4ZC-03

3つのサブフローを利用可能とするMPTCPパケットスケジューリングアルゴリズムECFの拡張

高梨優也† 木村成伴‡

†筑波大学情報学群情報科学類 ‡筑波大学システム情報系情報工学域

1 はじめに

MPTCP (Multi-Path TCP) は，マルチパスで通信可能なトランスポート層のプロトコルである．通信で用いるそれぞれのパスは，従来のTCPを拡張して実現しており，サブフローと呼ばれる．異なる経路の複数のサブフローからなるマルチパス通信により，通信の冗長化やスループットの向上が期待されるが，これを達成するためには，パケット送信時にどのサブフローで送るかを決定するパケットスケジューリングアルゴリズムが重要になる．例えば，Linux標準のアルゴリズムでは，帯域や遅延に差があるサブフローを用いてマルチパス通信を行うと，シングルパスのTCP通信よりもスループットが低下する可能性があることが知られている．これはRTTのみを考慮してパケットを分配していくため，高速サブフローが低速サブフローの送信完了を待つアイドル状態が増えてしまうことで高速サブフローが活かしきれないことが主な原因である．この問題を解決するために，パケットスケジューリングアルゴリズムECF (Earliest Completion First) [1] が提案されてる．しかし，本アルゴリズムでは，同時に利用するサブフローは最大2つまでとなっており，3つ以上サブフローがある場合にサブフローを効率的に使用することができない．ノートPCなどでは，Ethernetと無線LAN，携帯電話網，Bluetoothなど，3つ以上のネットワークインタフェースを持つものも多い．そこで本論文では，ECFを3つのサブフローを同時利用できるように拡張したアルゴリズムを提案する．

2 ECFアルゴリズム

ECFアルゴリズムでは，2つのサブフローの内，RTTが短い方の高速，長い方を低速と分類する．そして，前者の輻輳ウィンドウが空いている限り，パケットは高速サブフローで送る．空きがなくなった場合で，この輻輳ウィンドウが空くのを待つよりも，低速サブフローで送信した方が速い場合には，低速サブフローでパケットを送信する．そうでなければ，高速サブフローの輻輳ウィンドウが空くことを待つ．これによって，ECFでは特にサブフロー同士の帯域が不均一な場合に，Linux標準のアルゴリズムよりも効率よくスケジューリングできていることが示されている．

表 1: 文字と添え字の意味

文字	意味
x_f, x_s, x_t	高速, 中速, 低速サブフロー
$\mathrm{RTT}_f, \mathrm{RTT}_s, \mathrm{RTT}_t$	高速, 中速, 低速サブフローのRTT
CWND_f	高速サブフローの輻輳ウィンドウサイズ
k	コネクションレベルの送信バッファ内の未送信パケット数

3 提案方式

提案アルゴリズムをAlgorithm 1に示す．また，表1に，Algorithm 1で扱う文字と添え字の意味を示す．Algorithm 1は，3つのサブフローを RTT が短い順に x_f, x_s, x_t とし，パケットを送信するサブフローを選択して，戻り値とする．

まず，高速サブフロー x_f が利用可能な場合は高速サブフローを利用する．もし高速サブフローが利用不可能でかつ，中速サブフロー x_s を利用するよりも高速サブフローが利用可能になることを待つのが速いと判断されれば中・低速サブフローは利用せず，高速サブフローが利用可能になるのを待つ（4行目）．

中速サブフローが利用可能で中速サブフローの利用が高速サブフローを使い続けるより高速と判断されれば中速サブフローを利用する（6行目）．同様の手順で低速サブフロー x_t の利用が高速サブフローが利用可能になるのを待つことよりも高速と判断されれば低速サブフローも利用する（8行目）．

An Extension of MPTCP Packet Scheduling Algorithm ECF with Three Subflows
Yuya TAKANASHI†, Shigetomo KIMURA‡
†College of Information Science, University of Tsukuba
‡Faculty of Engineering, Information and Systems, University of Tsukuba

Algorithm 1 3つのサブフローに対応したスケジューリングアルゴリズム

if x_f is available **then**
  return x_f
else if $\mathrm{RTT}_f + \frac{k}{\mathrm{CWND}_f} * \mathrm{RTT}_f < \mathrm{RTT}_s$ **then**
  return no available subflow
else if x_s is available **then**
  return x_s
else if $\mathrm{RTT}_t < \mathrm{RTT}_f + \frac{k}{\mathrm{CWND}_f} * \mathrm{RTT}_f$ && x_t is available **then**
  return x_t
else
  return no available subflow
end if

4 シミュレーション実験

本章では，シミュレーション実験を行い，提案方式の有効性を示す．ネットワークシミュレータは，ns3.19を MPTCP が利用できるように拡張されたもの [2] を使用する．1つの端末に NIC が3つあると仮定し，2台の端末同士を3つの Ethernet で接続しているとする．各経路の帯域が均一な場合を想定した実験では，それぞれのサブフローの帯域を 8.6Mbps に，不均一な場合を想定した実験ではそれぞれを 8.6, 0.3, 0.1Mbps に設定し，10MB–60MB のファイル転送にかかる時間を測定する．そして，提案方式と Linux 標準のアルゴリズム minRTT，高速と中速の2経路を利用した ECF，高速の単一経路のみを利用した場合の転送時間を比較する．なお，各サブフローの伝搬遅延は 20ms とした．

図 1 に，帯域が不均一な場合の実験結果を示す．提案方式は minRTT アルゴリズムよりもファイル転送時間が約 9%削減した．これは，提案方式において，0.3, 0.1Mbps の経路の利用することは遅延につながると判断されたため，それらの経路の利用を抑えたためである．その結果，単一経路や ECF を使う場合とファイル転送時間がほぼ同じになった．

図 2 に，帯域が均一な場合の実験結果を示す．図より，提案方式は minRTT アルゴリズムとほぼ同じ結果になったことが分かる．これは全ての経路を利用したほうが効率が良いと判断されたため，それぞれの経路をまんべんなく利用した結果である．なお，提案方式は ECF アルゴリズムよりもファイル転送時間が約 33%削減した．以上より，3つの経路を利用する場合の MPTCP において，ECF を拡張した当提案方式では総合的に minRTT よりも良い結果を示した．

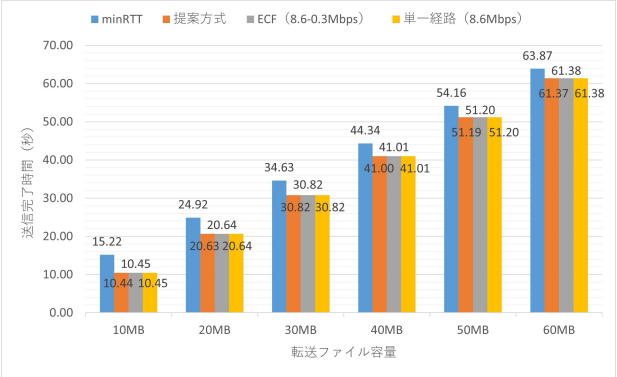

図 1: 不均一な経路におけるファイル転送時間

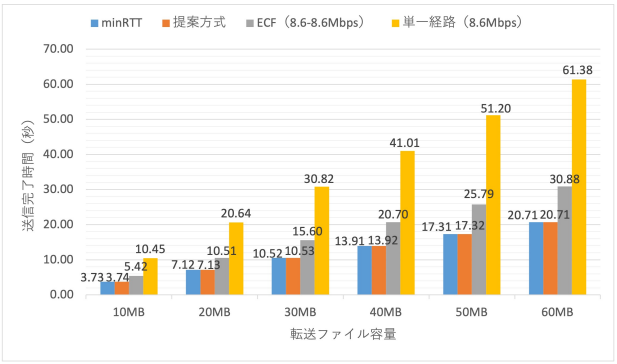

図 2: 均一な経路におけるファイル転送時間

5 まとめ

本論文では，2つの経路までしか利用できなかった ECF アルゴリズムを3つの経路で利用できるように拡張した．その結果，既存の minRTT と単一経路のみを利用する場合の両方のメリットを享受し，帯域が均一，不均一な経路とも良いパフォーマンスを得られた．しかし，既存の手法よりも目に見えて高速になるということはなかった．これを改善するため，後に送る予定のパケットを低速サブフローで先送りする等の対応が考えられる．

参考文献

[1] Yeon-sup Lim, Erich M. Nahum, Don Towsley, and Richard J. Gibbens, ECF: An MPTCP Path Scheduler to Manage Heterogeneous Paths, Proceedings of the 13th International Conference on emerging Networking EXperiments and Technologies (CoNEXT '17), pp. 147–159, 2017.

[2] Morteza Kheirkhah, Ian Wakeman, and George Parisis, Multipath-TCP in ns-3, Computing Research Repository (CoRR), http://arxiv.org/abs/1510.07721, 2015.