氏　　名	長田　繁幸
授与した学位	博　士
専攻分野の名称	工　学
学位授与番号	博甲第５５４３号
学位授与の日付	平成２９年　３月２４日
学位授与の要件	自然科学研究科　産業創成工学専攻 （学位規則第４条第１項該当）
学位論文の題目	データセンターネットワークにおけるスループット急落の回避に関する研究
論文審査委員	教授　横平　徳美　　教授　杉山　裕二　　准教授　野上　保之

学位論文内容の要旨

商用データセンターでは，大量のデータ処理を行うために，巨大なデータをいくつかに分割して複数のサーバに分割保存する分散ファイルシステムを使う方法が普及してきている。そのシステムでは，トランスポート層プロトコルに TCP（Transmission Control Protocol）を使う場合が多いが，データ処理を行うクライアントは，データ読込みを行うときに，複数のサーバに一斉にデータ送信要求を行うため，ネットワークの一部のスイッチにパケットの到着が集中する。そのとき，スイッチのポートバッファに入りきらなかった大量のパケットが破棄されるため，サーバに確認応答（ACK: Acknowledgement）が届かず，再送タイムアウトによる再送を待つことになる。この再送タイムアウト時間（RTO: Retransmission Timeout）の最小値は 200 ミリ秒が標準であるが，これはデータセンターのネットワークの往復時間（通常数 100 マイクロ秒）よりも非常に大きいため，輻輳が解消された後，ネットワークがアイドル状態であるにもかかわらず，再送タイムアウトを待ち続けることになり，その結果，スループットが急激に低下することが知られている。この現象のことを TCP インキャスト（以下，単にインキャスト）と呼ぶ。

従来，インキャストを回避するために，大きく 2 つのアプローチが検討されている。一方は，パケット到着の集中を避けるために，TCP のパラメータを調整する方法であるが，サーバ数が大きいときには，集中することが避けられず，インキャストを回避できない。 他方は，パケット消失が発生したときに素早く再送を行うために，RTO の最小値を， TCP 標準値の 200 ミリ秒から数 100 マイクロ秒の値に変更する方法（以下，FGTCP）である。この方法は一定の性能改善を示すが，サーバ数が大きいときは，やはりインキャストが発生する。

本論文では，まず，分散ファイルシステムを活用する一般的なデータセンターのネットワーク構成を念頭に，すべてのリンクの帯域幅が同じであり，かつ，クライアントリンクがボトルネックとなる場合を対象として，サーバ数が大きいときもインキャストを回避できるように，コネクション設定を直列化する方法を 3 つ検討する。1 つ目は，コネクションを 1 つずつ順番に設定することで，ポートバッファで待機するパケットをほぼ無くす方法である。2 つ目は，1 つ目のスループットを更に向上させることを目的に，TCP 輻輳制御のスロースタート特性を考慮して，次のコネクションの開始をスロースタート期間分だけ早めて，コネクションを順番に設定する方法である。3 つ目は，2 つ目の方法を適用できない環境でも適用できるように，複数のコネクションを 1 つのコネクションと見なして，直列化する方法である。これらの方法についてシミュレーションを行った結果，どの方法もポートバッファのオーバーフローを起こすことなく，また，3 つ目の方法が，最も高いスループットを達成することを明らかにした。

次に，本論文では，上述のネットワーク構成の前提を取り除いた環境に拡張して，ポートバッファのオーバーフローを許容する前提で，インキャストを回避する方法として，FGTCP の改善に取り組む。FGTCP の性能低下原因を分析したところ，再送が繰り返し発生したことで，TCP の指数バックオフ機構によって，再送毎に RTO が指数的に増加し，その結果，ネットワークにアイドル期間が生じていた。このアイドル期間を極力短くするために，4 つの方法を検討する。1 つ目は，RTO が過度に大きくならないように，再送が繰り返されたときの RTO 値の計算方法を，標準の指数的増加方法から線形的増加方法に変更する方法である。2 つ目は，その計算方法を更に改良して，指数的増加方法と線形的増加方法を，データセンターのネットワーク構成とその時点の RTO に応じて使い分ける方法である。また，3 つ目と 4 つ目は，クライアントがアイドル期間を検出したら，サーバに対して陽に再送を要求する方法である。それを実現するために，前者は新しく導入したオプションを送信し，後者は，通常の送受信では起こりえない 1 バイトの ACK を送信することで，サーバにアイドル期間であることを知らせる。サーバはこれらを受け取ると，アイドル期間が発生していると認識して，再送タイムアウトを待つことなくパケットを再送する。これらの方法をネットワークシミュレータに実装し，インキャスト回避の効果について確認したところ，FGTCP と比較して最大で 20%程度再送するパケットが増えるが，どの方法もほとんどの場合で，スループット性能が改善した。また，4 方式を互いに比較すると，ネットワークの環境に依存して，1 つ目と 2 つ目の方式のどちらかが最も高いスループット性能を示した。

論文審査結果の要旨

商用データセンターでは,大量のデータ処理を行うために,巨大なデータをいくつかに分割して複数のサーバに分割保存する分散ファイルシステムを使う方法が普及してきている。そのシステムでは，トランスポート層プロトコルにTCP（Transmission Control Protocol）を使う場合が多いが，データ処理を行うクライアントは，データ読込みを行うときに，複数のサーバに一斉にデータ送信要求を行うため，ネットワークの一部のスイッチにパケットの到着が集中する。そのとき，スイッチのポートバッファに入りきらなかった大量のパケットが破棄されるため，サーバに確認応答（ACK: Acknowledgement）が届かず，再送タイムアウトによる再送を待つことになる。この再送タイムアウト時間（RTO: Retransmission Timeout）の最小値は200ミリ秒が標準であるが，これはデータセンターネットワークの往復時間（通常数100マイクロ秒）よりも非常に大きいため，輻輳が解消された後，ネットワークがアイドル状態であるにもかかわらず，再送タイムアウトを待ち続けることになり，その結果，スループットが急激に低下することが知られている。この現象のことをTCPインキャスト（以下，単にインキャスト）と呼ぶ。

本論文では，まず，すべてのリンクの帯域幅が同じであり，かつ，クライアントリンクがボトルネックとなる場合を対象として，サーバ数が大きいときもインキャストを回避できるように，コネクション設定を直列化する方法を3つ検討している。そして，それら3つの方法の中で，複数のコネクションを1つのコネクションを見なして直列化する方法が最も優れていることを明らかにしている。

次に，本論文では，上述のネットワーク構成の前提を取り除いた環境に拡張して，インキャスト回避法を検討している。従来，優れたインキャスト回避法として，パケット消失が発生したときに素早く再送を行うために，RTOの最小値をTCP標準値の200ミリ秒から数100マイクロ秒の値に変更する方法（以下，FGTCP）が提案されている。しかし，FGTCPでも，サーバ数が大きい時はスループット低下が起こってしまう。本論文では，このFGTCPの改善に取り組んでいる。まず，FGTCPのスループット低下の原因が，再送が繰り返し発生することで。TCPの指数バックオフ機構によって再送毎にRTO値が指数的に増加し，その結果，ネットワークにアイドル期間が生じていることであるということを明らかにしている。次に，このアイドル期間を極力短くするために，4つの方法を検討している。そして，それら4つの方法の中で，再送が繰り返されたときのRTO値の計算方法を標準の指数的増加方法から線形的増加方法に変更する方法，および，その計算方法を更に改良して，指数的増加方法と線形的増加方法をデータセンターのネットワーク構成とその時点のRTO値に応じて使い分ける方法が最も優れていることを明らかにしている。

以上のように，本論文は，今後益々重要性が増すと言われているデータセンターにおける大量データ処理を効率良く行うための方法を提案しており，情報通信技術を基盤とする現代社会に対する貢献は極めて大きいと考えられる。よって，本論文は，博士(工学)の学位に値すると認める。