

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

A Study on improvement of the network QoS in the large-scale data acquisition system by using DCB

山木戸 啓亮†
Keisuke Yamakido

長坂 康史†
Yasushi Nagasaka

1. はじめに

近年の情報通信技術の急速な発展と普及により、様々な分野でネットワーク分散型のデータ収集システムが活用されるようになってきた。特に、大規模高エネルギー物理学実験では、巨大な測定器で測定されるデータを数千規模のノードによって構成された大規模データ収集システムで処理している。

大規模データ収集システムは、図 1 に示す様な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おり、目的の物理事象を解析するために、多数のノードで生成された膨大な測定データを一つの解析専用コンピュータへと集めるために多対一の TCP/IP 通信を行っている。

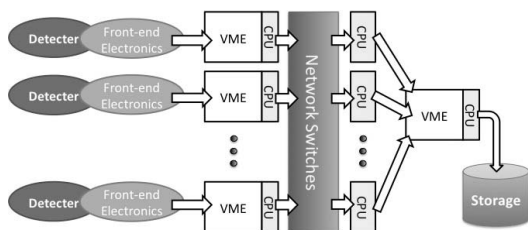


図 1 大規模データ収集システム概要

しかし、データを効率良く収集する上で、ノード間の各インタフェースのボトルネックがデータの輻輳を招き、通信効率が低下してしまう問題がある^[1]。

そこで、本研究では、データ収集システムのネットワークに、ロスレス・イーサネット技術の一つである Data Center Bridging (DCB) に対応した機器を導入することでデータの輻輳を抑制し、ネットワークの通信効率や信頼性を向上させることを目的とする。

2. DCB

2.1 DCB の概要

DCB とは、高い信頼性が要求される FibreChannel (FC) のプロトコルを信頼性が低い Ethernet フレームとして運ぶ FibreChannel over Ethernet (FCoE) の技術のために開発されたロスレス Ethernet 技術の一つである。従来の Ethernet ネットワークの機能を拡張した一連の機能を持ち、優先度に基づいてトラフィックを区別することができる。

DCB は、IEEE802.1 WG 内の DCB Task Group (TG)^[2]において標準化が進められている。DCB の主要な規格は次の 3 つである。

2.2 IEEE802.1Qaz: ETS

Enhanced Transmission Selection (ETS)は、各トラフ

ィックに対して定義された優先度をもとに、帯域制御を可能にする規格である。ETS では、Priority Group (PG) という概念を導入し、従来の絶対優先方式では困難であった最低帯域を PG 毎に保証することが可能である。

2.3 IEEE802.1Qbb: PFC

輻輳によるフレームロスをなくすためには、リンクレベルによるフロー制御が必須となるが、現在の IEEE802.3x による PAUSE では、性質の異なるトラフィックが同一物理リンク上を流れていた場合でも、すべてのトラフィックが止められてしまう。Priority-based Flow Control (PFC) は、輻輳によるフレームロスを回避するために、トラフィック毎に優先度を与え、優先度毎にリンクを論理的に分離する。これにより、特定の優先度に対してのみ PAUSE を送信することができ、トラフィック毎のフロー制御が可能となる。

2.4 IEEE802.1Qau: CN

Congestion Notification (CN)では、受信側ポートの受信キューをモニタし、輻輳を送信側のポートに通知する。通知を受けた送信側ポートはトラフィックシェイピングを行うことで、フレームロスが発生しないように転送量を調整する。TCP にも輻輳管理の仕組みがあるが、DCB の CN では TCP よりさらに下位の層で輻輳管理が行われ、より効率的な輻輳制御ができるので、上位のプロトコルとして TCP を使用する場合にもメリットがある。

3. データ収集トラフィック

高エネルギー物理学実験の大規模データ収集システムでは、イベントが発生するタイミングでそのイベントに関する情報を一斉に一台のコンピュータに集めているため、トラフィックは、多対一の同期トラフィックとなる。

このような特殊な環境では、従来の TCP/IP の QoS では、輻輳を完全に制御することは難しく、通信効率が低下してしまう可能性がある。また、データ収集システムでは、システムの汎用性を重視しているため、TCP/IP の活用は必要条件となっている。

本研究では、データ収集システムにおいて、多対一の通信が行われるノード間に DCB を用いることを提案する。これにより、既存の TCP の可用性を維持したまま、通信効率や信頼性を高めることが可能であると考えられる。

4. 性能測定

4.1 測定方法

データ収集システムを模した仮想環境に DCB を実装することで、ネットワークの通信効率や信頼性を評価を行うために、ネットワークの性能測定を行った。

測定には、DCB の規格に対応した Intel (R) 社の NIC X540-T1 を搭載した 2 台の PC を直接接続している。使用する PC の性能を表 1 に、実験環境を図 2 にまとめる。

†広島工業大学大学院工学系研究科

表1 PC性能(PC1, PC2)

CPU	Intel(R) Xeon CPU 2.13 GHz
Memory	2048 MB
NIC	Intel® X540-T1
OS	ScientificLinux6.4 CERN x86_64
TCP Algorithm	Cubic

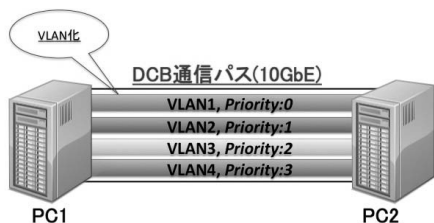


図2 測定環境

図2に示すとおり、本測定では、輻輳が発生し易い多対一通信を再現するために、VLANを用いた仮想ネットワークで4つのフロー生成し、各VLANに固有の優先度を割り当てた。また、DCBの有効性を確かめるために、DCB機能を有効にした場合と、無効にした場合との転送レートと再送パケット数の測定を行った。また、測定時間は10 sec、測定間隔は0.1 sec、測定回数は5回を設定し、転送レートを計測するネットワークベンチマークツールにiperf3^[3]を用い、PC1からPC2へのTCP通信を行う。また、DCB有効時の各規格については、ETSは無効とし、PFCとCNは有効とする。

4.1 測定結果

DCBを無効にした状態での4フローの転送レートを図3に、再送パケット数を図4に示す。

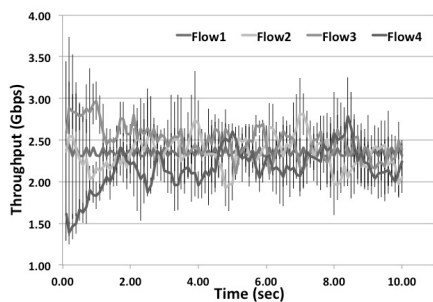


図3 転送レート(DCB:無効)

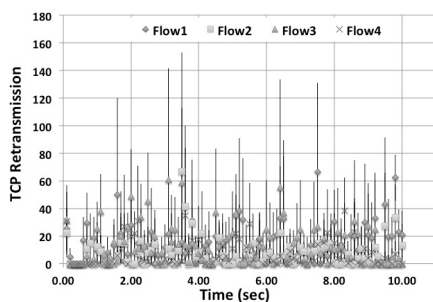


図4 再送パケット数(DCB:無効)

これらの図より、DCBを無効にした状態では各フローの転送レートの変動が大きく、平均転送レートは2.345 Gbps

となり標準偏差は0.242 Gbpsとなった。また、平均再送パケット数は4フロー合計で4007回発生した。

次に、DCBを有効にした状態での4フローの転送レートを図5に示す。

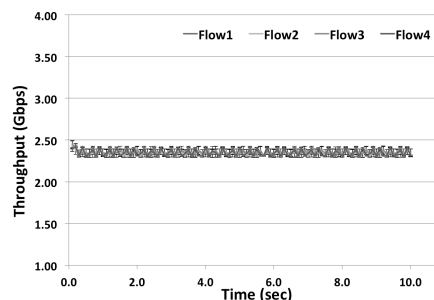


図5 転送レート(DCB:有効)

この図より、DCBを有効にした状態では、各フローの転送レートは、平均転送レートは2.355 Gbpsとなり標準偏差は0.038 Gbpsとなった。一方で、再送パケット数は、4フロー合計で0回と、一回も発生しなかった。

5. 考察

図5より、DCBを有効にした場合では、各フローの転送レートの上限と下限がそれぞれ約2.4 Gbps、約2.3 Gbpsとなっている。これは、ETSが無効であることから、CNによってトラフィックシェイピングが行われ、受信側より送信側のポートに対して帯域制御が行われていると考える。

また、各再送パケット回数より、DCB有効時には、輻輳が発生する前に転送レートを制御していることから、本実験を通してパケットの再送回数が0回という結果に至った。トラフィックに対して帯域制御メカニズムが働く場合、パケットロスを低減出来るが、帯域制限の影響で、転送レートが落ちる傾向があると考えられる。しかし、本実験では、DCB有効時の転送レートの低下は見られず、無効時に比べ約10 Mbpsほど高いことが分かる。

したがって、DCBを用いることでTCP/IPネットワークにおける通信効率と信頼性の向上が可能であり、データ収集システムに利用することで、既存の問題点を解決することが可能であると考えられる。

6. おわりに

本研究では、多対一通信の環境であるデータ収集システムの信頼性および通信効率を高めるため、通信にDCBを用いることを提案した。測定結果より、高い輻輳制御が可能であり、ロスレス・イーサネットをデータ収集システムに適用可能であることが分かった。

今後はシステムとしての有効性を追求するために、DCBに対応した環境でノード数を増やし、より多くのフローが存在する環境での検証が必要と考える。

参考文献

- [1] 沖恭志, 長坂康史, 「大規模データ収集システムにおける通信の効率化に関する研究」, 広島工業大学紀要 研究編, Vol. 47, pp. 127-131, 2013.
- [2] Data Center Bridging Task Group, <http://www.ieee802.org/1/pages/dcbbridges.html>, (2014年6月2日)
- [3] iperf3, <https://code.google.com/p/iperf/>, (2014年6月9日)