L-016

DCB を用いたデータ収集システムにおける輻輳制御アルゴリズムの提案
A Proposal of Congestion Control Algorithm in Data Acquisition Systems using Data Center Bridging

今池 怜生† 西村 俊彦† 岩井 慎之介† 長坂 康史†
Reo Imaike Toshihiko Nishimura Shin-nosuke Iwai Yasushi Nagasaka

1. はじめに

近年，情報技術の発展に伴い，大規模科学実験の分野においても大量のデータを収集，及び解析するために大規模データ収集システムを利用するようになってきた．この大規模データ収集システムは多くの場合，実験効率を高める目的で多数のネットワークスイッチを介した並列処理を行っている．また，多数のノードで生成された膨大なデータを 1 つの解析用コンピュータへ集めるために，多対一の同時通信を行うが，そのトラフィックの特徴から通信経路中で輻輳が発生し，スループットの低下の原因となるパケット損失を招く恐れがある．この TCP/IP 通信に代表される Ethernet においてパケット損失の問題を解決するために，ロスレス Ethernet の実現を可能とする Data Center Bridging (DCB)技術の研究が進められている [1]．また，DCB の他に TCP による輻輳制御がネットワーク上の輻輳を防ぐ機構として存在する．この 2 種類の機構によって DCB はフロー制御を，TCP は輻輳制御を行い，輻輳を回避している．TCP の輻輳制御アルゴリズムは様々な種類のものがあり，その種類によってネットワーク全体の性能が変わる．

しかし，DCB 環境下で動作する大規模データ収集システムに適した輻輳制御の手法については確立されていない．

そこで本研究では，DCB 環境における輻輳制御アルゴリズムの種類による違いを調査し，そこで得られた結果を基に，DCB を用いたデータ収集システムに適合する輻輳制御アルゴリズムを提案する．

2. DCB

2.1 DCB 概要

DCB は，Ethernet の機能を拡張した規格である．データセンタ内の Storage Area Network (SAN)や LAN といった複雑化したネットワークの通信の統合を図る．また，DCB はロスレス Ethernet を実現するために，優先度に基づいてトラフィックを区別し，通信帯域保証や輻輳検知を行う機能を備えている．IEEE802.1WG 内の DCB Task Group において標準化が進められ，主要な規格として Enhanced Transmission Selection (ETS)，Priority-based Flow Control (PFC)，Congestion Notification (CN)がある．

2.2 ETS

ETS は IEEE802.1Qaz で定義されており，異なるアプリケーションの各トラフィックに，最低帯域を割り当てて制御することを可能にする規格である．IEEE802.1Qaz では優先度グループ (Priority Group)と呼ばれる概念を導入し，それぞれの Priority Group に対し最低帯域を，全体に対する割合として指定することが可能である．

2.3 PFC

PFC は IEEE802.1Qbb で定義されており，現在の IEEE802.3x における PAUSE を拡張した規格である．PFC は，トラフィックへ Priority を付与し，論理的に Priority 毎にリンクを分離することができる．また，PFC が各々の受信キューの状態を監視しており，受信キューが溢れそうになったら輻輳に起因するパケットロスを回避するために PAUSE を送信してフロー制御を行う．

2.4 CN

CN は IEEE802.1Qau で定義されている規格である．PFC ではスイッチの受信キューが溢れないように隣接ノード間でフロー制御を行っていたが，CN ではネットワーク上において輻輳の原因となるパケットを検知し，そのパケットを送信している送信元へ輻輳を通知することで送信量を制限し，ネットワーク上の輻輳を制御する．

3. 輻輳制御アルゴリズム

3.1 輻輳制御アルゴリズム概要

輻輳制御は，送信者側の輻輳ウィンドウ (cwnd)を制御し，ネットワークの輻輳状況に合わせてデータの送信を抑制する．この cwnd の制御を行うアルゴリズムを輻輳制御アルゴリズムと呼ぶ．

本研究では，DCB 環境下における輻輳制御アルゴリズムの違いを確認するため，様々な輻輳制御アルゴリズムの中からロスベース方式の CUBIC，遅延ベース方式の Vegas，ハイブリッド方式の Illinois を扱うこととする．

3.2 CUBIC

CUBIC は，パケットロスを輻輳検知の指標として用いているロスベース方式と呼ばれる種類の輻輳制御アルゴリズムである．この CUBIC では，特有の増加関数を用いて cwnd を制御する．さらに，最後にパケットロスを起こしてから経過した時間を基にしており，Round Trip Time (RTT)を輻輳検知の指標として用いていないため，RTT の大小様々なトラフィックが存在していても公平性を保つことが可能である．

3.3 Vegas

Vegas は，RTT の実測値と推測値を比較して cwnd を制御する遅延ベース方式と呼ばれる種類の輻輳制御アルゴリズムである．RTT の実測値が RTT の推測値より大きい場合は cwnd を 1 減らしてスループットを抑制し，RTT の実測値が RTT の推測値より小さい場合は 1 増やし，スループットを大きくする．

† 広島工業大学, Hiroshima Institute of Technology

3.4 Illinois

Illinois は，ロスベース方式と遅延ベース方式を組み合わせたハイブリッド方式の輻輳制御アルゴリズムである．パケットロス前に測定した RTT 値と過去最小の RTT 値を比較し，パケットロス前に測定した RTT 値の方が小さい場合は cwnd を大きく増加させる．そうでない場合は cwnd の増加率を減少，或いは，負の値にする．また，3 回重複 ACK を受信すると cwnd を減少させる．

4. 性能評価

4.1 性能評価実験

DCB 環境における輻輳制御アルゴリズムの種類による違いを測定する．性能評価実験は DCB 環境下で行ない，1 対 1 の直接接続による性能評価実験を実施した．ネットワーク構成を図 1 に示す．また，Server のスペックを表 1 に，Client のスペックを表 2 に示す．実験は Client から Server へ向けて Iperf3 を使用してパケットを送り，60 秒間の測定を行って，スループット及び cwnd の時間推移における大きさの変化を求めた．

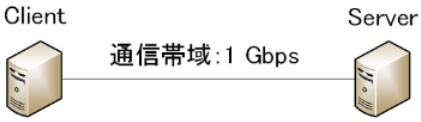

図 3　ネットワーク構成

表 1　Server スペック

CPU	Intel Xeon CPU E5-1620 v3
RAM	16384 MByte
OS	Scientific Linux 7.2 X64_86
NIC	Intel X540-T2

表 2　Client スペック

CPU	Intel Xeon CPU E5-1410 v2
RAM	8192 Mbyte
OS	Scientific Linux 7.2 X64_86
NIC	Intel X540-T1

性能評価実験で求めた輻輳制御アルゴリズム毎のスループットの大きさを表 3 に，cwnd の時間推移による大きさの変化を図 2 に示す．表 3 から CUBIC は 9.17 Gbps，Vegas は 3.62 Gbps，Illinois は 9.37 Gbps であることが分かった．なお，再送回数は全ての輻輳制御アルゴリズムの場合で 0 回であった．また，図 2 より，CUBIC は 8 秒から cwnd の大きさが 3.06 MB で一定になるという結果が得られた．Vegas は常に不規則な値をとり，cwnd の最小値は 0.04 MB，最大値は 0.11 MB となった．Illinois は 3 秒まで 1.01 MB となり，8 秒までは 1.33 MB で 9 秒からは 2.31 MB となった．

表 3　各輻輳アルゴリズムにおけるスループット

輻輳制御アルゴリズム	Throughput [Gbps]
CUBIC	9.17
Vegas	3.62
Illinois	9.37

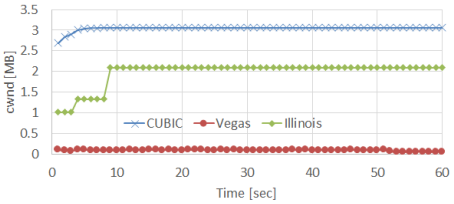

図 1　各輻輳制御アルゴリズムにおける cwnd の時間推移

ここで，ロスベース方式の CUBIC がパケットロスに起因する再送を行っているか確認するために，CUBIC について DCB 機能が ON の時と OFF の時の cwnd の時間推移における大きさの変化を求めた．その結果を図 3 に示す．

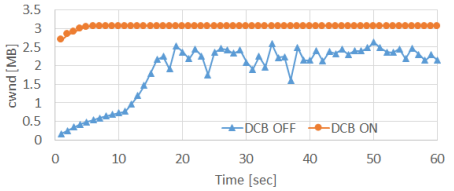

図 2　DCB ON と DCB OFF による cwnd の大きさの違い

図 3 より，DCB ON 時の cwnd の大きさは DCB OFF 時と比べて大きく，一定であることが分かる．それに対して DCB OFF 時はパケットロスを検知して cwnd のサイズを小さくしていることが分かる．よって，DCB ON 時にはパケットロスに起因する再送を行っていないと判断できる．

5. 考察

表 3 より，CUBIC 及び Illinois に関しては Vegas に比べてスループットが大きいことが分かる．また，図 2 の cwnd の変化を見ると CUBIC と Illinois については cwnd の大きさが途中から一定であることが分かる．パケットロスを検知するタイプの輻輳制御アルゴリズムは一般的に図 3 の DCB OFF 時のようなグラフを描くが，CUBIC や Illinois はパケットロスを輻輳検知の指標としているため，ロスレス Ethernet の DCB 環境では動作しなかったと考えられる．それに対して，Vegas は cwnd が細かく増減していることが分かる．これはパケットロスを輻輳検知の指標としていないためロスレス Ethernet の DCB 環境下でも動作したと考える．よって，提案するアルゴリズムは Vegas のようなパケットロスを輻輳検知の指標としないアルゴリズムでスループットが高い値を示すような cwnd の制御を行うものである．

6. まとめ

本研究では，DCB 環境における輻輳制御アルゴリズムの種類による違いを確認し，そこで得られた結果を基に，DCB を用いたネットワークに適合する輻輳制御アルゴリズムを提案した．今後は，提案した輻輳制御アルゴリズムが DCB 環境下のデータ収集システムに適合しているかの検討及び公平性について調査し，独自の輻輳制御アルゴリズムを作成する．

参考文献

[1] 山木戸 啓亮, 長坂 康史, “大規模データ収集システムにおける DCB を用いたネットワーク QoS の向上に関する研究”, 情報科学技術フォーラム講演論文集, Vol.13, pp.119-120 (2013).