

介護・福祉施設の ICT 化における輻輳制御の改善

阿部 竜司 (鶴岡工業高等専門学校 生産システム工学専攻, s230004@tsuruoka.kosen-ac.jp)

ザビル サラウッディン ムハマド サリム (福島大学 共生システム理工学類, szabir@sss.fukushima-u.ac.jp)

穴戸 道明 (鶴岡工業高等専門学校 創造工学科, m-shishido@tsuruoka-nct.ac.jp)

Improving congestion control in the implementation of ICT at nursing and welfare facilities

Ryuji Abe (Department of Advanced Engineering, National Institute of Technology, Tsuruoka College, Japan)

Salahuddin Muhammad Salim Zabir (Graduate School of Symbiotic Systems Science and Technology, Fukushima University, Japan)

Michiaki Shishido (Department of Creative Engineering, National Institute of Technology, Tsuruoka College, Japan)

要約

近年の介護・福祉施設では、少子高齢化にともなう生産年齢人口の低下によってもたらされる「介護人材不足」の課題が顕著化している。こうした介護・福祉施設の課題解決のひとつとして ICT 化が挙げられるが、ICT 化による課題も同時に表面化する。それは、輻輳による通信速度の低下である。本研究では、輻輳が引き起こす通信の課題解決のため、ネットワークシミュレーションを行い、自身が実装した輻輳制御アルゴリズムである Reactive TCP の性能評価を行った。実験では、通信量が増加していった際のスループットとパケット損失数を計測した。Reactive TCP は、損失ベースと遅延ベースを状況に応じて使い分ける方式としており、パケット損失を輻輳の指標としている。このアルゴリズムでは、通常は損失ベースで通信を行い、輻輳を検知しない場合には引き続き損失ベースでの通信が行われる。輻輳検知の際、遅延ベースに切り替えて処理を行い、損失ベースによる通信が再開される。実験には3種類のデータを用意し、それぞれデータ①からデータ③とした。データ①はパケットが破棄されたタイミングで切り替え、データ②は切り替えのタイミングを遅く、データ③は切り替えのタイミングを早くしたアルゴリズムとなっている。実験結果より、スループットとパケット損失数の両方でデータ①のアルゴリズムが高い効果を発揮した。しかし、実際のインターネット環境とは異なるため、実際のインターネット環境を用いた実験の検討が必要である。

Abstract

In recent years, nursing and welfare facilities have been facing a growing shortage of nursing care workers due to a decline in the working-age population caused by a falling birthrate and aging population. ICT is regarded as one of the solutions to these issues in nursing and welfare facilities, however, issues that arise from ICT have also surfaced. One such problem is the slowdown of communication speed caused by congestion. In this study, network simulations were conducted, and the performance of Reactive TCP, an implemented congestion control algorithm, was evaluated. In the experiment, throughput and the volume of packet losses were measured as the communication volume increased. Loss-based and delay-based schemes are used by Reactive TCP depending on the situation, and packet loss is used as an indicator of congestion. The algorithm is normally operated on a loss basis, and communication is continued on a loss basis if congestion is not detected. When congestion is detected, a switch to delay-based processing is made, and loss-based communication is resumed. Three sets of data were used in the experiment, labeled data (1) through data (3). The algorithm for data (1) is switched when a packet is discarded, for data (2), the switching timing is delayed, and for data (3), the switching timing is accelerated. It was shown by the experimental results that the algorithm in data (1) was highly effective in both throughput and volume of packets lost. However, since this differs from the actual internet environment, further experiments in the actual internet environment need to be considered.

キーワード

介護・福祉施設, ネットワーク, 輻輳制御, ハイブリッド方式, スループット

1. 緒言

近年の介護・福祉施設では、少子高齢化にともなう生産年齢人口の低下によってもたらされる「介護人材不足」の課題が顕著化している。「第8期介護保険事業計画に基づく介護職員の必要数(厚生労働省, 2023)」と「我が国における生産年齢人口の推移と将来推計(厚生労働省, 2023)」を比較すると、前者は増加しているのに対し、後者は減少しており、それらの推計は対照的な結果である。厚生労働省の推計によると、2019

年度と比較して、2040 年度にはおよそ69 万人もの介護人材が必要とされている。2040 年度以降もその数は増え続けていくことが容易に推測されるため、「介護人材不足」は今後もさらに深刻となり、さらなる介護職員の負担の増加やサービスの質の低下が予想される。

こうした介護・福祉施設の課題解決には、適切な ICT 化によるインフラの高度化が極めて重要である。介護・福祉施設における ICT 化の具体例としては、電子カルテや、請求業務や事務業務をサポートする介護ソフト、コロナ禍で大きな拡大を見せたオンライン診療などがある。こうした技術は介護職員の負担軽減を可能とするため、ICT 化の需要は高まり続けると推測される。一方、ICT 化による課題も顕著化する。

それは、輻輳による通信速度の低下である。一般に、大多数の人が同じネットワーク上で大量のデータ通信を行ったケースが主な原因として挙げられ、輻輳が発生すると、通信速度低下による業務効率の低下といった問題が生じる。また、介護・福祉施設では、膨大な数のセンシティブな個人情報を扱っているため、情報の損失には十分注意しなければいけない側面もある。すなわち、介護・福祉施設のICT化に伴い、日々の通信データ量が増加していることから、早急な対策が必要である。

本研究では、ネットワークシミュレーションにより意図的に輻輳を発生させ、自身が実装した輻輳制御アルゴリズムのスループットとパケット損失数を計測した。そして、アルゴリズムの変更による効果を検証した。

2. TCP プロトコルおよびアルゴリズムの概要

2.1 TCP

現在のインターネットで標準的に用いられている通信プロトコルとして、TCP/IPがある。本研究ではTCP (Transmission Control Protocol) に着目した。TCPとはOSI参照モデルの第4層にあたる、トランスポート層で動作する通信プロトコルであり、接続型である。送受信するアプリケーション間でデータを転送する前に、TCP接続プロセスと呼ばれる接続手順を確立する必要がある。この接続手順の確立は三度のステップで行うため、3ウェイハンドシェイクと呼ばれる。

図1にTCPにおける3ウェイハンドシェイクの動作を示す。ステップ1では、送信元があて先に対して接続要求 (SYN) を送信する。このとき送信元は、TCPヘッダの制御フラグのSYNビットを1にして送信する。ここで、SYNビットがセットされたセグメントをSYNセグメント、または単にSYNと呼ぶ。ステップ2では、あて先が送信元からの通信を受け入れる場合には、受け入れたことを示す確認応答 (ACK) に加え、送信元へのSYNを送信する。この際、SYNとACKの両方の制御フラグを1にして送信する。ここで、ACKビットがセットされたセグメントをACKセグメント、または単にACKと呼ぶ。ステップ3では、送信元があて先からのSYNを受け入れ、ACKを送り返す。以上の3ステップにより接続を確立させることで、信頼性の高い通信を行う。

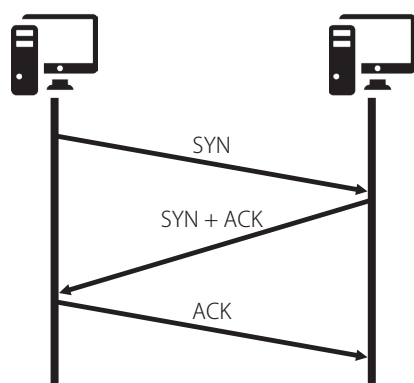


図1：3ウェイハンドシェイク

2.2 TCP の輻輳制御

図2にTCPの輻輳制御プロセスを示す。輻輳制御アルゴリズムは、スロースタートプロセス、輻輳回避プロセス、リカバリープロセスと呼ばれるアルゴリズムの使い分けにより輻輳の制御を行っている。TCPでは、いかに輻輳を避けつつ効率的にデータ送信量を制御するかは、古くから研究が進められ、これまでに多くの輻輳制御アルゴリズムが開発された。

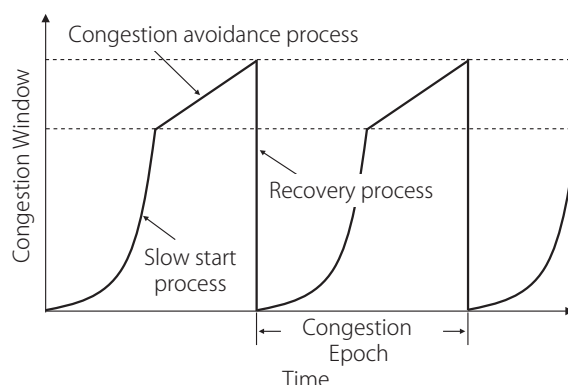


図2：TCPの輻輳回避プロセス

スロースタートプロセスとは、送信するセグメントを徐々に増やしていくアルゴリズムである。あるアプリケーションがデータ送信を始める際に、どの程度のデータレートが適切であるかは送信者側には判別できない。そのため、通信開始直後に大量のデータを送信し始めると、既に混雑しているネットワークに送信される可能性がある。この場合、通信を開始してからすぐに輻輳が発生することになる。この状態を避けるためTCPでは、通信開始時にスロースタートプロセスに従ってデータを送信する。スロースタートの更新式は式(1)に示される。

$$cwnd = cwnd + mss \quad (1)$$

スロースタートプロセスでは、送信元のcwndの初期値を1にセットする。その後、送信元がcwndで送出するセグメントの数を決め、あて先に向けて送出する。cwndは送信元においてひとつのACKセグメントを受信するたびに1セグメントだけ増やされる。そのため、RTT (Round Trip Time) 間隔でcwndの値が倍となり、指数的に増えていく。この指数的な増加はセグメントの損失が発生するまで続く。セグメント損失が発生した場合は、リカバリープロセスに移行し、スロースタート閾値 (sssthresh) を計算する。以降は、cwndのサイズがsssthreshを超えた場合にスロースタートプロセスを終了して輻輳回避プロセスに移行する。

スロースタートは指数的に輻輳ウィンドウを増加させるため、時間の経過とともにデータ送信量が増大する傾向を示す。セグメントの消失後も、スロースタートによるセグメントの送信を継続した場合、再び輻輳が発生する可能性もあるため、これを防ぐためのアルゴリズムが輻輳回避プロセスである。このプロセスでは、cwndがsssthreshに達した後は、ACKを受け取るたびに増やす輻輳ウィンドウの増加分を緩和すること

で、輻輳の発生を抑制する。このときのcwndの更新式は式(2)に示される。

$$cwnd = cwnd + mss / cwnd \quad (2)$$

これにより、cwndはRTTごとにmssずつ線形に増加していく形となり、輻輳が発生したときのcwndサイズまで徐々に送信量を上げていくことになる。その後、セグメントの損失が発生した場合は、リカバリープロセスに移行する。

輻輳を検知した際、毎回スロースタートプロセスから通信を再開するのは、必ずしも転送効率が良いとはいえない。転送効率を上げるため、輻輳検知による、再送後のcwndを小さくし過ぎないようにするアルゴリズムがリカバリープロセスである。リカバリープロセスにおけるセグメントの損失またはタイムアウトが発生した場合のcwndの制御方法は、TCPのバージョンによって異なる。

TCPの輻輳制御アルゴリズムには、大きく分けて損失ベース方式、遅延ベース方式、ハイブリッド方式の三つがある。損失ベース方式は、輻輳状態であるか判断するための指標として、パケット損失を用いる手法である。パケット損失の発生を検出するまではRTTごとに輻輳ウィンドウを増加させていき、パケット損失の発生が検出された場合には、輻輳ウィンドウを減少させることで輻輳を制御する。この制御方式はその方法からAIMD (Additive-Increase/Multiplicative-Decrease) 方式と呼ばれている。AIMD制御のイメージについては図3に示される。

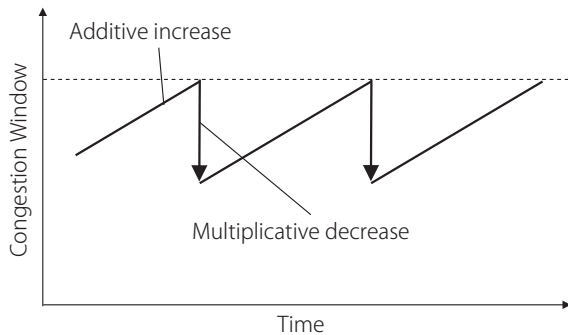


図3：AIMDのイメージ

遅延ベース方式は、輻輳状態であるか判断するための指標として、パケットの遅延を用いる方式である。遅延が非常に小さい場合、輻輳ウィンドウを増加させ、遅延が大きい場合、輻輳ウィンドウを減少させることで輻輳を制御する。ハイブリッド方式は損失ベース方式と遅延ベース方式を組み合わせた方式である。そのため、損失ベースにおけるパケット損失と、遅延ベースにおけるパケットの遅延の両方を輻輳状態の指標としている。

2.3 NewReno

図4にNewRenoの輻輳ウィンドウの変化を示す。NewRenoは、損失ベースの輻輳制御アルゴリズムのひとつで、パケットが破棄されるまでRTTごとに輻輳ウィンドウを暫時増加さ

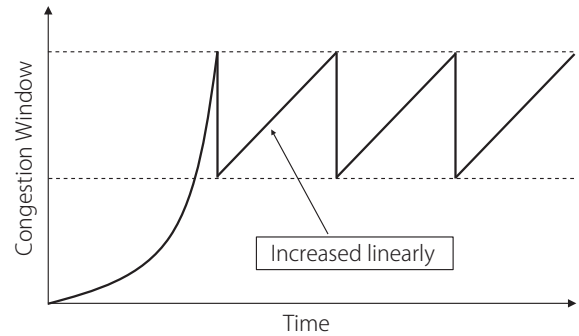


図4：NewRenoの輻輳ウィンドウの変化

せ、パケットが破棄されると、破棄された直前の値の半分まで輻輳ウィンドウを減少させることで輻輳を制御する（Henderson et al., 2012）。ここでRTTとは、データ送信後、応答が返ってくるまでの時間のことをいう。

2.4 Vegas

図5にVegasの輻輳ウィンドウの変化を示す。Vegasは、遅延ベースの輻輳制御アルゴリズムのひとつで、データ転送を実行しながらRTTを計測し、実際のスループットと期待スループットのふたつを比較する。この際、「実際>期待」になった場合は、ネットワークが混んでいるとみなして輻輳ウィンドウサイズを減少させ、転送速度を落とす。値が「実際<期待」になった場合は、帯域に余裕があるとみなして輻輳ウィンドウサイズを増加し、転送速度をあげる。式(3)に時刻tにおける輻輳ウィンドウのサイズ $\omega_s(t)$ について、時刻t+1で更新された輻輳ウィンドウのサイズの値を表す式を示す（Low et al., 2000: 3）。

$$\omega_s(t+1) = \begin{cases} \omega_s(t) + \frac{1}{D_s(t)} & \text{if } \frac{\omega_s(t)}{d_s} - \frac{\omega_s(t)}{D_s(t)} < \alpha_s \\ \omega_s(t) - \frac{1}{D_s(t)} & \text{if } \frac{\omega_s(t)}{d_s} - \frac{\omega_s(t)}{D_s(t)} > \alpha_s \\ \omega_s(t) & \text{else} \end{cases} \quad (3)$$

d_s は往復の伝搬遅延であり、 $D_s(t)$ は実際に測定されたRTTを表す。 $\omega_s(t)/d_s$ とは期待スループット、 $\omega_s(t)/D_s(t)$ は時刻tにおける実際のスループットを指しており、これらの差を閾値 α_s と比較した結果に応じて $\omega_s(t)$ を増減させることで、輻輳

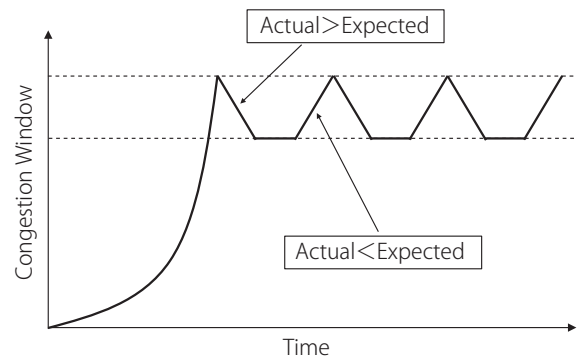


図5：Vegasの輻輳ウィンドウの変化

を制御する。

2.5 Reactive TCP

図6に Reactive TCP の動作の概要を表したフローチャートを示す。こちらは、損失ベースと遅延ベースを状況に応じて使い分ける方式となっており、パケット損失を輻輳の指標としている。このアルゴリズムでは、通常は損失ベースで通信を行い、輻輳を検知しない場合には引き続き損失ベースでの通信を行う。輻輳を検知した場合には遅延ベースに切り替えて処理を行い、損失ベースによる通信を再開するといった動作を行う。本研究ではフローチャートに示した処理を実装し、実験を行った。

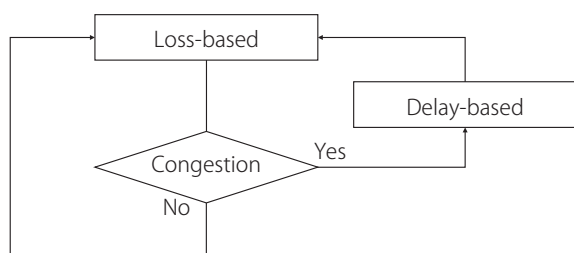


図6：Reactive TCPのフローチャート

この手法はハイブリッド方式と呼ばれ、各方式が混在した環境において公平性を維持する(長谷川他, 2011)。これより、スループット低下の抑制が見込まれるため、有効な輻輳制御方式といえる。

3. 実験方法

3.1 開発環境

ホスト OS は Windows 11、ゲスト OS は Ubuntu 20.04.5 LTS とした。また、シミュレータには、ネットワークシミュレータである ns2 (Network Simulator 2) を使用した。ns2 は米国 DARPA の研究プロジェクト VINT の研究成果であり、プロトコル等のコアモジュールの実装は C++ 言語、トポロジーなどのシミュレーションシナリオを記述する言語として OTCL スクリプト言語を利用している (Issariyakul and Hossain, 2009)。

3.2 実験環境の構築

図7に構築したネットワークトポロジーを示す。本研究では、ns2 を用いてネットワークトポロジーを作成した。トポロジーには、TCP 輻輳制御の研究を行う際によく用いられる基本的なシミュレーションモデルを利用し、シミュレーション中に損失ベースと遅延ベースの TCP を切り替えるアルゴリズムを実装した。

このトポロジーは6個のノードと5つのリンクで構成されており、ノードはそれぞれ node0 ~ node5 とする。ここで、Source はパケットの送信元を示し、Sink はパケットのあて先を示す。ノード間のリンクの詳細については表1に示される。

表1：リンクの詳細

node0-node2 (TCP)	Full duplex, 20 Mbps, Delay 10 ms, DropTail
node1-node2 (CBR)	Full duplex, 20 Mbps, Delay 10 ms, DropTail
node2-node3 (TCP, CBR)	Full duplex, 20 Mbps, Delay 20 ms, DropTail
node3-node4 (TCP)	Full duplex, 20 Mbps, Delay 10 ms, DropTail
node3-node5 (CBR)	Full duplex, 20 Mbps, Delay 10 ms, DropTail

3.3 シミュレーションの詳細

TCP Source である node0 で TCP フローを生成し、node4 に向かって流し始める。対して CBR Source である node1 では CBR フローを生成し、node5 に向かって流し始める。そのため、node0、node2、node3、node4 間には TCP フロー、node1、node2、node3、node5 間には CBR フローを流れることになり、重複している node2、node3 間でパケットどうしの衝突を引き起こし、輻輳を発生させた (水野, 2011; 銭, 2014)。このときの node2、node3 間はボトルネックリンクと呼ばれる。

実験では、CBR を 1、2、3、…、10 Mbps と変化させていった際のスループットとパケット損失数を計測した。なお、1 回のシミュレーション時間を 30 秒とする。その他の実験の

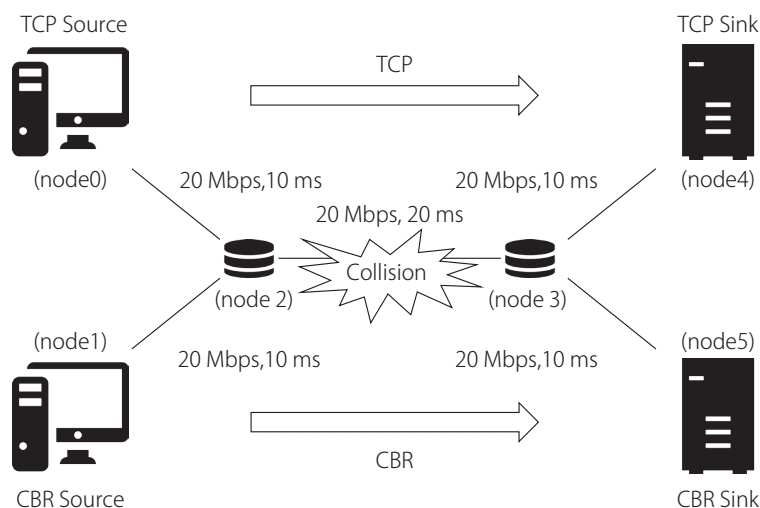


図7：ネットワークトポロジー

表2：実験の詳細

Data ①	- Hybrid method combining loss-based and delay-based methods. - Using NewReno and Vegas.
Data ②	Slows down the switching time of ①. (3/2 times as long as in ①)
Data ③	Accelerate the switching time of ①. (2/5 times as long as in ①)

詳細については表2に示される。

3.4 スループットの計算

スループットの計算には、式(4)を用いる。本実験では、1回のシミュレーション時間を30秒としているため、 Δt には30sを代入して計算を行う。

$$\text{Throughput}[\text{bps}] = 8 * \frac{\text{data}}{\Delta t} \quad (4)$$

3.5 パケット損失数の計算

パケット損失数の計算には、トレースファイルを用いる。トレースファイルに記述されている各種情報から、TCPのパケットが破棄された情報のみを抽出し、カウントする。これによりシミュレーション中に破棄されたTCPの総パケット数を算出する。また、シミュレーション毎にパケット損失数を求めた場合、非常に多くのデータが算出されるため、本実験では、最も通信量が多いCBRが10Mbpsのときの各シミュレーションにおけるパケット損失数を算出した。

4. 実験結果および考察

4.1 スループット

図8にCBRとスループットの関係を示す。得られたデータに加え、参照のため既存のTCPであるNewRenoの測定結果を記し、実験で得られたデータと比較した。

CBRが7Mbps以下においてはデータ③が最も高いスループットを記録している。8Mbps以上においてはデータ①が最も高いスループットを記録している。したがって、通信量が多い環境ではデータ①のTCPを用いることにより高い効果を

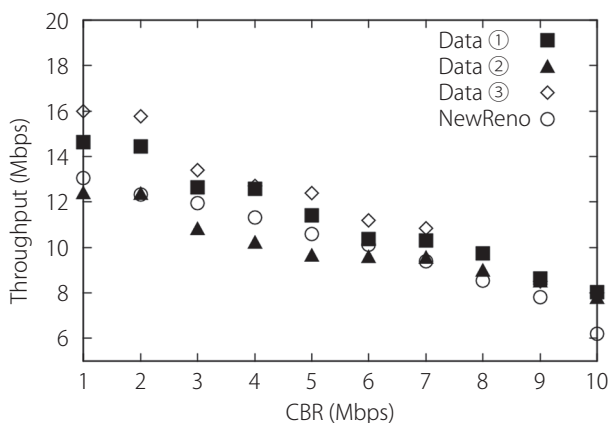


図8：CBRとスループットの関係

発揮し、通信量が少ない環境ではデータ③のTCPを用いることで高い効果を発揮できると考えられる。データ②のTCPに関しては、本実験環境において特に目立った特徴は見られなかったが、実験条件等を変更した場合に効果を発揮する可能性もあるため、パラメータの変更やCBRのさらなる増加といった検討が必要である。

また、CBRが1Mbpsのときと10Mbpsのときのスループットの差を計算し、通信量が増加した際の各TCPの減少幅についても考える。本実験において最も減少幅が大きいのはデータ③であった。反対に最も小さいのはデータ②であった。データ③が最も大きくなった要因として、CBRが2Mbps以下といった通信量が非常に小さい環境において高いスループットを記録しているにもかかわらず、3Mbps以上で急激に低下していることがグラフより明らかである。これは、通信の安定性に欠ける。一方、データ②では、スループットは全体的に低いものの、CBRの増加による影響をあまり受けていない。したがって、通信の安定性を重視する場合にはデータ②を用いることが最適である。

4.2 パケット損失数

表3にCBR = 10Mbpsにおけるパケット損失数を示す。本実験において最も通信量が多い実験では、データ①が6 packetsと最も小さい結果となった。この結果から、NewRenoよりもパケット損失数が大幅に減少していることがわかる。このようになった要因として、二つの理由が考えられる。一つ目は、ハイブリッド方式による切り替えのタイミングである。本実験では輻輳が発生し、輻輳回避プロセスに入る直前にTCPを切り替えているため(データ①)、本来の輻輳回避プロセスに入るNewRenoよりもパケット損失数が少ないと推察される。二つ目は、通信量が全体的に少ないことである。本実験ではCBRが10Mbpsまでとしているが、実際のインターネット環境と比較すると非常に小さいため、NewRenoが効果を十分に発揮できなかったと推察される。

介護・福祉施設のシステムなどと併用した場合、通信速度に加えてデータの確実性も要求されるため、本実験において総合的に高いスループットと安定性を記録していて、かつパケット損失数が少ないデータ①が最適といえる。

本研究では、Reactive TCPの切り替えの部分をシミュレータで実装したが、より具体的な処理内容を検討し、エミュレータなどで実装を行うことでより詳細な実験結果が得られるため、検討の余地がある。また、実際のインターネット環境で同様の実験を行うことで、より正確な実験結果も得られると考える。実験の方法としては、一般家庭向けインターネット環境や、教育組織等で用いられる教育機関向けネットワーク環境、クラウドサーバーなどの異なるネットワーク環境に実験端末を設置し、本論で述べたReactive TCPのアルゴリズム

表3：パケット損失数(CBR, 10 Mbps)

	Data ①	Data ②	Data ③	NewReno
Packet Loss (packets)	6	11	8	32

により通信を行う。その際にWiresharkといったパケットキャプチャツールにより計測、分析を行う方法や、iPerf3といった測定ツールを利用する方法などが挙げられる。

5. 結言

本研究では、ネットワークシミュレータであるns2を用いてネットワークポロジを作成し、同一環境下における輻輳発生時のスループットとパケット損失数を計測した。得られた結果の要約を以下に示す。

- CBR が7 Mbps以下においては、切り替えのタイミングを早めたデータ③が最も高いスループットを記録し、8 Mbps以上では、パケットが破棄されるタイミングで切り替えたデータ①が最も高いスループットを記録したことから、通信量が多い環境ではデータ①が効果を発揮し、通信量が少ない環境ではデータ③が効果を発揮する。
- スループットの減少幅では、切り替えのタイミングを遅くしたデータ②が最も小さくなったことから、安定性を重視する場合にはデータ②が最適である。
- パケット損失数では、データ①が最も少なく、NewRenoが最も多い結果となったが、実際のインターネット環境とは異なるため、パラメータ変更などの検討が必要である。
- Reactive TCPは、ネットワークの輻輳の緩和に有効であり、スループットとパケット損失の2つの観点から、膨大な数のセンシティブな個人情報を扱う介護・福祉施設における安定かつ正確な通信に寄与するといえる。しかし、本手法はネットワーク環境の特性やトラフィックの発生パターンに依存する。したがって、すべての通信環境で一様な改善が得られるとは限らない。

引用文献

- 長谷川剛・村田正幸（2011）. TCPの輻輳制御機構に関する研究動向.
- Henderson, T., Floyd, S., Gurtov, A., and Nishida, Y. (2012). *The NewReno modification to TCP's fast recovery algorithm*. RFC 6582.
- Issariyakul, T. and Hossain, E. (2009). *Introduction to network simulator NS2*. Springer.
- 厚生労働省（2023）. 第8期介護保険事業計画に基づく介護職員の必要数.
- 厚生労働省（2023）. 我が国における生産年齢人口の推移と将来推計.
- Low, S., Peterson, L., and Wang, L. (2000). *Understanding TCP vegas: A duality model*.
- 水野秀樹（2011）. ns2によるネットワークシミュレーション入門 第1版. 森北出版.
- 銭飛（2014）. ns3によるネットワークシミュレーション 第1版. 森北出版.

Copyright © 2025 Society for Science and Technology



This article is licensed under a Creative Commons [Attribution-NonCommercial-NoDerivatives 4.0 International] license.



<https://doi.org/10.11425/sst.14.19>

受稿日：2025年3月3日
受理日：2025年5月7日
発行日：2025年6月30日