

情報収集型サーバにおける複数 TCP セッション間の協調による QoS 制御の評価

An Evaluation of QoS Control over Information Gathering Server by Coordination between Multiple TCP Sessions

坂尻 康隆† 谷川 陽祐† 戸出 英樹†
Yasutaka Sakajiri Yosuke Tanigawa Hideki Tode

概要

現在のインターネットを構成する IP 網では、高負荷時などにおいて個々のユーザの通信品質 (Quality of Service, QoS) は保証されていない。特に、緊急性を有するクライアントやワイヤレスなど劣悪なアクセス環境にあるクライアントに対しては、一定レベルの品質の提供が不可欠である。本稿では、生体センシング情報などのクライアントの情報と同時に収集するサーバに焦点を当て、優先度の高いクライアントに対し高い QoS を確保するという方針の下で、TCP ソケットバッファサイズ及び TCP コネクション数の調整手法を提案する。また、シミュレーションの結果、本提案方式は物理限界を超えない範囲で優先度に応じた通信品質の確保が実現できることを明らかにする。

1. はじめに

現在のインターネットを構成する IP 網では、無線、有線、光ブロードバンドなど多様なアクセス環境を有することに加え、基本的に Best-Effort サービスのみを提供している。そのため、多くの通信が同時に集中する高負荷時などにおいて個々のユーザに十分な通信帯域が配分されずとは限らず、通信品質 (QoS) は保証されていない。特に、緊急性を有するクライアントやワイヤレスなど劣悪なアクセス環境にあるクライアントに対しては、QoS の観点から一定レベルの品質を提供することが不可欠である。

また、現在の通信サービス形態として、以下の 3 形態が挙げられる。

- (1) 1 対 1 の通信(P2P も含む)
- (2) サーバからの情報取得型サービス
- (3) 情報収集型のサービス

特にこれまでは、(1)、(2) が主体であり、(3) は注目が少なかったが、例えば、IP 網上に散在するクライアントの情報 (生体センシング情報) を同時にサーバに収集し、その情報を基に医師が診断を行うような生体センシングシステム(図 1)など今後重要になるサービス形態である。

本研究では、上記の情報収集型のサーバに焦点を当て、FEC(Forward Error Correction)、誤り制御、輻輳制御、フロー制御などをコネクション毎に独立して適用し品質改善を図るアプローチではなく、情報収集型サーバに接続された複数セッション間の協調制御により、物理的制約条件の範囲内で、最低限品質を確保したいクライアントなど特定コネクションのスループット性能改善を目指す。

提案方式では、既存のネットワークへの適用を考え、経路途中のルータ等の改良を要さない、End-to-End(サーバ-

クライアント)による TCP(トランスポート層) レベルでの制御を行っている。具体的には TCP ソケットバッファサイズ調整と、TCP コネクション数調整の 2 つのアプローチを用いている。詳細は後述するが、一般的にソケットバッファサイズを大きくすると、スループットは増加する。そのため、目標スループットやネットワークの状況を考慮に入れた調整を行うことで、優先度の高いクライアントに対して高い QoS を確保することや、劣悪なアクセス環境にあるクライアントに対しても物理限界を超えない範囲で QoS を確保することが可能になるなど、個々のコネクション間で差別化、公平化を行うことができる。また、TCP コネクション数を増加させることによって、スループットの増加が期待される。ネットワークが逼迫している環境、あるいは、狭帯域なワイヤレス環境などにおいてコネクション数調整を行うことで、同様に差別化、公平化を図ることができる。

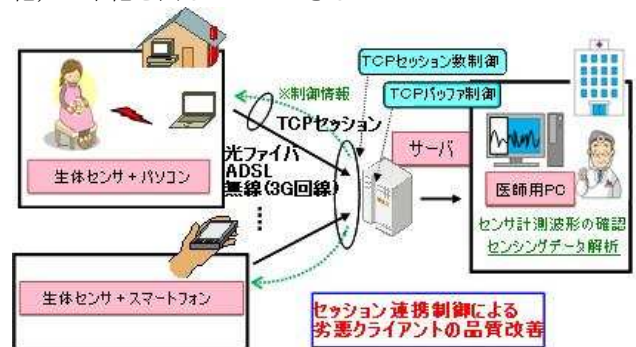


図 1: 対象システムの例: 生体センシングシステム

2. 関連研究

大容量のデータを効率的に転送するためのプロトコルとして、グリッドコンピューティングにおける GridFTP[1] が提案されている。GridFTP は、既存の TCP の問題点を解消することを目的として設計されており、様々な機能が追加されている。具体的には、既存の FTP に対して、TCP ソケットバッファサイズの自動交渉、並列データ転送、データ転送の第三者制御、部分ファイル転送、セキュリティ、信頼性のあるデータ転送といった機能が挙げられる。しかし、GridFTP において、ソケットバッファサイズ及び並列 TCP コネクション数をどのように決定するかについて、これまで十分な検討は行われていない。

バッファサイズ調整に関しては様々な研究が行われているが[2,3]、既存の研究では、Scalable Automatic Buffer Tuning [4]のように Web サーバなどの情報配信型サーバを対象にすることに加え、スループットに応じた調整のみ

を行っており、各接続の優先度等は考慮されていない。また、TCP 接続数や目標スループット調整に関しては GridFTP を利用した GridFTP-APT[5]があるが、これは、目標スループットの達成を考慮に入れない、使用帯域内でスループットを最大化するものである。

既存方式では、基本的に個別の接続において、帯域を最適に使用するための方式が規定されている。一方、本提案方式は情報収集型サーバを対象とし、各接続毎に設定された優先度に応じて目標スループットを定め、それに従ってソケットバッファサイズ及び並列 TCP 接続数調整を行っている。

3. 提案方式

本提案方式は、優先度の高いクライアントに対して目標値に応じた QoS を確保することを基本方針とし、各接続に割り当てる TCP ソケットバッファサイズ及び並列 TCP 接続数を決定する。最初に、予め優先度に応じて決められた目標スループットを基に、各接続に必要なバッファサイズを算出する。そして、全ての接続に、まず最低限必要なバッファサイズを割り当て、余ったリソースを優先度に応じて振り分けるという手順で、最終的な各接続のバッファサイズを決定する。また、後述するアルゴリズムに従って、目標スループットを満たすような TCP 接続数を決定する。

3.1 ソケットバッファサイズ調整

TCP では、送受信側が互いの受信バッファサイズを把握し合い、その範囲内でネットワークの状況に応じてパケットをまとめて送信することにより効率を上げている。そのため、バッファサイズが小さいと伝送レートが低下する要因になる。逆に、バッファサイズが大きすぎる場合には、損失パケットの再送の際に、再送されるデータ単位が大きくなる。そのため、再送パケットが再び損失した場合、逆に効率が悪くなる。またサーバのバッファ資源の浪費につながる。従って、ソケットバッファサイズは以上のことを考慮に入れた上で、最適な値に設定される必要がある。

一般的に最適なソケットバッファサイズは、往復ネットワーク経路上に滞在できるパケット量である「帯域遅延積」により決定される。利用可能な帯域を最適に使用するためには、少なくともこの帯域遅延積分のバッファサイズが必要となる。

本提案方式では、目標スループットを満たすために、ソケットバッファサイズ W を目標スループット T_t 、ラウンドトリップタイム RTT を用いて以下の式で決定する。

$$W[KB]=T_t[Mbps]*RTT[ms]/8 \quad (1)$$

(1)式で算出されるバッファサイズを各接続に割り当てる。もしバッファに余剰が発生する場合は、目標スループットの比に応じて、余剰分をさらに振り分ける。一方、割り当ての際にバッファが不足する場合は、全セッションの目標スループットを最低優先度の目標スループットに設定し、改めてバッファサイズを決定する。その後もしバッファに余剰があれば、高優先度のものから順に振り分ける。なお、最低優先度の目標スループットを全接続が確保できない場合は、当該目標ス

ループット比で配分する。

3.2 TCP 接続数調整

複数の TCP 接続を集約することにより、単一の TCP 接続を用いる場合に比べて、より高いスループットが期待できる。これは、以下の理由によって説明できる。

- TCP の輻輳回避フェーズにおいて、競合する他の TCP 接続よりも、より大きな帯域を利用できる。これは、TCP の輻輳回避フェーズでは、Additive Increase Multiplicative Decrease(AIMD)型のウィンドウフロー制御が行われており、パケット棄却率が小さいネットワークでは、複数の TCP 接続を集約することによりウィンドウサイズの増加速度が向上し、効率的なデータ転送を行えるからである。

- TCP のスロースタートフェーズにおける転送レートの立ち上がりが速くなる。

スロースタートフェーズでは、1 ラウンドトリップタイムごとに、輻輳ウィンドウが2倍になる。このため、N本の TCP 接続を集約することにより、転送レートの立ち上がりがN倍になる。

しかし、集約する TCP 接続数 N が大きすぎると、以下のような理由によりスループットが低下する。

- TCP 接続あたりのウィンドウサイズが小さくなり、タイムアウトが頻繁に発生してしまう。

- サーバにおいて、TCP プロトコルスタックの処理に要するオーバーヘッドが大きくなる。

このため、ネットワークの状況に応じて、集約する TCP 接続数 N の最適値を決定する必要がある。

TCP 接続数調整のアルゴリズムは以下の通りである。

並列 TCP 接続数の初期値を N_0 、加算増加量を α 、乗算減少量を $\beta(0<\beta<1)$ とする。

(I) スループットを計測し、目標スループットを満たしていれば終了。

(II) 接続数を N_0 で初期化。

(III) 並列 TCP 接続数を α だけ増加させる。

(IV) スループットを計測し、目標スループットを満たしていれば終了。

一方、満たしていない場合、スループットが増加していれば、(VI)へ、減少していれば(V)へ処理が進む。

また $\alpha=1$ のとき、これまで増加していたスループットが減少に転じていれば並列 TCP 接続数を N_b として終了。

(V) 現在の並列 TCP 接続数に β を掛ける。そして (VI) へ処理が進む。

(VI) α 、 β を(2)(3)式で更新。(III)へ戻る。

$$\alpha \leftarrow \max(1, T_t/T_n * RTT * 10) \quad (2)$$

$$\beta \leftarrow \max(N_b/N_n, T_n/T_t) \quad (3)$$

このようなアルゴリズムにおいて、 T_t 、 T_n 、 N_n 、 N_b はそれぞれ目標スループット、現在のスループット、現在の接続数、1つ前の接続数であり、(2)式中の10はRTTの値を調整するためのパラメータである。

加算増加量 α 、乗算減少量 β の更新式に関しては、できるだけ早く目標スループットあるいは最大スループットに到達するために以下の点に考慮している。

- 現在値が目標値に遠い場合は α を大きく、 β を小さく

する。

・RTT が増加すると最適なコネクション数も増加するため[5], RTT が大きい場合, α を大きくする。

4. 性能評価

本章では, ns-2[6] を用いた計算機シミュレーションにより, 提案方式の有効性を定量的に評価する。

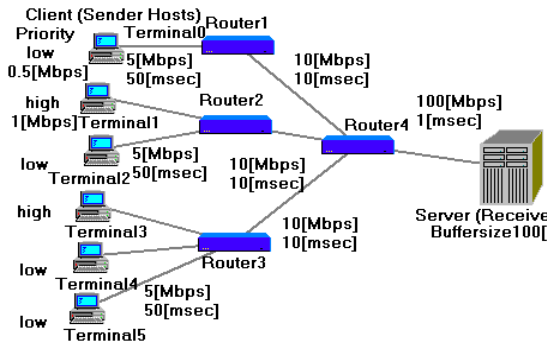


図 2: シミュレーションモデル 1(Model1)

まず, 図 2 に示すモデルでシミュレーションを行った。シミュレーションに用いたパラメータを表 1 に示す。以降では, 特に断りのない限りこのパラメータを使用している。

表 1: シミュレーションで用いたパラメータ

TCP 輻輳制御	TCP Reno
各ルータのバッファサイズ	40[Packet]
高優先度クライアントの目標スループット	1.0[Mbps]
低優先度クライアントの目標スループット	0.5[Mbps]
並列 TCP コネクション数の初期値 N_0	1
サーバの TCP 受信ソケットバッファサイズ	100[KB]

各コネクション毎に均等にソケットバッファを割り当て, TCP コネクション数を 1 とした比較方式(equality)と, 提案方式(proposal)の優先度ごとの平均スループットを比較した結果を図 3 に示す。

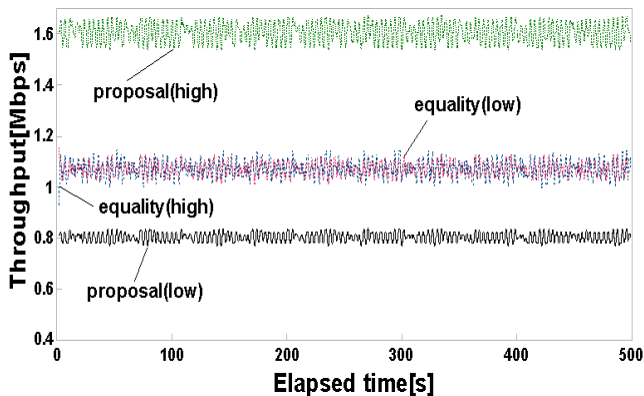


図 3: スループット変化(バッファサイズ 100[KB])(Model1)

図 3 より, 提案方式では比較方式と比べ, 低優先度ク

ライアントのスループットが低くなっているが, 高優先度クライアントに対しては比較的高いスループットが維持されている。つまり提案方式では, 高優先度, 低優先度クライアント共に目標値に対して余裕をもってスループットを満たしているが, 比較方式では高優先度クライアントが状況により目標スループットを下回ることが確認された。

さらに, サーバの受信ソケットバッファサイズを 60[KB]に変更した時の優先度ごとの平均スループットを図 4 に示す。

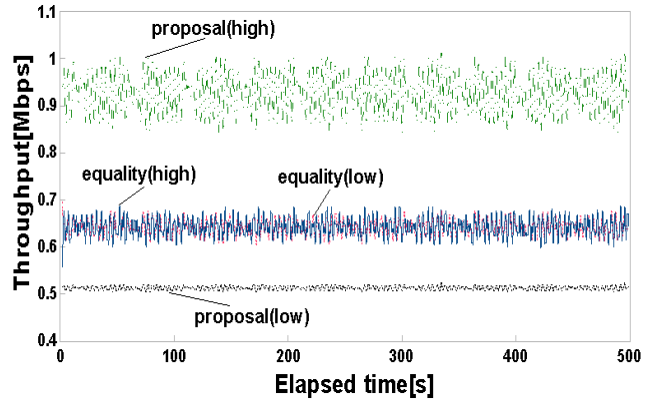


図 4: スループット変化(バッファサイズ 60[KB])(Model1)

図 4 より, 提案方式, 比較方式共に高優先度クライアントの目標スループットを下回っていることがわかる。サーバの受信ソケットバッファサイズが図 3 の場合よりも小さくなっているため, 高優先度クライアントの通信に十分なバッファを割り当てることができずウィンドウサイズが減少し, その結果, 目標スループットを満たすことができなくなったと考えられる。ただし, 提案方式では低優先度の目標スループットである 0.5[Mbps]を確保しつつ, 高優先度のスループットを目標値の 1.0[Mbps]に最大限近づけていることがわかる。

次に図 5 のモデルにおいて, Router3-4 間に干渉トラヒックを発生させ, シミュレーションを行った。なお干渉トラヒックは, interference Terminal0,1,2 のそれぞれから interference Server へ転送され, シミュレーション時間 100-500[s]の間に発生させている。

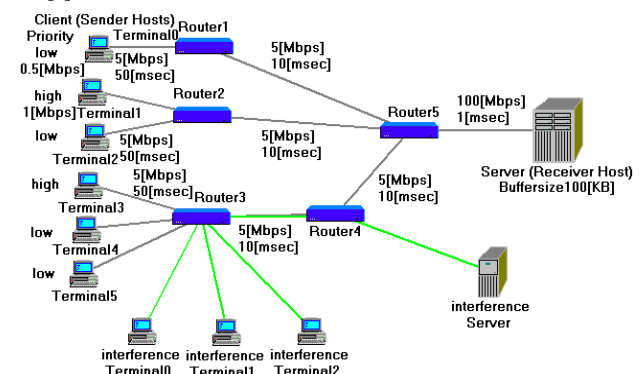


図 5: シミュレーションモデル 2(Model2)

サーバの受信ソケットバッファサイズ調整機能のみを適用, バッファサイズと TCP コネクション数調整機能の両方を適用した際の各端末ごとのスループットの時間的変化をそれぞれ図 6, 図 7 に示す。なお, グラフの簡素化の

ため、ほぼ同じ結果となった Terminal0 と 2 及び Terminal4 と 5 に関しては、それぞれ Terminal0 と Terminal4 の結果のみを記載している。

図 6 より、バッファサイズ調整機能のみの適用では、高優先度クライアント Terminal3 の通信が目標スループットを下回っていることがわかる。一方、TCP コネクション数調整機能を加えた図 7 では、干渉トラフィック追加後も Terminal3 は目標スループットの 1.0[Mbps]を満たしている。これは、干渉トラフィックが発生した場合、多くの通信が同じ回線を共有することで、個々のコネクションの実効帯域幅が小さくなるため、単純にバッファサイズを調整しただけでは目標スループットを満たすことができないためと考えられる。一方、TCP コネクション数調整を行えば、他の通信より多くの帯域を獲得できるため、多少ネットワークが混雑し輻輳が発生しても目標スループットを維持できると考えられる。

以上のシミュレーション結果から、本提案方式は、バッファなどのリソースが不足しているあるいは実効帯域の制約がある場合は目標スループットの達成は厳しいものの、比較的にリソースに余裕がある場合には、個々のコネクションに対して優先度に応じて設定された目標スループットを確保できていることが確認された。

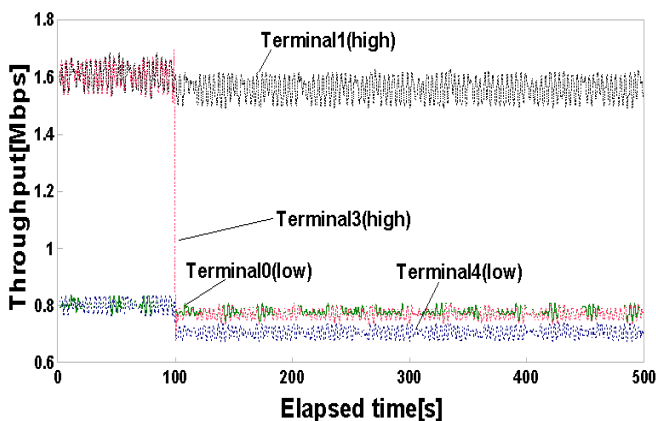


図 6: スループット変化 (バッファサイズ調整のみ)(Model2)

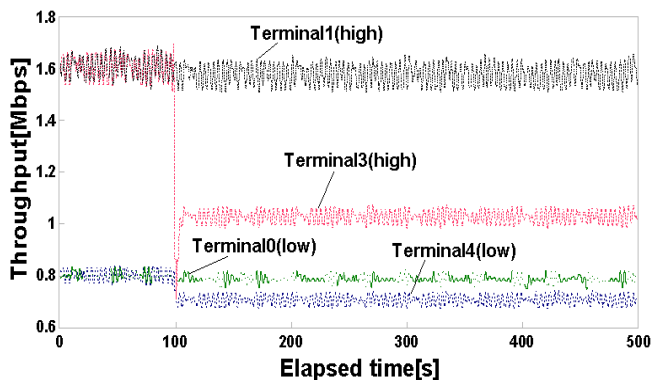


図 7: スループット変化 (バッファサイズ調整 + コネクション数調整)(Model2)

5. まとめ

本研究では、情報収集型サーバを対象とし、優先度の高いクライアントに対し高い QoS を確保することを最優先にしつつも、優先度の低いクライアントに対してもアプリケーションが要求する最低限度の QoS を確保すると

いう方針の下で、現在のネットワークにおいて実現性の高い End-to-End のみの制御である、トランスポート層レベルの制御手法、具体的には TCP ソケットバッファサイズ及び TCP コネクション数の調整手法を提案した。また、シミュレーションの結果、物理リソースが逼迫している場合を除き、本提案方式では優先度に応じた通信品質の確保が実現できることを明らかにした。

参考文献

- [1] I. Mandrichenko, W. Allcock, and T. Perelmutov, "GridFTP v2 protocol description," GGF Document Series GFD.47, May 2005. Also available as <http://www.gridforum.org/GFD.47.pdf>.
- [2] NLAR, "Auto Tuning Enabled FTP Client And Server," "<http://dast.nlanr.net/Projects/Autobuf/>"
- [3] Brian L. Tierney, Dan Gunter, Jason Lee, Martin Stoufer, "Enabling Network-Aware Applications," Proceedings of IEEE HPDC- 10'01, 2001.
- [4] 松尾孝広, 長谷川剛, 村田正幸, 宮原秀夫, "スケラビリティおよび公平性を考慮した TCP バッファ制御" 信学技報, SSE99-167, Mar. 2000.
- [5] T. Ito, H. Ohsaki, and M. Imase, "GridFTP-APT: Automatic Parallelism Tuning Mechanism for Data Transfer Protocol GridFTP", Proceedings of 6th IEEE International Symposium on Cluster Computing and the Grid (CCGrid2006), May 2006.
- [6] The network simulator ns2: "<http://www.isi.edu/nsnam/ns/>"