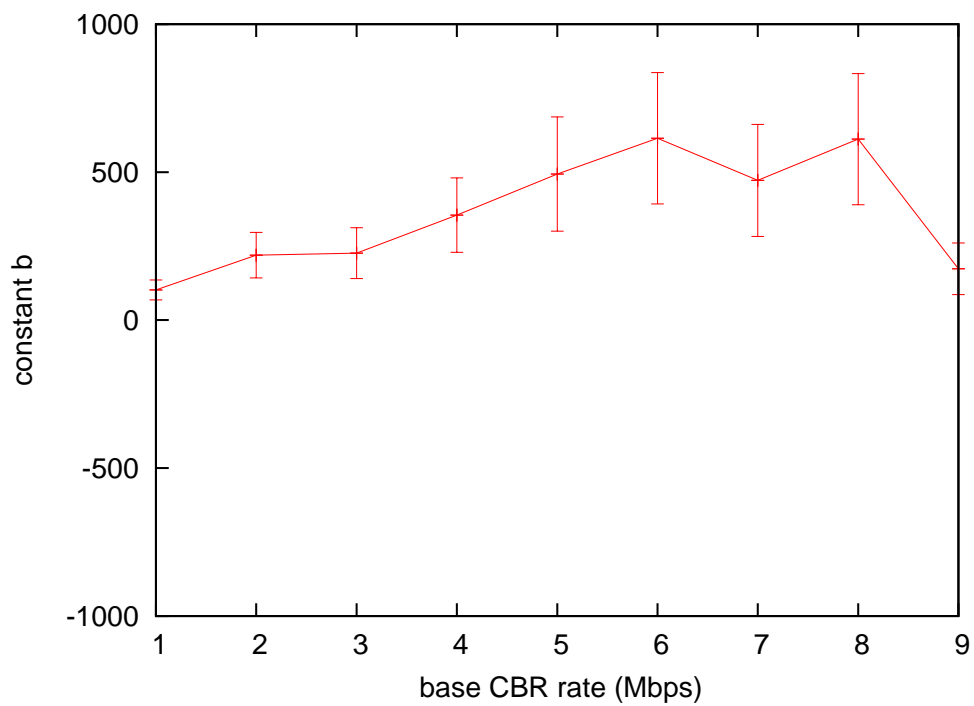
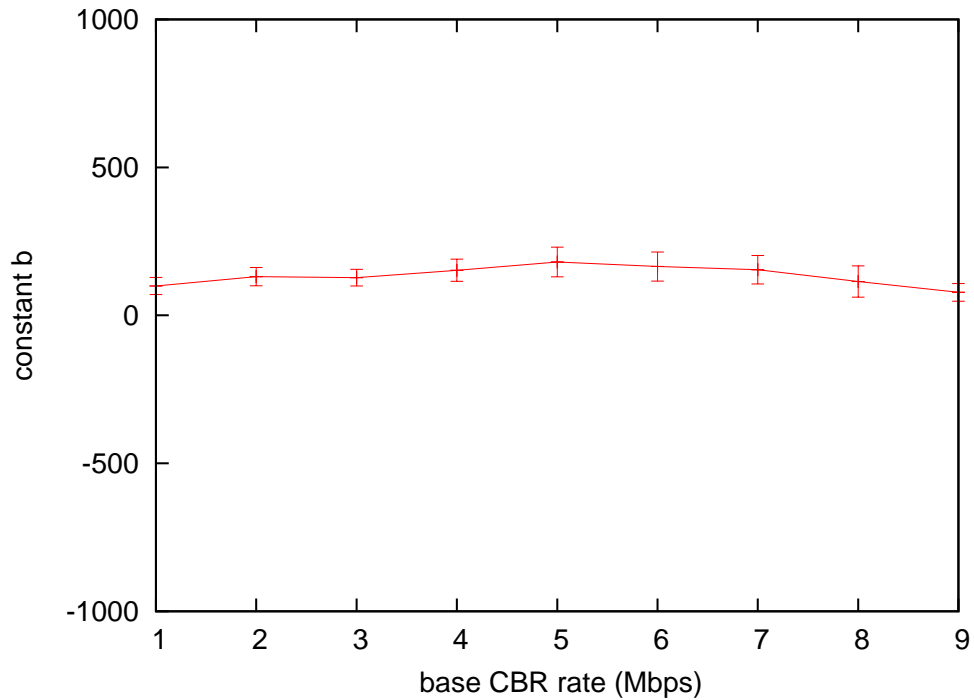


図 4: 850 回分のシミュレーションから求めた  $b_i$  と 80%信頼区間



(a) 763 回分のシミュレーションから求めた  $b_i$  と 80%信頼区間



(b) 732 回分のシミュレーションから求めた  $b_i$  と 95%信頼区間

図 5: 100 ノードのネットワークにおける定数  $b$

### 3.3.1 平均二乗誤差

目標 RTT の達成度を、以下で求める平均二乗誤差 (MSE : Mean Square Error) により評価する。あるシミュレーションにおいて、送信側端末が受信した  $i$  個目の RR メッセージから算出される RTT を  $t_i$  とする。こうして得られた直近 100 個の RTT の観測値について単純移動平均 (SMA: Simple Moving Average) を求めたものを  $T_i$  とし、次式で求める。

$$T_i = \frac{1}{100} \sum_{k=i-99}^i t_k \quad (9)$$

なお、シミュレーションで得られた RTT の観測値の個数を  $n$  とすると、 $i$  は  $100 \leq i \leq n$  である。このシミュレーションにおける目標 RTT を  $\hat{T}$  とすると、平均二乗誤差  $M$  を次式で求める。

$$M = \frac{1}{n-99} \sum_{k=100}^n (T_k - \hat{T})^2 \quad (10)$$

平均二乗誤差  $M$  が小さいほど、多くの時間において目標 RTT に近い RTT が達成できていることを意味する。

### 3.3.2 変動係数

RTT の安定性を、以下で求める変動係数 (CV : Coefficient of Variation) を用いて評価する。前節で定義した指標を用いて、あるシミュレーションにおける遅延の相加平均  $T$  と、標準偏差  $\sigma$  をそれぞれ次式で求める。

$$T = \frac{1}{n-99} \sum_{k=100}^n T_k \quad (11)$$

$$\sigma = \sqrt{\frac{1}{n-99} \sum_{k=100}^n (T_k - T)^2} \quad (12)$$

これより、このシミュレーションにおける変動係数  $C$  を次式で求める。変動係数  $C$  が小さいほど RTT が安定していることを意味する。

$$C = \frac{\sigma}{T} \quad (13)$$

### 3.3.3 遅延ジッタ

受信側端末が設定、管理するゆらぎ吸収バッファの必要サイズなどに対する影響を評価するため、以下で求めるジッタを用いる。3.3.2 節で定義した指標により、あるシミュレーショ

ンにおけるジッタ  $J$  を次式で求める．

$$J = \max_{100 \leq k \leq n} \{|T_k - \hat{T}|\} \quad (14)$$

すなわち，ジッタは目標 RTT に対する得られた RTT の最大誤差である．例えば，平均二乗誤差  $M$  と変動係数  $C$  がともに小さいが，ジッタ  $J$  が大きい場合には，一時的な輻輳などによって瞬時的に RTT が変動していると考えられる．

### 3.4 提案手法の性能評価

本節では，提案手法による送信レート制御によって目標 RTT を達成，維持できることを示す．まず，3.4.1 節において，バックグラウンドトラフィックである TCP セッション数がシミュレーションを通して一定の場合に，提案手法による送信レート制御によって，目標 RTT が達成，維持できることを示す．次に，3.4.2 節において，バックグラウンドの TCP のセッション数が変化した場合に，その変化に応じて送信レートを制御し，セッション数が変化する前に達成していた目標 RTT を維持できることを示す．本節で行うシミュレーション評価では，パラメータとして表 1 に示すものを用いる．

表 1: パラメータ設定

パラメータ	値
定数 $N$	10
定数 $K$	100 [個]
送信レート更新間隔 $T$	3 [s]
定数 $b$	100
レート変更の下限 $\Delta_{min}$	-1 [Mbps]
レート変更の上限 $\Delta_{max}$	+1 [Mbps]
最小送信レート $a_{min}$	100 [kbps]
最大送信レート $a_{max}$	9.9 [Mbps]

#### 3.4.1 バックグラウンドの TCP セッション数が一定の場合の評価

提案手法を用いることで，アプリケーションが要求する RTT を達成できることを示すために，バックグラウンドの TCP セッション数をシミュレーション時間を通して一定にした評価を行う．TCP セッション数が 20 本，30 本，40 本のそれぞれの場合について，トポ

ロジをランダムに 10 通り生成し，トポロジごとにランダムに 10 通りのセッションの組み合わせを設定することにより，100 回ずつのシミュレーションを行った．目標 RTT としては，提案手法の評価に用いるものと同じトポロジ，セッションの組み合わせにおいて，提案手法による通信を行うノード間に，提案手法の代わりに TCP を用いた場合の，シミュレーションを通じた平均 RTT，または TCP の平均 RTT よりも 2% 小さい値を用いる．なお，シミュレーション時間は 300 s とし，シミュレーション開始後，1 s までにバックグラウンドの TCP セッションがランダムに通信を開始し，1 s に提案手法を用いたセッションが通信を開始する．

図 6，図 7，図 8 に，それぞれ TCP セッション数を 20 本，30 本，40 本，目標 RTT を自身が TCP の場合の平均 RTT とした場合の，100 回のシミュレーションのそれぞれにおける変動係数  $C$  と平均二乗誤差  $M$  の関係，および変動係数  $C$  とジッタ  $J$  の関係をそれぞれ示す．それぞれの図において丸印は平均値を表す．図 6(a)，図 7(a)，図 8(a) より，いずれの TCP セッション数についても，多くの場合において，平均二乗誤差が小さく，目標 RTT を達成できていることがわかる．また，変動係数が小さいことから，提案手法によって得られる平均 RTT がおおよそ安定していると考えられる．また，目標 RTT と平均 RTT の差の最大値を表すジッタは，たかだか 25 ms である．すなわち，パケットが最も早く届いたときと最も遅く届いたときの差が 50 ms よりも小さいことを意味しており，これは ITU-T Y.1541 勧告における遅延ジッタの要求を満たしている．

図 9 に，TCP セッション数が 20 本の場合の，あるシミュレーションにおける RTT と送信レートの変動を示す．図 9(a) には自身が TCP を用いて通信した場合の，ACK 受信によって算出される RTT と過去 9 秒間の RTT の移動平均の変化，および平均 RTT，すなわち提案手法の目標 RTT を示している．なお，9 秒は，提案手法で RTCP メッセージの送受信により 100 個の RTT の観測値を得るために必要な時間である．図に示されるとおり，TCP を用いた場合においても，セグメントごとの平均 RTT の変動は小さい．これは，バックグラウンドトラフィックである TCP のセッション数が比較的少なく，ネットワークに負荷がかかっていない状態であるためである．バッファにパケットがほとんど蓄積しないため，観測される RTT はおおよそ往復伝搬遅延に等しくなっている．図 9(b) には，提案手法について，上記平均 RTT を目標 RTT としたときの，RR メッセージ受信により算出される RTT と直近 100 個の RTT の観測値の移動平均の変化，および目標 RTT を示しているが，同様の理由で，バックグラウンドトラフィックの影響をほとんど受けないため，目標 RTT を達成，維持できている．また，このような環境においては，リンクが空いているため，送信レートを上げてもバッファにパケットがあまり溜まらず，RTT がすぐには大きくなる．しかしながら，送信側ノードが送信レートを上げ続けると，いずれバッファリング遅延によって RTT が大きくなり，送信ノードは送信レートを下げる．このような振る舞いが原因で，図 9(c) に示さ

れるように送信レートが変動しているものと考えられる．なお，図 9 に示されたシミュレーションにおける平均二乗誤差は 0.0239，変動係数は 0.0012，ジッタは 0.4948 ms であった．

一方，図 10 は同じく TCP セッション数が 20 本の場合の別のシミュレーションの結果を示しているが，目標 RTT には到達しているものの変動が大きく，平均二乗誤差 20.489，変動係数 0.02，ジッタ 14.015 ms である．図 9(a) と比較して，図 10(a) では TCP の平均 RTT が大きく，また，変動していることがわかる．これは，経路上のリンクを共有している TCP セッションが複数存在し，バッファにパケットがある程度蓄積しているためである．このような環境で提案手法を用いた場合，図 10(b) に示されるように提案手法の平均 RTT も大きく変動してしまう．バックグラウンドの TCP のトラヒックがフロー制御，輻輳制御，再送制御によって変化すると，提案手法での RTT の観測値も変化し，提案手法は送信レートを変更する．この送信レート制御は TCP の制御を引き起こし，さらに RTT が大きく変動するため，提案手法は再度送信レートを変更する．このようなバックグラウンドトラヒックとの相互作用によって平均 RTT が不安定になる．なお，図 9(c) に示されるように送信レートが変動しているにも関わらず，図 9(b) の平均 RTT がほぼ一定であるのは，競合する TCP セッションがほとんどなく，提案手法自身の送信レート制御によってバッファ内パケット数が増減するため，平均 RTT の変化に対して制御が即応できることによる．一方，図 10(b) では，提案手法で送信レートを上げ下げしても，TCP セッションから送出されたパケットとともにバッファに蓄積されるためバッファ内パケット数に与える影響が小さく，また，送信レート変更の影響が TCP セッションの制御を介してフィードバックされるため，平均 RTT の変動に対する制御に遅れが生じている．ただし，図 10(b) のように平均 RTT の変動が大きい結果は多くなく，平均二乗誤差  $M$  が 20 以上，変動係数  $C$  が 0.02 以上，ジッタ  $J$  が 14 ms 以上のものは 100 回のシミュレーション中 19 回である．また，丸印で表された，TCP セッション数 20 本の全シミュレーションの結果の平均値は，平均二乗誤差 4.244，変動係数 0.00937，ジッタ 3.185 ms である．

図 10 に示されるようなバックグラウンドの TCP セッションとの相互作用による平均 RTT の変動は，セッション数が 30 の場合により多く発生するため，図 6 と図 7 を比較すると，より右上に点が多く現れている．ただし，図 7 では，平均二乗誤差 5，変動係数 0.01 付近に点が多く集まっており，図 10 の結果と比較してその変動や目標 RTT からの誤差は小さい．TCP セッション数 30 本の場合における平均値は，平均二乗誤差 8.791，変動係数 0.01634，およびジッタ 5.844 ms である．そのうちの一つのシミュレーション結果を図 11 に示す．図 11 では，平均二乗誤差は 3.799，変動係数は 0.0136，ジッタは 5.7492 ms であり，TCP セッション数が 30 本の場合でも，目標 RTT を安定して達成できていることがわかる．

さらに TCP セッション数が増えた場合の図 8 を，図 7 と比較すると，変動係数は小さくなる一方で，平均二乗誤差が増加する傾向にある．図 8 に示す平均値はそれぞれ，平均二



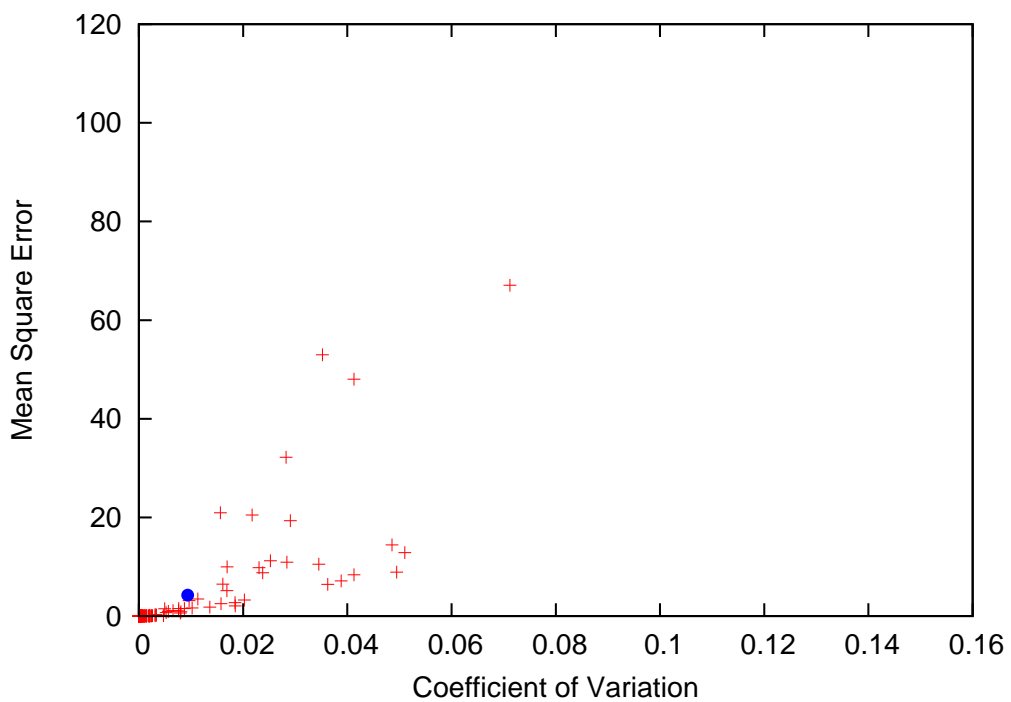
乗誤差 9.587, 変動係数 0.0165, ジッタ 6.089 ms である。平均二乗誤差 49.068, 変動係数 0.0035, ジッタ 7.498 ms であったシミュレーションの結果を図 12 に示す。図 12(a) に示されるように, TCP の RTT の変動が大きい, これはネットワークの負荷が高いため, パケット棄却が頻発し, TCP セッションが輻輳ウィンドウを大きく変更するためである。提案手法を用いた場合, 送信レートを下限である 100 kbps まで下げているが (図 12(c)), 目標 RTT に達していない。そのため, 平均二乗誤差は大きくなるものの, 変動係数は小さいという結果となっている。

次に, 目標 RTT を TCP の平均 RTT の 98% に設定した場合の結果を, TCP セッション数 20 本, 30 本, 40 本のそれぞれについて図 13, 図 14, 図 15 に示す。なお, 20 本, 30 本の結果は, 外れ値を除くそれぞれ 98 回, 99 回のシミュレーション結果である。図 6, 図 7, 図 8 とそれぞれ比較して, いずれの TCP セッション数についても, 平均二乗誤差がより大きく, 変動係数がより小さく, ジッタがより大きくなる傾向が見られる。セッション数が 20 本の場合については, そもそもバッファにあまりパケットが蓄積されておらず, 往復伝搬遅延に近い値であった目標 RTT をさらに 2% 下げたため, 送信レートを最低レートである 100 kbps に設定しても達成が困難である。そのため, 平均二乗誤差は大きくなる。また, TCP セッションとの競合もなく, また, 自身の送信レートも変更しないため, RTT は安定しており, 変動係数は 0 に近くなる。さらに, 目標 RTT を TCP の平均 RTT とした場合と同様に, TCP セッション数が増えるといずれの評価指標も大きくなる。TCP セッション数が 30 本, 40 本の場合のいずれも, ほとんどの結果において, 送信レートが 100 kbps まで低下しているが, 目標 RTT は達成できていない。図 16 は TCP セッション数が 30 本の場合のあるシミュレーションの結果であるが, 平均二乗誤差 18.658, 変動係数 0.0007, ジッタ 4.878 ms である。なお, 図 16(a) における直線は, TCP の平均 RTT の 98% を表している。図に示されるとおり, 提案手法では目標 RTT を達成できていないが, 平均 RTT は安定している。

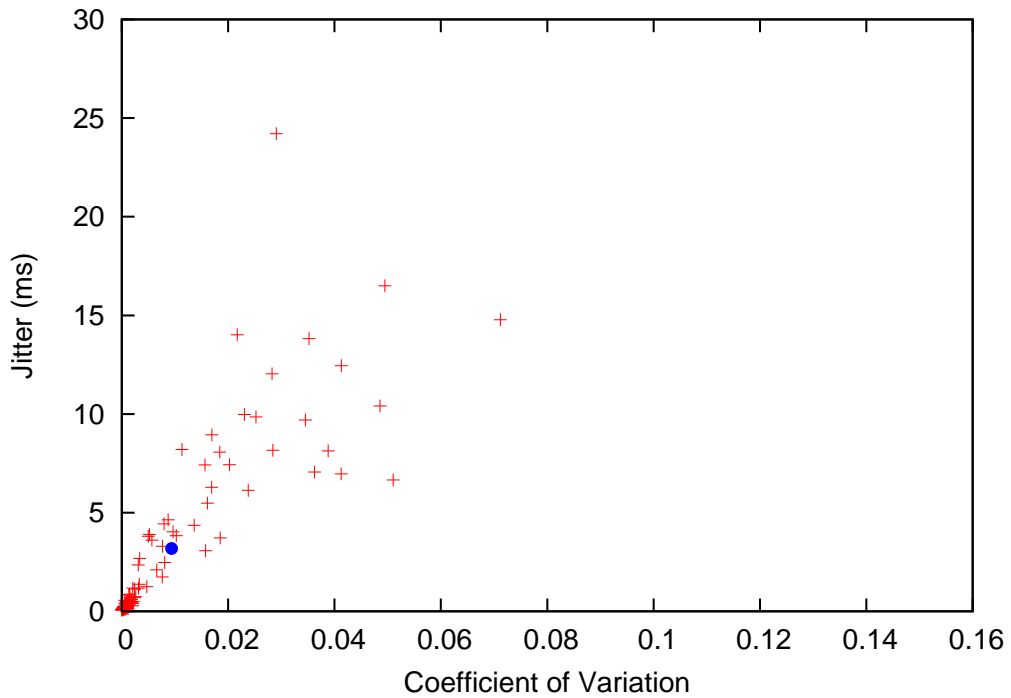
以上より, 提案手法による送信レート制御により, TCP と同じ RTT をおおよそ達成することができることが示された。目標 RTT を TCP の平均 RTT とする場合, 平均二乗誤差が 20 以下, 変動係数 0.03 以下, ジッタ 15 ms 以下を目標 RTT 達成と定義すると, バックグラウンドの TCP セッションが 20 本では 85%, 30 本では 84%, 40 本では 71% 達成可能であることがわかった。また, 目標 RTT を自身が TCP の場合の平均 RTT の 98% に設定した場合には, TCP セッション数が 20 本, 30 本, 40 本のそれぞれについて, 85%, 84%, および 78% で達成している。目標 RTT を TCP の平均 RTT よりも小さく設定した場合については, 本報告のシミュレーション条件では, 下限まで送信レートを下げても目標 RTT を達成できないため, TCP と同じか少し大きい値に設定する必要がある。

バックグラウンドの TCP セッション数によっては, 提案手法における平均 RTT が, TCP の場合の平均 RTT よりも大きく変動する場合が認められるが, 次節では, バックグラウン

ドトラヒックが変化した場合に，提案手法が目標 RTT を維持できることを示す．

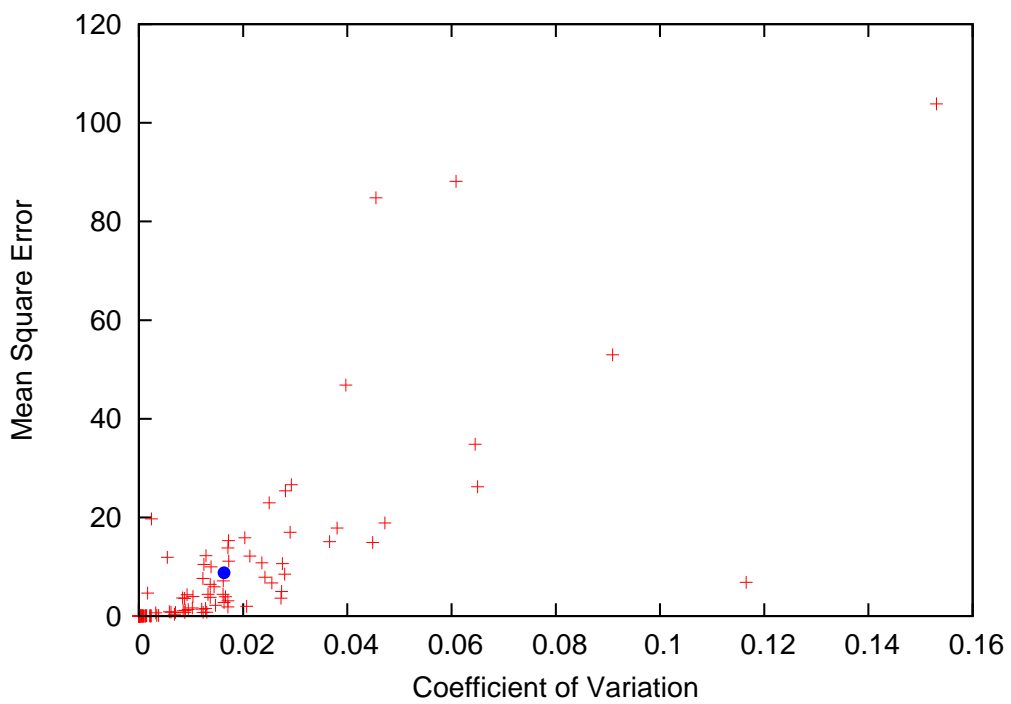


(a) 変動係数  $C$  と平均二乗誤差  $M$  の関係

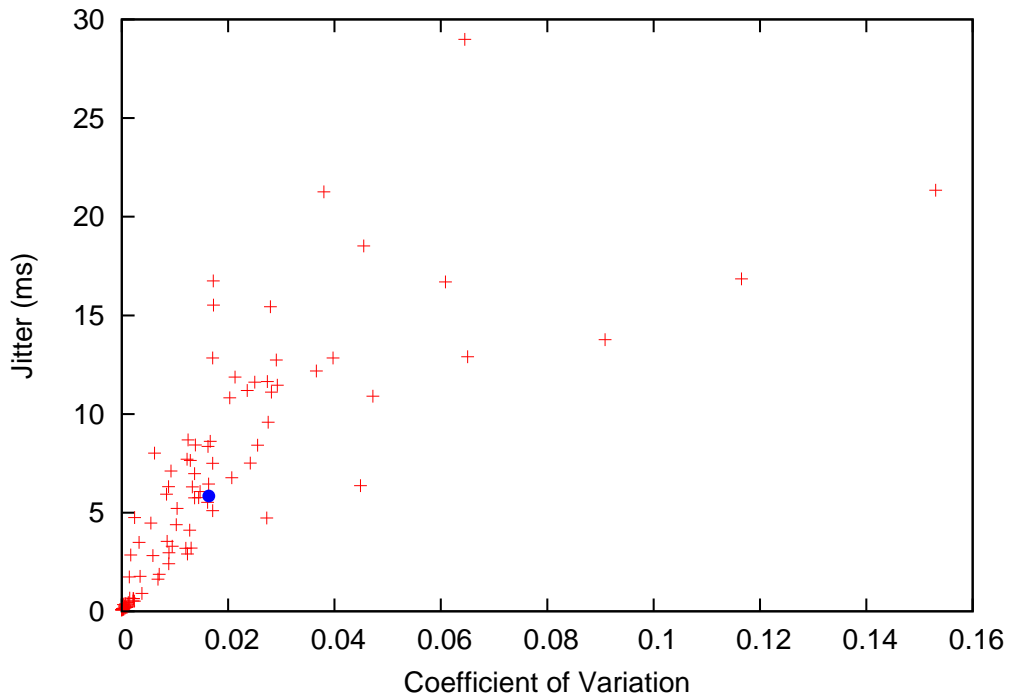


(b) 変動係数  $C$  とジッタ  $J$  の関係

図 6: 平均二乗誤差, 変動係数とジッタの評価結果 (TCP セッション数 20, 目標 RTT = TCP 平均 RTT)

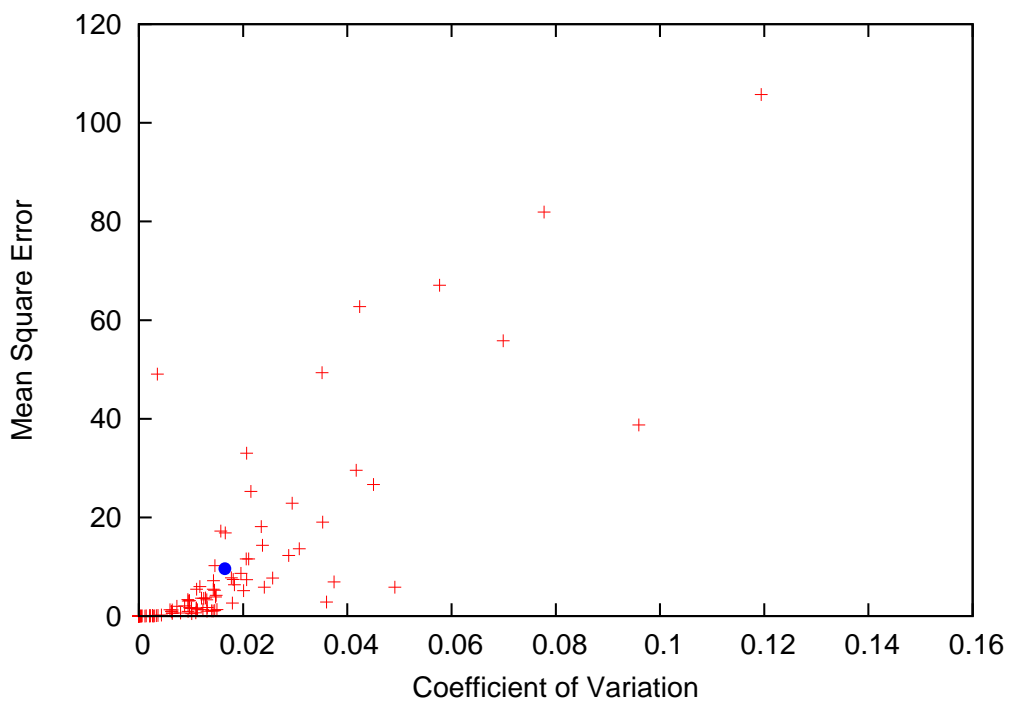


(a) 変動係数  $C$  と平均二乗誤差  $M$  の関係

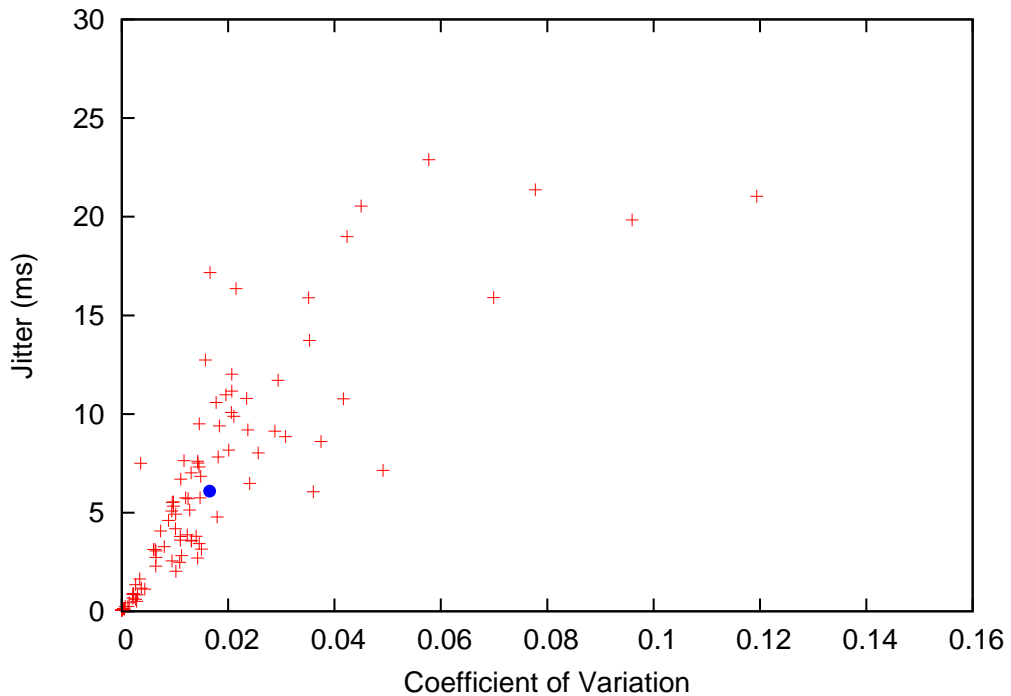


(b) 変動係数  $C$  とジッタ  $J$  の関係

図 7: 平均二乗誤差, 変動係数とジッタの評価結果 (TCP セッション数 30, 目標 RTT = TCP 平均 RTT)

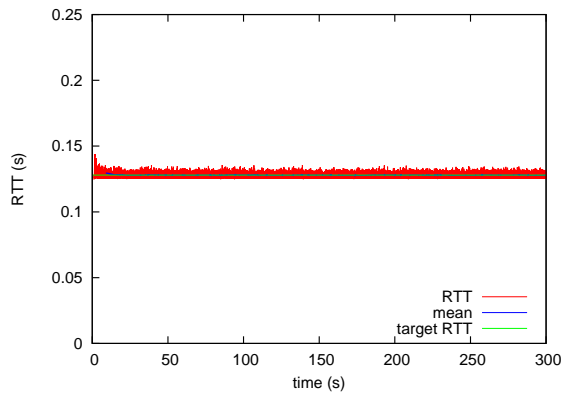


(a) 変動係数  $C$  と平均二乗誤差  $M$  の関係

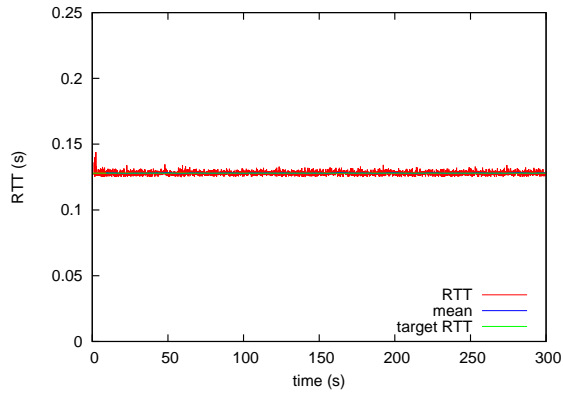


(b) 変動係数  $C$  とジッタ  $J$  の関係

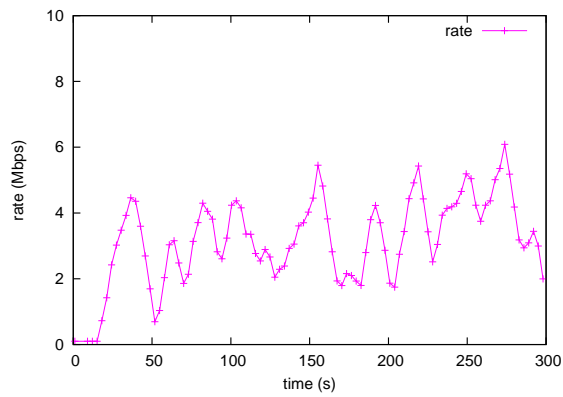
図 8: 平均二乗誤差, 変動係数とジッタの評価結果 (TCP セッション数 40, 目標 RTT = TCP 平均 RTT)



(a) 自身が TCP の場合の RTT

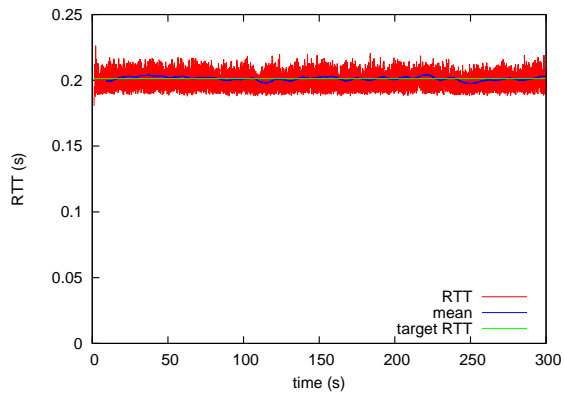


(b) 提案手法における RTT

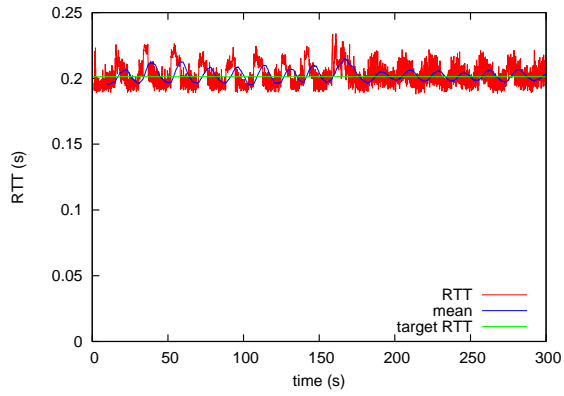


(c) 提案手法における送信レートの変化

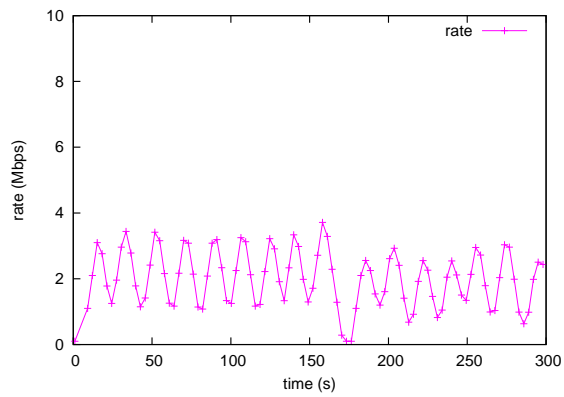
図 9: RTT と送信レートの時間変動 (TCP セッション数 20, 目標 RTT = TCP 平均 RTT, シミュレーション番号 5)



(a) 自身が TCP の場合の RTT

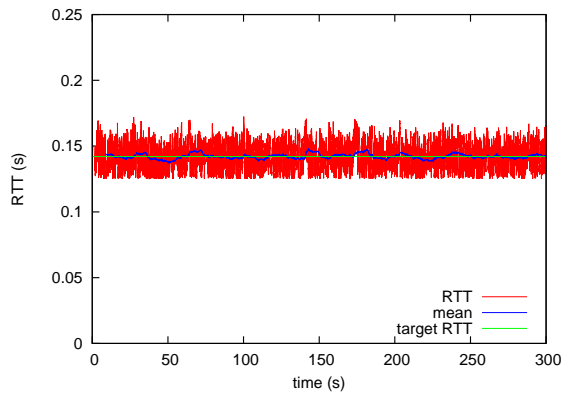


(b) 提案手法における RTT

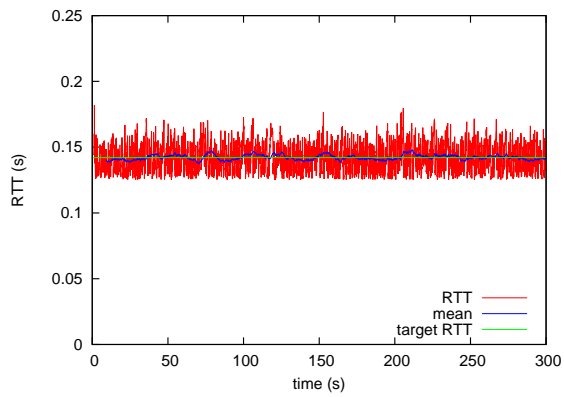


(c) 提案手法における送信レートの変化

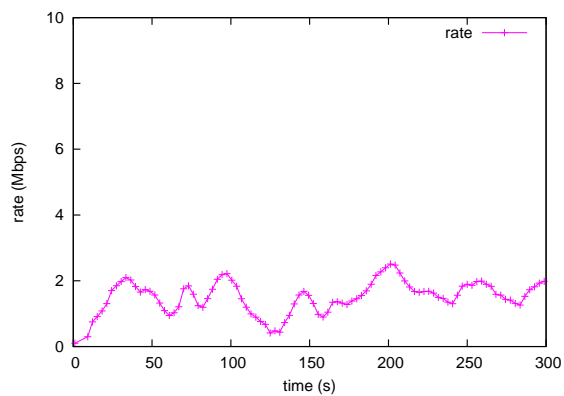
図 10: RTT と送信レートの時間変動 (TCP セッション数 20, 目標 RTT = TCP 平均 RTT, シミュレーション番号 25)



(a) 自身が TCP の場合の RTT



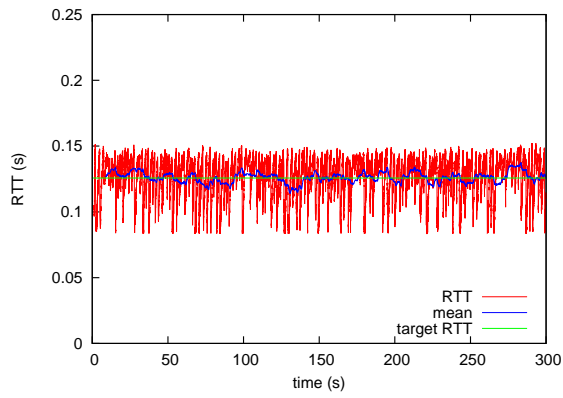
(b) 提案手法における RTT



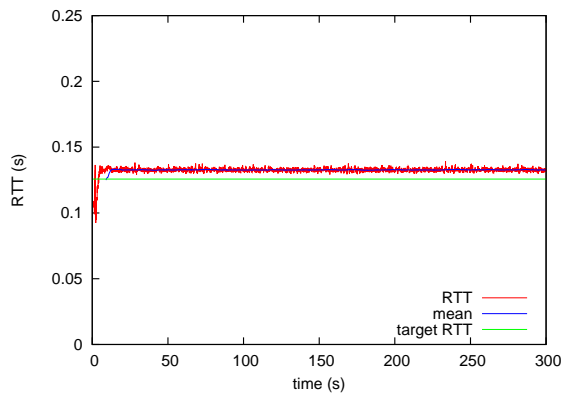
(c) 提案手法における送信レートの変化

図 11: RTT と送信レートの時間変動 (TCP セッション数 30 , 目標 RTT = TCP 平均 RTT , シミュレーション番号 33)

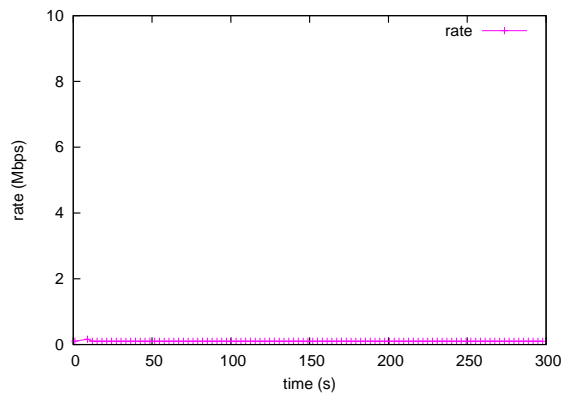




(a) 自身が TCP の場合の RTT

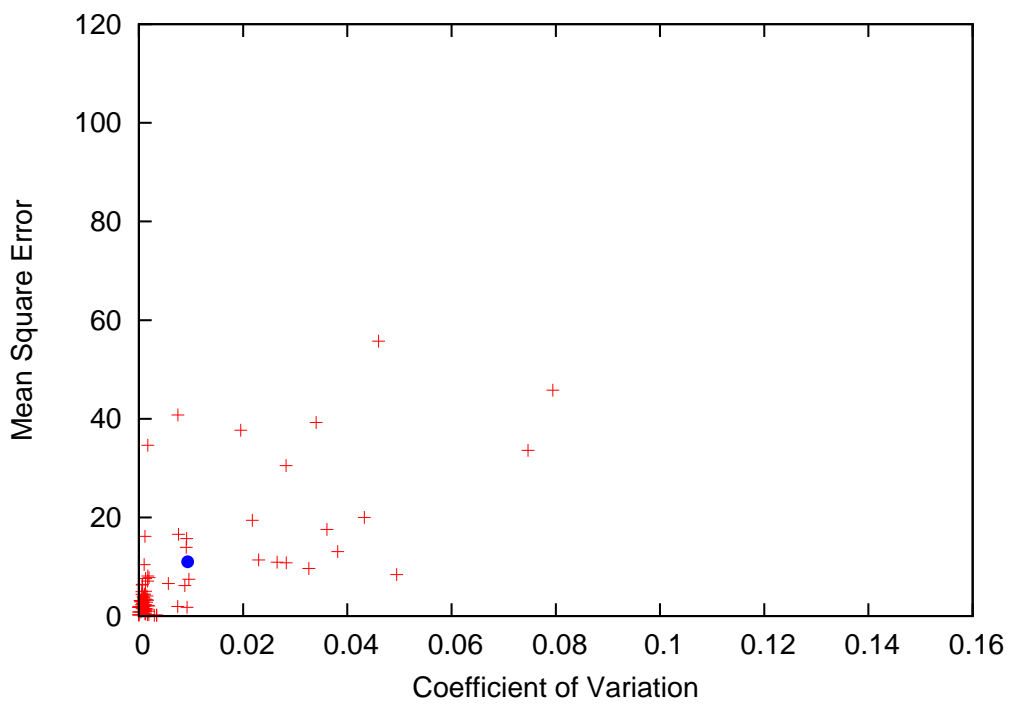


(b) 提案手法における RTT

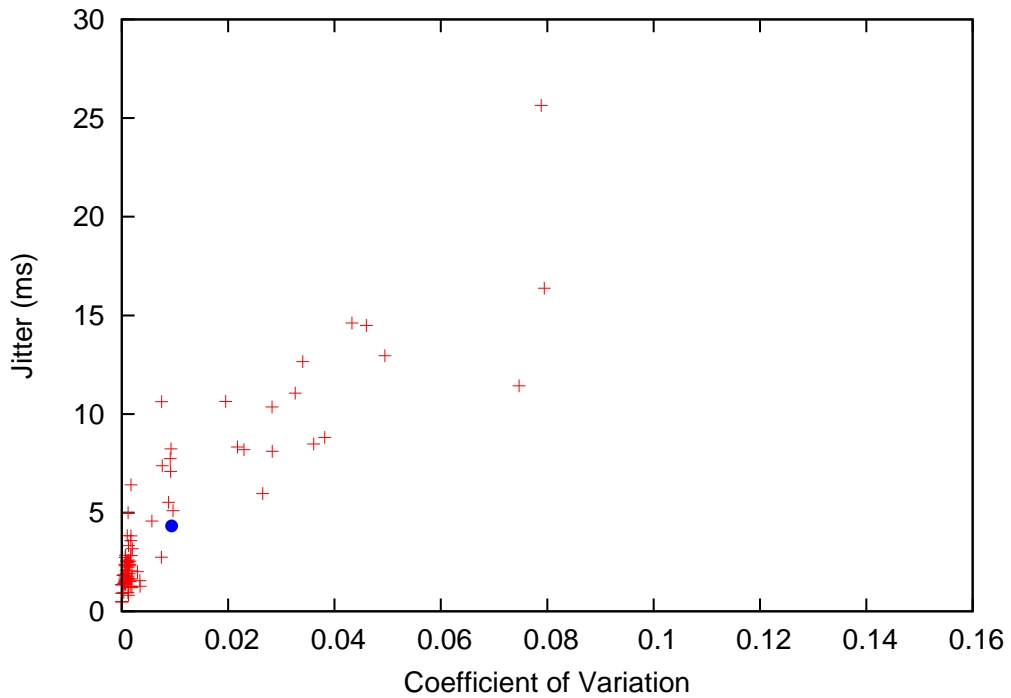


(c) 提案手法における送信レートの変化

図 12: RTT と送信レートの時間変動 (TCP セッション数 40 , 目標 RTT = TCP 平均 RTT , シミュレーション番号 87)

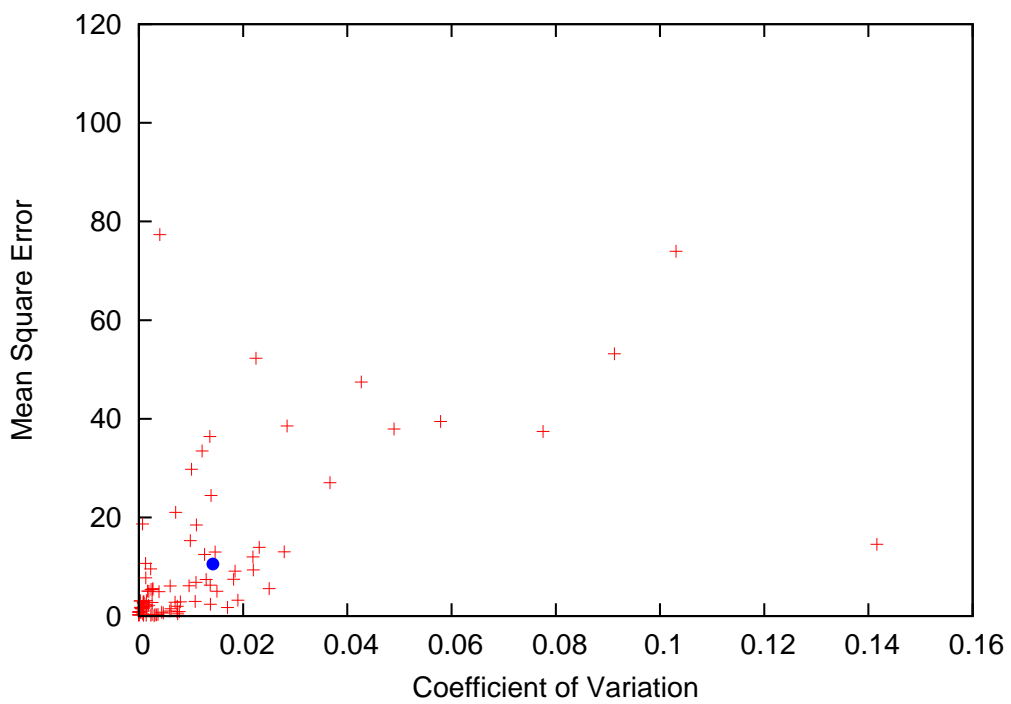


(a) 変動係数  $C$  と平均二乗誤差  $M$  の関係

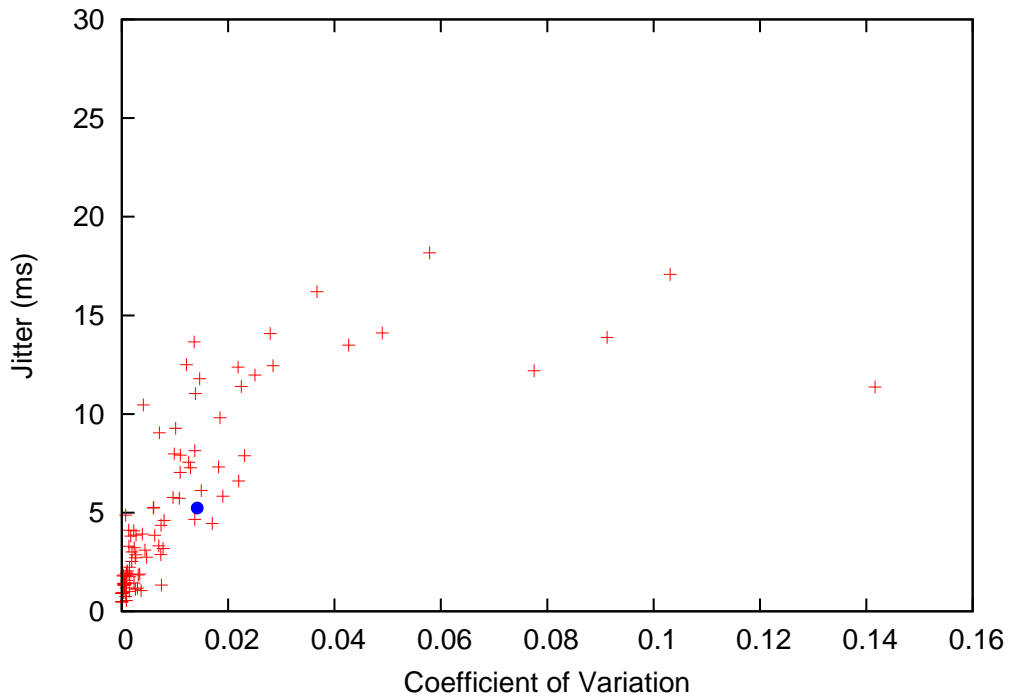


(b) 変動係数  $C$  とジッタ  $J$  の関係

図 13: 平均二乗誤差, 変動係数とジッタの評価結果 (TCP セッション数 20, 目標 RTT = TCP 平均 RTT の 98%)

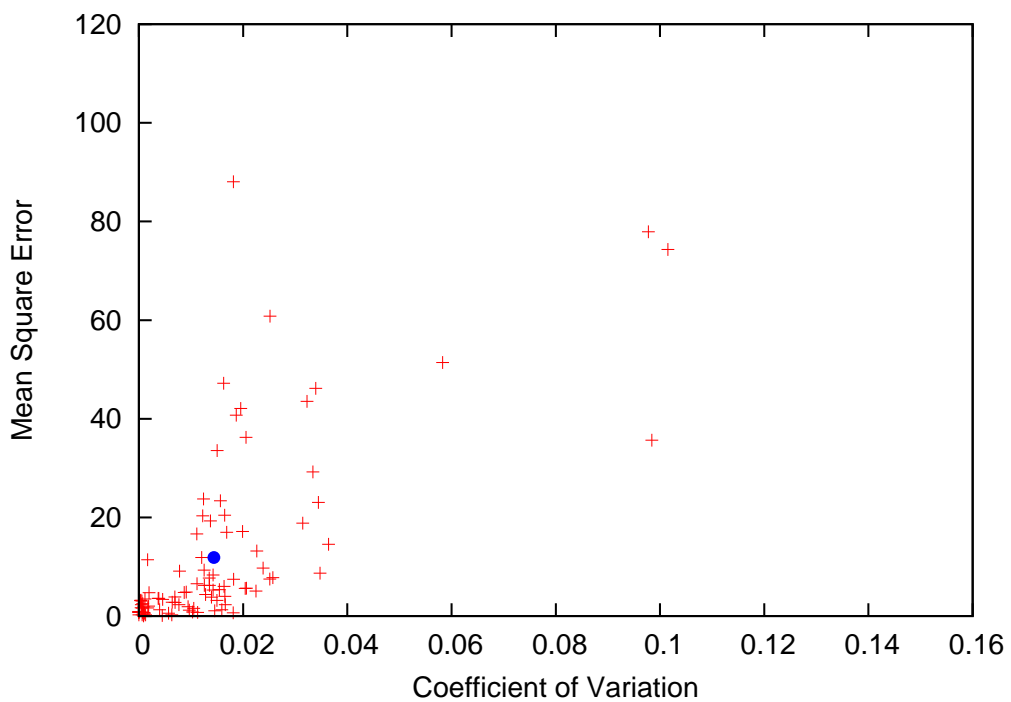


(a) 変動係数  $C$  と平均二乗誤差  $M$  の関係

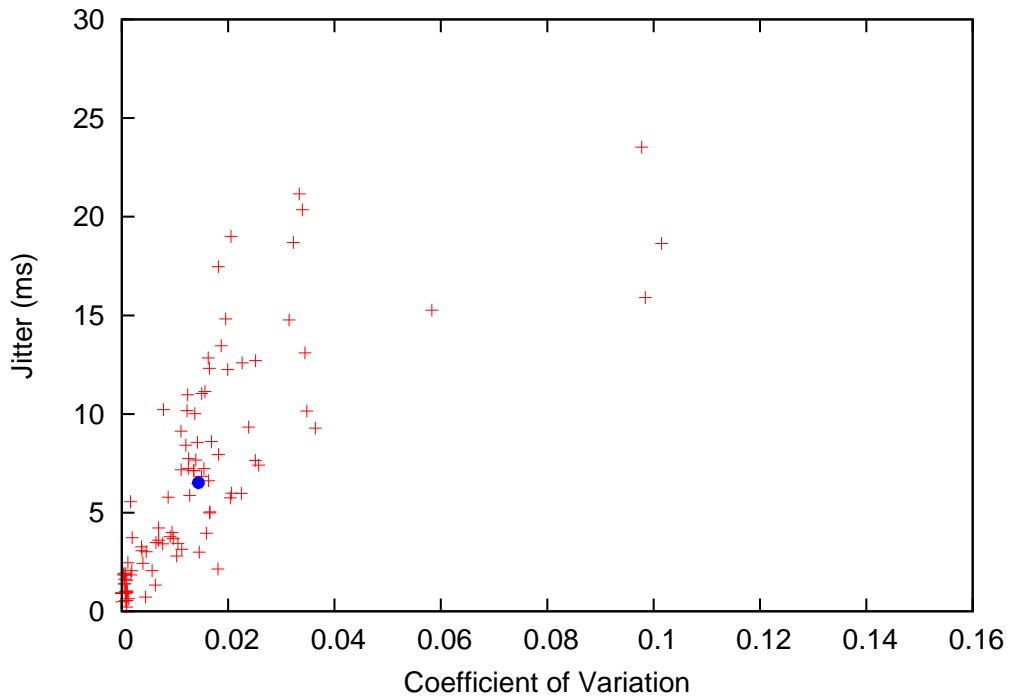


(b) 変動係数  $C$  とジッタ  $J$  の関係

図 14: 平均二乗誤差, 変動係数とジッタの評価結果 (TCP セッション数 30, 目標 RTT = TCP 平均 RTT の 98%)

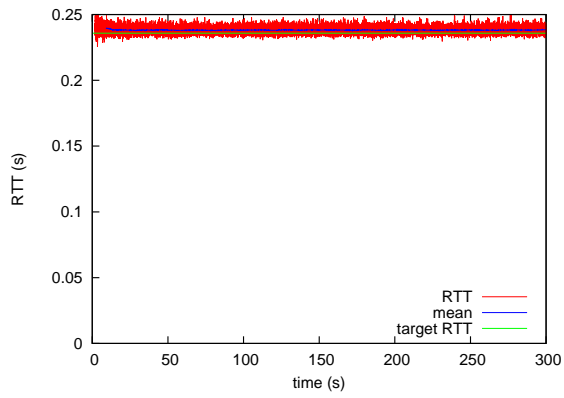


(a) 変動係数  $C$  と平均二乗誤差  $M$  の関係

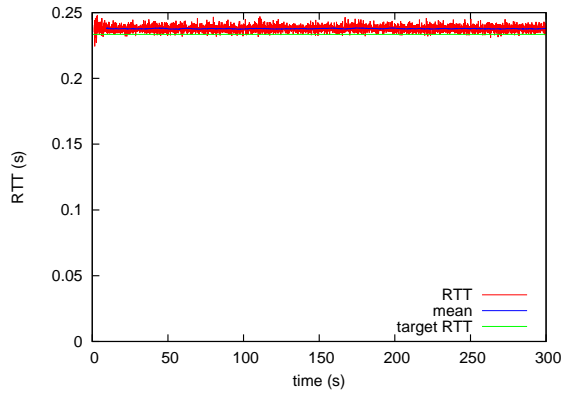


(b) 変動係数  $C$  とジッタ  $J$  の関係

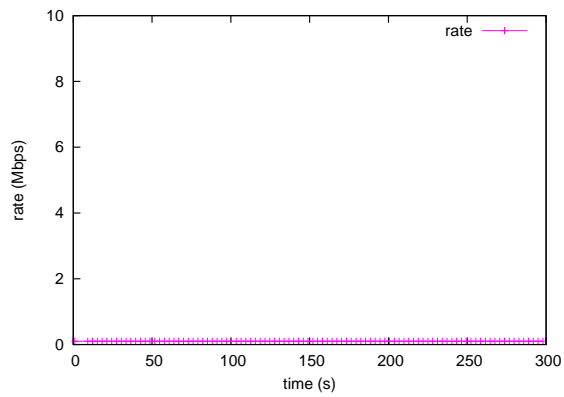
図 15: 平均二乗誤差, 変動係数とジッタの評価結果 (TCP セッション数 40, 目標 RTT = TCP 平均 RTT の 98%)



(a) 自身が TCP の場合の RTT



(b) 提案手法における RTT



(c) 提案手法における送信レートの変化

図 16: RTT と送信レートの時間変動 (TCP セッション数 30, 目標 RTT = TCP 平均 RTT の 98%, シミュレーション番号 1)

### 3.4.2 バックグラウンドの TCP セッション数が変化する場合の評価

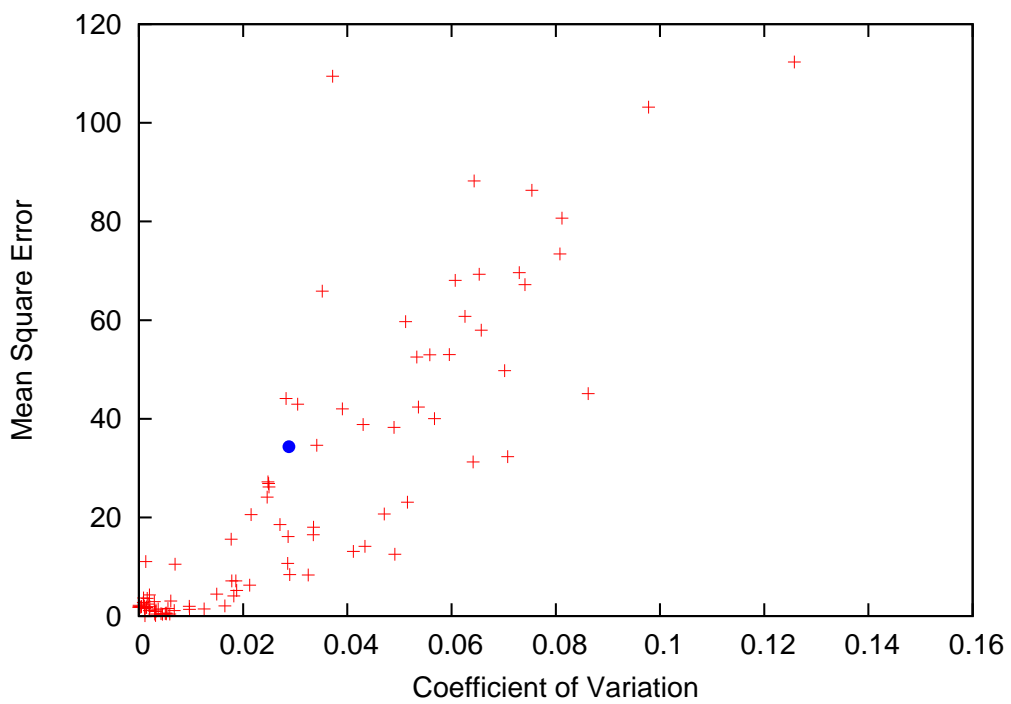
提案手法を用いることで、バックグラウンドトラフィックの変動があった場合にも、目標 RTT を維持できることを示すために、バックグラウンドの TCP セッション数のある時刻において増加させるシミュレーション評価を行った。バックグラウンドトラフィックの TCP セッション数を 20 本から 25 本に変化させる場合と、25 本から 30 本に変化させる場合のそれぞれについて、そのトポロジをランダムに 10 通り生成し、トポロジごとにランダムに 10 通りのセッションの組み合わせを設定することにより、100 回ずつのシミュレーションを行った。シミュレーション時間は 600 s とし、シミュレーション開始後、1 s までにバックグラウンドの TCP セッションがランダムに通信を開始し、1 s に提案手法を用いたセッションが通信を開始する。さらに、シミュレーション時間 300 s に新たに 5 本の TCP セッションが通信を開始する。目標 RTT としては、提案手法の代わりに TCP を用いた場合の、時刻 320 s から 600 s の平均 RTT の 98% に設定した。

図 17 に TCP セッション数を 20 本から 25 本に変更した場合、図 18 に TCP セッション数を 25 本から 30 本に変更した場合、それぞれにおける変動係数  $C$  と平均二乗誤差  $M$  の関係、および変動係数  $C$  とジッタ  $J$  の関係をそれぞれ示す。それぞれの図において丸印は平均値を表す。ただし、いずれも 100 回のシミュレーション評価のうち 1 回の外れ値を除いた 99 回分の評価結果である。TCP セッション数が固定の、図 13、図 14、図 15 の場合と比較して、平均二乗誤差、変動係数、ジッタがいずれも大きくなっている。これは、バックグラウンドの TCP セッションの本数を増やす前の期間における平均 RTT の変動によるものである。本シミュレーション評価では、目標 RTT を TCP セッションの本数を増やした後の TCP の平均 RTT にもとづいて設定しているため、TCP セッションの本数が増える前は自身が TCP である場合の平均 RTT よりも目標 RTT が大きい値になっている。図 9 に示されるように、目標 RTT を TCP の平均 RTT に設定した場合には、RTT の変化を抑えることができているが、これは、目標 RTT が往復伝搬遅延に近く、バッファにパケットが蓄積され始めるとすぐに送信レートを下げたためである。一方、目標 RTT を TCP の平均 RTT よりも大きく設定した場合には、送信側端末は、ある程度バッファにパケットが蓄積されて平均 RTT が目標 RTT に近づくまで、送信レートを上げ続ける。待ち行列理論によると、バッファリング遅延はリンク容量と送信レートの差の逆数で与えられるため、送信レートがある程度以上高い領域では、微少な送信レートの増減が RTT の大きな変化となって現れる。そのため、目標 RTT を達成するための送信レートの増減が過制御に陥りやすく、平均 RTT が大きく変動してしまう。

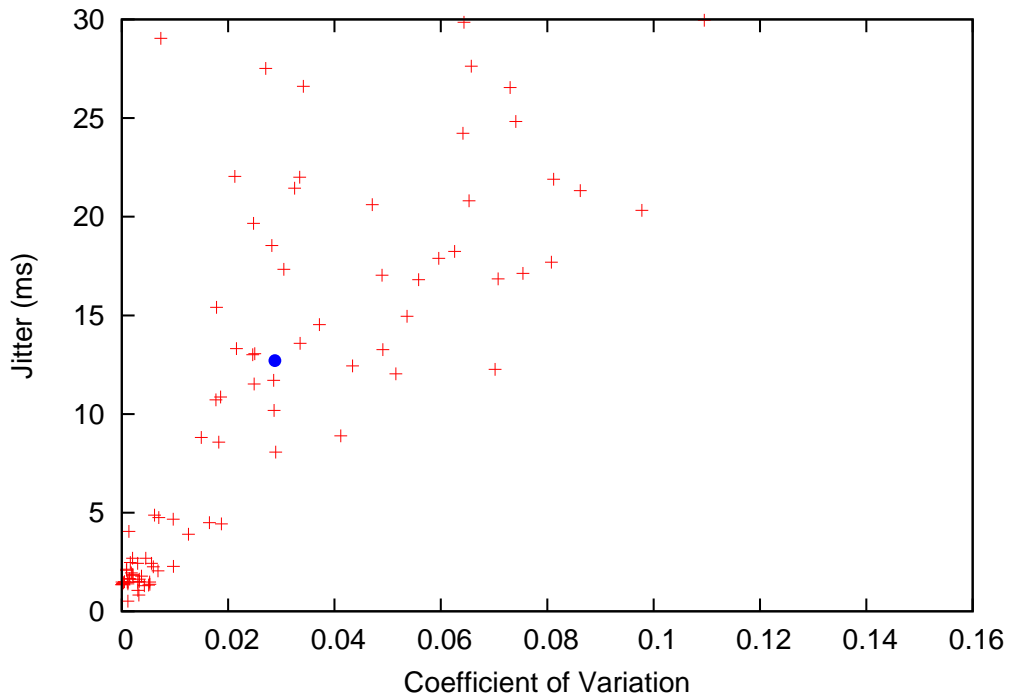
図 19 に、バックグラウンドの TCP セッション数を 20 本から 25 本に変化させた場合の、あるシミュレーション結果を示す。このシミュレーションでは、目標 RTT が高く設定されて

いる 300 s 以前に平均 RTT の変動が見られる。しかしながら、平均 RTT は目標 RTT の近傍を変動しており、また、TCP セッション数が増加した後は、より小さな変動で目標 RTT を達成できている。図 19 におけるシミュレーション時間を通した平均二乗誤差は 32.312、変動係数は 0.0707、ジッタは 16.852 ms であった。同様に TCP セッション数を 20 本から 25 本に変化させた別のシミュレーション結果を図 20 に示すが、この例では TCP セッション数の増加後の平均 RTT の変動はないものの、送信レートが最低レートである 100 kbps に達している。そのため、平均二乗誤差 34.621、変動係数 0.0341、ジッタ 26.61 ms と、図 19 と比較して変動係数は小さいが、平均二乗誤差とジッタが大きくなっている。さらに、図 21 には、TCP セッション数の増加後も平均 RTT が安定しない結果を示している。TCP セッション数変更前後で目標 RTT をおおよそ達成しているが、平均二乗誤差 67.182、変動係数 0.0740、ジッタ 24.82 ms であり、変動が大きい。さらに TCP セッション数を 25 本から 30 本に変化させた場合の結果を図 22、図 23 に示すが、同様に平均 RTT の変動は大きい。なお、図 23 において TCP セッション数が増えることによって提案手法の送信レートが増加している。これは、TCP セッション数の増加により、提案手法のセッションの経路には含まれないリンクが新たにボトルネックとなり、提案手法とリンクを共有する TCP セッションの送信レートが低下したためである。

図 17、図 18 において、平均二乗誤差が 20 以下、変動係数が 0.03 以下、ジッタが 15 ms 以下となるのは、それぞれ 52%、45% であり、TCP セッション数を固定した場合よりも大きく低下している。なお、平均二乗誤差 40 以下、変動係数 0.06 以下、ジッタ 25 ms 以下と条件を緩和した場合には、それぞれ 67%、63% となる。図 17 や図 18 には、平均二乗誤差、変動係数、ジッタが小さい点が含まれているが、これは、TCP セッション数の増加前後で、提案手法のセッションとリンクを共有するバックグラウンドトラヒックの変化がほとんど現れない場合に得られた結果である。



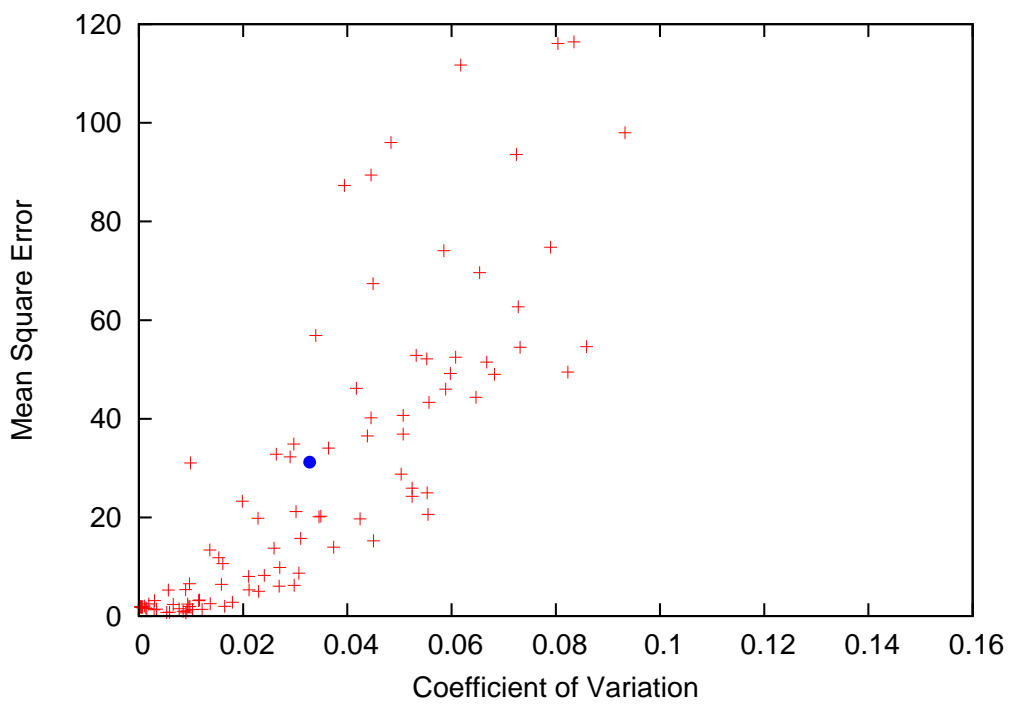
(a) 変動係数  $C$  と平均二乗誤差  $M$  の関係



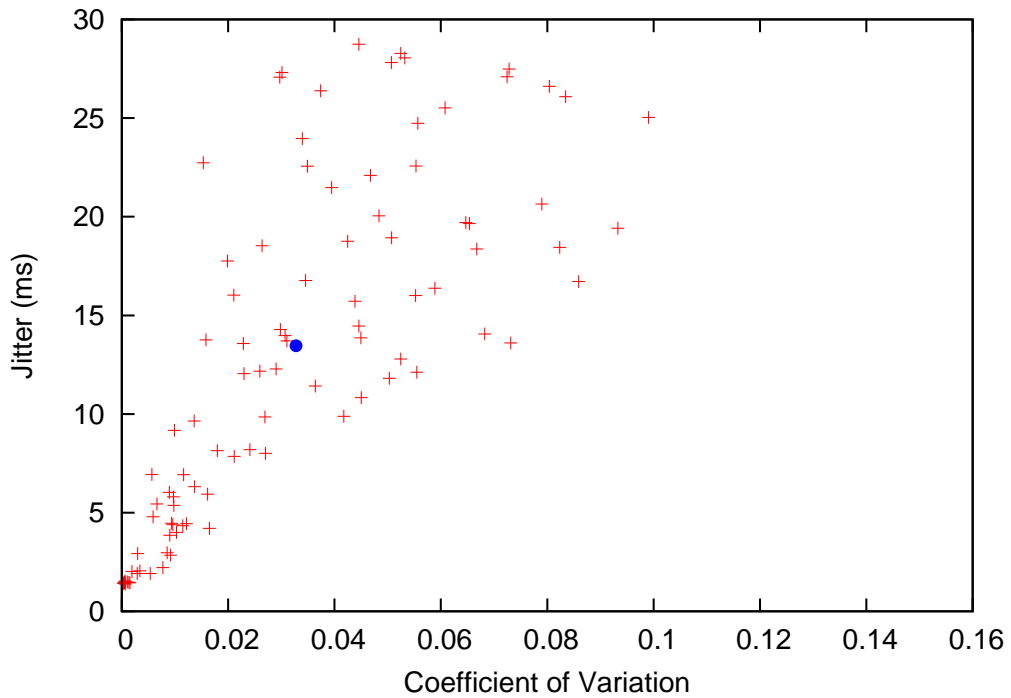
(b) 変動係数  $C$  とジッタ  $J$  の関係

図 17: 平均二乗誤差, 変動係数とジッタの評価結果 (TCP セッション数 20 から 25 に変化)



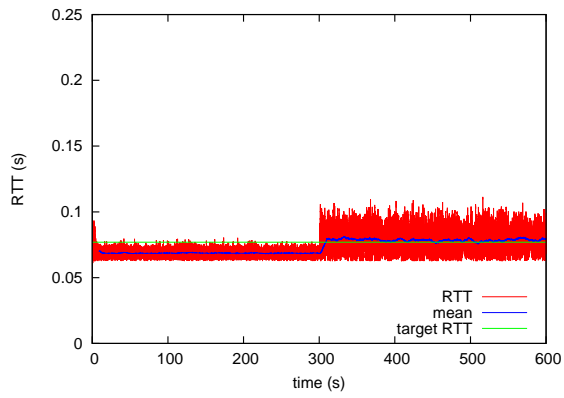


(a) 変動係数  $C$  平均二乗誤差  $M$

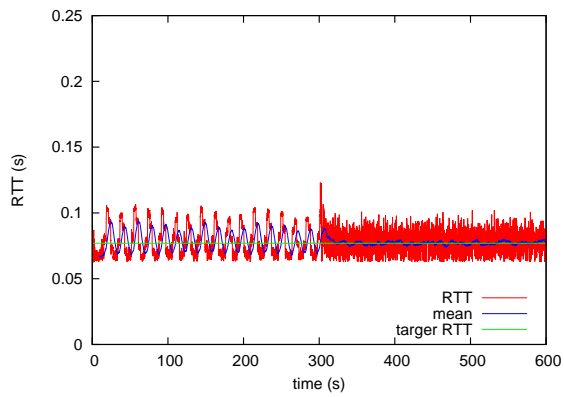


(b) 変動係数  $C$  とジッタ  $J$  の関係

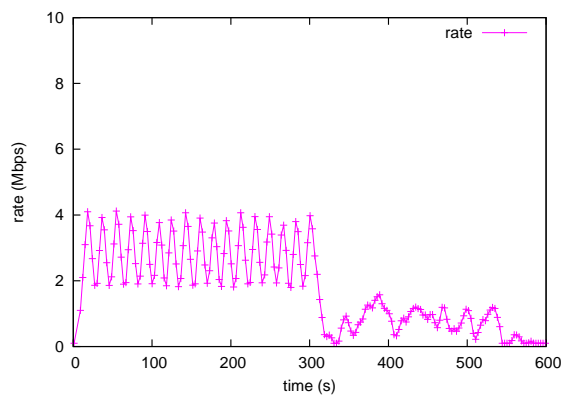
図 18: 平均二乗誤差, 変動係数とジッタの評価結果 (TCP セッション数 25 から 30 に変化)



(a) 自身が TCP の場合の RTT

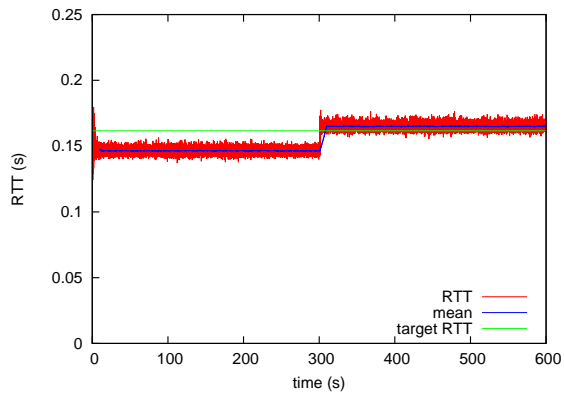


(b) 提案手法における RTT

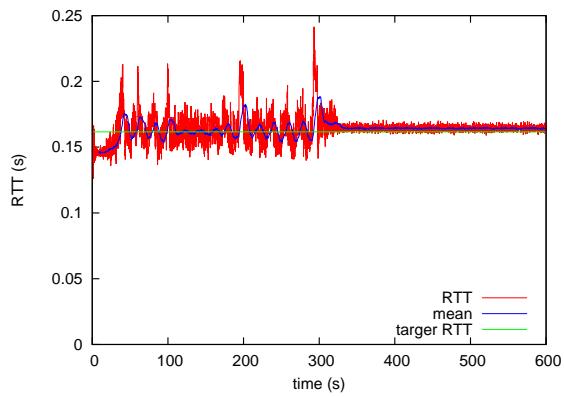


(c) 提案手法における送信レートの変化

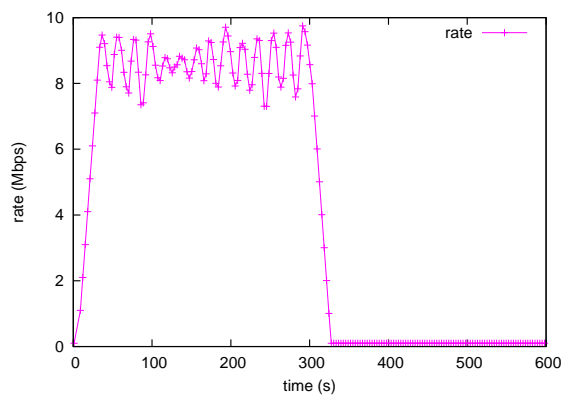
図 19: RTT と送信レートの時間変動 (TCP セッション数 20 から 25, シミュレーション番号 19)



(a) 自身が TCP の場合の RTT

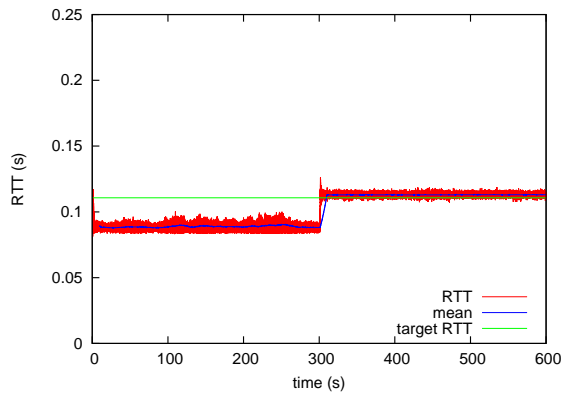


(b) 提案手法における RTT

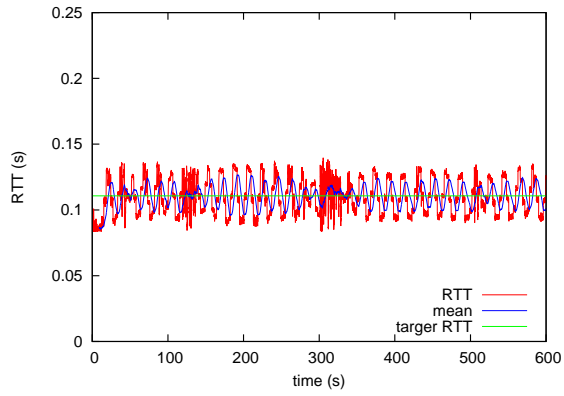


(c) 提案手法における送信レートの変化

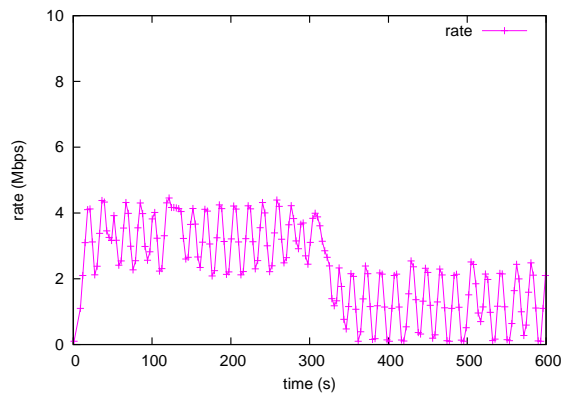
図 20: RTT と送信レートの時間変動 (TCP セッション数 20 から 25, シミュレーション番号 21)



(a) 自身が TCP の場合の RTT

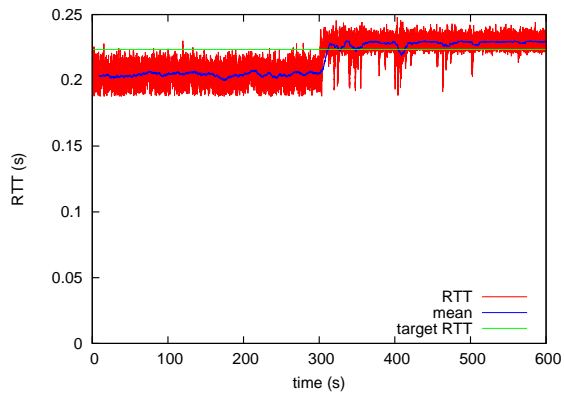


(b) 提案手法における RTT

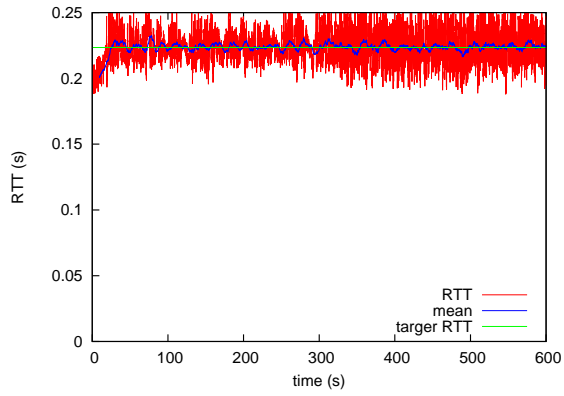


(c) 提案手法における送信レートの変化

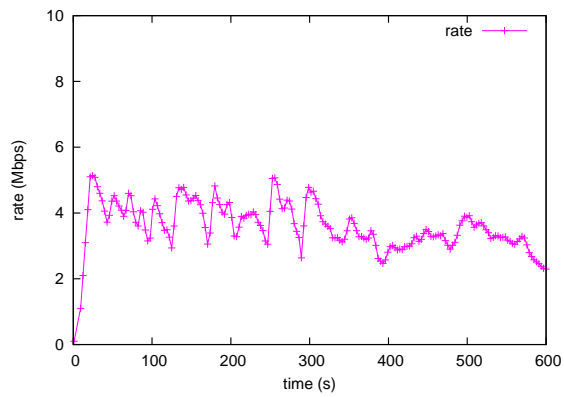
図 21: RTT と送信レートの時間変動 (TCP セッション数 20 から 25 , シミュレーション番号 58)



(a) 自身が TCP の場合の RTT

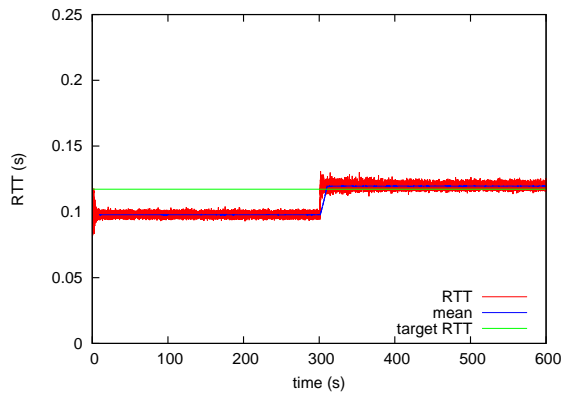


(b) 提案手法における RTT

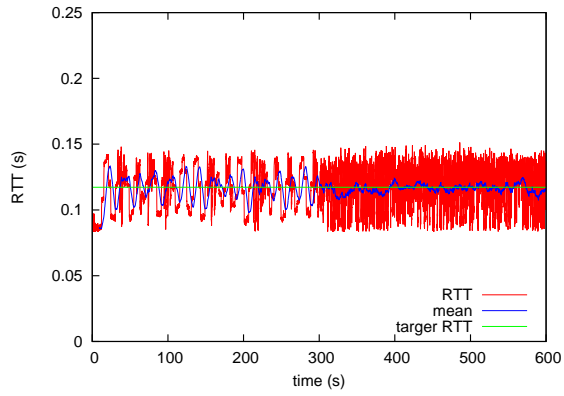


(c) 提案手法における送信レートの変化

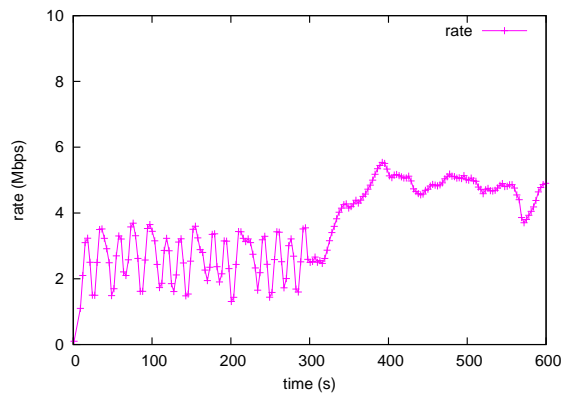
図 22: RTT と送信レートの時間変動 (TCP セッション数 25 から 30 , シミュレーション番号 37)



(a) 自身が TCP の場合の RTT



(b) 提案手法における RTT



(c) 提案手法における送信レートの変化

図 23: RTT と送信レートの時間変動 (TCP セッション数 25 から 30 , シミュレーション番号 8)

## 4 おわりに

本報告では，リアルタイムマルチメディアアプリケーションを対象とした，ネットワークの負荷変動に対して所望のエンド間遅延を達成，維持することのできる自律的な送信レート制御手法を提案を行い，シミュレーション評価を行った．ネットワーク内部の構造や動作原理の情報を必要とせず，エンド間で観測される遅延のみにもとづいて，送信レートの制御を行い，所望のエンド間遅延を達成，維持する手法を提案した．

シミュレーションによる評価の結果，バックグラウンドトラフィックの TCP セッション数が固定のネットワークにおいては，自身が TCP の場合と同程度のエンド間遅延を達成，維持できることが示された．また，TCP セッション数が変化する場合には，ある程度 TCP セッション数の変化前後で目標 RTT に近い平均 RTT を維持できているものの，TCP の平均 RTT よりも大きい値を目標 RTT として設定されている期間において平均 RTT の変動が大きいことがわかった．先に述べたように，提案手法による目標 RTT の達成度合いや安定性は，リンクを共有する TCP セッションの本数や，それらとの相互作用に大きく依存しているため，その影響を緩和するために，今後はノード数やセッション数を増やしたより大規模なネットワークを対象に，提案手法の有効性を評価したい．また，送信レートの更新間隔や，RTT の観測範囲，定数  $b$  などの制御パラメータを変化させ，提案手法のパラメータ依存性や，パラメータのネットワーク環境への依存性についても検証したい．

## 謝辞

謝辞本報告を終えるにあたりまして、日頃より熱心にご指導、ご教授を頂きました大阪大学大学院情報科学研究科の村田正幸教授に心より感謝申し上げます。ならびに、本報告の作成に終始ご指導頂きました大阪大学大学院情報科学研究科の若宮直紀准教授に深く感謝致します。また、大阪大学大学院情報科学研究科の荒川伸一助教、ならびに大阪大学大学院経済学研究科の大下裕一助教には平素から適切な助言を頂き、指導して頂きました。心より感謝致します。最後に、日頃から様々な相談に応じてくださった梶岡慎輔氏、Narun Asvarujanon 氏、山本宏氏、岩井卓也氏をはじめとする村田研究室の皆様方に厚く御礼申し上げます。



## 参考文献

- [1] J. Postel, “RFC793: TCP: Transmission Control Protocol,” Sept. 1981.
- [2] J. Postel, “RFC768: UDP: User Datagram Protocol,” Aug. 1980.
- [3] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, “RFC3550: RTP: A Transport Protocol for Real-Time Applications,” July 2003.
- [4] ITU-T Rec. H.323, “Packet-based multimedia communications systems,” Nov. 1996.
- [5] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, “RFC3261: SIP: Session Initiation Protocol,” June 2002.
- [6] I. Busse, B. Deffner, and H. Schulzrinne, “Dynamic QoS control of multimedia applications based on RTP,” *Computer Communications*, vol. 19, pp. 49–58, Jan. 1996.
- [7] S. Kadur, F. Golshani, and B. Millard, “Delay-jitter control in multimedia applications,” *Multimedia systems*, vol. 4, pp. 30–39, Feb. 1996.
- [8] D. Verma, H. Zhang, and D. Ferrari, “Delay jitter control for real-time communication in a packet switching network,” in *Proceedings of IEEE Triangle Conference on Communications Software*, pp. 35–43, Apr. 1991.
- [9] Y. Mansour and B. Patt-Shamir, “Jitter control in QoS networks,” *IEEE/ACM Transactions on Networking*, vol. 9, pp. 492–502, Aug. 2001.
- [10] J. Pedrasa and C. Festin, “Value-based utility for jitter management,” in *Proceedings of IEEE Region 10 Conference*, pp. 1–5, Nov. 2006.
- [11] D. Hay and G. Scalosub, “Jitter regulation for multiple streams,” *ACM Transactions on Algorithms*, vol. 6, pp. 1–19, Dec. 2009.
- [12] W. He, K. Nahrstedt, and X. Liu, “End-to-end delay control of multimedia applications over multihop wireless links,” *ACM Transactions on Multimedia Computing, Communications and Applications*, vol. 5, pp. 1–20, Nov. 2008.

- [13] W. He and K. Nahrstedt, “Impact of upper layer adaptation on end-to-end delay management in wireless ad hoc networks,” in *Proceedings of 12 th IEEE Real-Time and Embedded Technology and Applications Symposiums*, pp. 59–70, Apr. 2006.
- [14] D. Verma, H. Zhang, and D. Ferrari, “Guaranteeing delay jitter bounds in packet switching networks,” in *Proceedings of IEEE Triangle Conference on Communications Software*, vol. 91, pp. 35–46, Apr. 1991.
- [15] N. Wakamiya, K. Leibnitz, and M. Murata, *Autonomic Network Management Principle: From concepts to Applications*, ch. A Self-organizing architecture for scalable, adaptive and robust networking. Elsevier, 2010.
- [16] K. Kaneko, *Life: An introduction to complex systems biology*. Elsevier, 2006.
- [17] N. Wakamiya, L. Kenji, and M. Murata, “Noise-assisted control in information networks,” in *Proceedings of Frontiers in the Convergence of Bioscience and Information Technologies*, pp. 833–838, Oct. 2007.
- [18] 若宮直紀, 村田正幸, “生物ダイナミクスに学ぶ情報ネットワーク制御,” システム制御学会誌特集「ダイナミクスに基づく情報処理の諸相」, vol. 54, pp. 15–21, Jan. 2010.
- [19] K. Sato, Y. Ito, T. Yomo, and K. Kaneko, “On the relation between fluctuation and response in biological systems,” *National Academy of Sciences*, vol. 100, pp. 14086–14090, Nov. 2003.
- [20] ITU-T Rec. Y.1541, “Network performance objectives for IP-based services,” Nov. 1996.
- [21] N. Wakamiya, M. Murata, and H. Miyahara, “On video coding algorithms with application level QoS guarantees,” *Computer Communication Journal*, vol. 23, pp. 12–15, Jan. 2000.
- [22] “The network simulator ns-2.” <http://www.isi.edu/nsnam/ns/>.
- [23] A. L. Barabási and R. Albert, “Emergence of scaling in random networks,” *Science*, vol. 286, pp. 509–512, Oct. 1999.
- [24] L. Li, D. Alderson, W. Willinger, and J. Doyle, “A first-principles approach to understanding the internet’s router-level topology,” *SIGCOMM Computer Communication Review*, vol. 34, Aug. 2004.