

スループット予測による経路選択を用いた OpenFlow環境でのMPTCP転送

前田 達憲¹ 阿部 洋丈² 加藤 和彦²

概要: ネットワークのスループットを向上させる手法としてマルチパス転送がある。Software-Defined Network(SDN) 環境ではプログラムによって自由に経路選択が可能であるため、これまでのネットワーク環境よりも Multipath TCP(MPTCP) 転送をより容易に実現することができる。しかし、自由に経路選択が可能であることはより効果的な経路選択が可能となる反面、膨大な経路の中から適切な転送経路の選択が必要となり、そのための手法が重要となる。本研究では、マルチパスの経路を選択する際の指標としてスループット予測を用いる MPTCP 転送を提案する。スループット予測を用いることで複数の経路の中からスループットが最大になるであろう経路を選択することができ、スループットが向上する。また、OpenFlow を用いて提案手法を実装し、動作検証を行った。

キーワード: Software-Defined Network, OpenFlow, MPTCP, スループット予測

MPTCP with path selection mechanism based on predicted throughput on OpenFlow-enabled environment

TATSUNORI MAEDA¹ HIROTAKE ABE² KAZUHIKO KATO²

Abstract: Multipath TCP (MPTCP) is a method that aims to increase data transfer throughput over computer networks. Because Software-Defined Network (SDN) can give us freedom of choice in selecting paths in a network, SDN will make it easier to implement and deploy MPTCP. On the other hand, selecting appropriate paths from among possible paths would become more important. In this paper, we propose a design of data transfer system that utilizes MPTCP in OpenFlow environment. The key point in the proposed design is to use throughput prediction as a criterion when selecting appropriate paths. For maximizing throughput of MPTCP, the proposed system tries to select paths that are likely to achieve highest throughput according to results of throughput prediction. We also show our prototype implementation of the system and the result of its operability check.

Keywords: Software-Defined Network, OpenFlow, MPTCP, Throughput Prediction

1. はじめに

企業や自治体のサーバにはサービスの運用に必要である貴重なデータが保存されている。それらのデータを保護し

てサービスの運用を継続させるためには、災害などの被害を抑えたり早急に復旧するための予防措置であるディザスタリカバリが重要である。

ディザスタリカバリの手法として、広域環境でのレプリケーションがあげられる。データ保護手法であるレプリケーションでは、メインのシステムでデータが更新されるたびにネットワークを通じてバックアップシステムに複製を作成することで、ほぼ最新のデータを複製する。しかし、広域環境のレプリケーションではネットワークがボ

¹ 筑波大学 システム情報工学研究科
University of Tsukuba
Graduate School of Systems and Information Engineering

² 筑波大学 システム情報系
University of Tsukuba
Faculty of Engineering, Information and Systems

トルネックとなり、データの複製の頻度が低下する。それによって、ディザスタリカバリの重要な指標である障害時にどのくらい最新のデータを復旧させるかを示す RPO (Recovery Point Objective) が大きくなる。つまり、複製しているデータの“最新度”が下がり、障害時に失われるデータが増大する。したがって、広域でのレプリケーションではネットワークのスループット向上が重要な課題となっている。

ネットワークのスループットを向上させる手法としてマルチパス転送がある。マルチパス転送は、複数の経路を用いて転送を行う手法であり、リソースの利用効率や冗長性の向上によりスループットが向上する。TCP 転送をマルチパスで行う MPTCP 転送の実現方法として、Linux Kernel 実装 [1] が行われている。しかし、Linux Kernel 実装では、経路ごとに異なるゲートウェイを通るマルチホーム環境でなければならず、また、送受信ノードの両方で対応している必要があり、実用にはハードルが高い。

これに対し、より容易で効率的な MPTCP 転送を実現する手法として、Software-Defined Network(SDN)を用いた手法があげられる。SDN はソフトウェアによってネットワークを制御を行うというコンセプトであり、近年注目を集めている。従来のネットワークでは、ネットワーク機器が自律分散的に転送処理を行っていたため自由に経路を選択することが困難であったのに対し、SDN ではソフトウェアによって転送制御を行うため自由に転送経路を選択することができ、パケットを複数の経路に分散させるようプログラムすることで MPTCP 転送を実現することができる。したがって、ネットワークを SDN に置き換えることでノードに対する変更やマルチホームである必要が無くなり、より容易で MPTCP 転送を実現することが出来る。また、SDN では自由に経路選択が可能であることはこれまでのネットワークよりも効果的な経路選択が可能となるが、同時に膨大な経路の中から適切な転送経路の選択が必要となり、そのための手法が重要となる。

本研究では、SDN による MPTCP 転送の経路選択の際にスループット予測を用いる手法を提案する。スループット予測は、ネットワークの動的、静的な情報からスループットを予測するものであり、グリッドでのタスクスケジューリングや、マルチパス転送でのパス選択などの効率的なデータ転送に用いることができるため重要な研究課題である。ネットワークのスループット予測を経路選択の際の指標として用いることで、スループットが最大となるであろう経路を選択することができ、それによってスループット向上が期待できる。具体的なアプローチとして、OpenFlow[2] 環境で、SVR (Support Vector Regression) によるスループット予測手法 [3] を経路選択の指標として用いる MPTCP 転送を実現する。

本稿では、まず、提案手法の前提技術である OpenFlow,

MPTCP 転送、スループット予測について説明する。そして、提案手法の詳細や実装方法、予定している実験について説明し、最後にまとめと今後の課題について述べる。

2. OpenFlow

OpenFlow[2] は、SDN のコンセプトを実現する手法の一つであり、ネットワーク研究者が既存の環境に影響をあたえずにキャンパスのネットワークで実験を行うためにスタンフォード大学で研究された手法である。OpenFlow には、ソフトウェアによる自由度の高いネットワーク制御、システムソフトウェアと連携してネットワーク設定の自動化が可能、アジャイルやテスト、バージョン管理などのソフトウェア開発の手法を用いることができるといった利点があり、近年データセンターやクラウドインフラ関係で特に注目を集めている。

OpenFlow は、ネットワーク制御を行う OpenFlow コントローラと呼ばれるソフトウェア、データ転送を行う OpenFlow スイッチ、OpenFlow コントローラと OpenFlow スイッチとの通信を規定する OpenFlow プロトコルの 3 つの構成要素からなる (図 1)。OpenFlow では、転送処理を行う OpenFlow スイッチを OpenFlow コントローラで制御することによってネットワークを制御する。また、フローテーブルはデータの転送ルールを記述するものである。

従来のネットワークスイッチでは制御のためのファームウェアが転送処理を行うスイッチと一体になっており (図 2)、各スイッチが自立分散制御を行っていたのに対し、OpenFlow では、OpenFlow スイッチと OpenFlow コントローラによって制御と転送が分離しており、OpenFlow コントローラによる中央集中制御となっている。

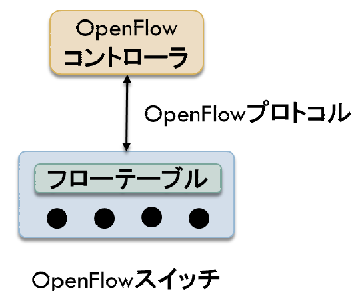


図 1 OpenFlow 概要図

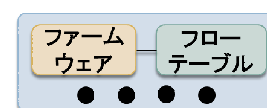


図 2 従来のスイッチ

OpenFlow の転送処理では、まず、OpenFlow スイッチは到着したパケットを OpenFlow コントローラへ送り、OpenFlow コントローラはプログラムに従って OpenFlow

スイッチへ転送ルールのインストール等の処理を行う。転送ルールにはどのパケットをどう処理するかが規定されており、OpenFlow スwitchは転送ルールに適合するパケットは OpenFlow コントローラへ送らずにルールに規定された処理を行い、どう処理したら良いかわからないパケットのみ OpenFlow コントローラへ送る。これによって転送ルールがインストールされているパケットの処理は従来のものとほぼ同等のパフォーマンスを得ることができる。また、転送ルールはスイッチのポート番号や、L4 までのヘッダ情報を組み合わせて記述することができる自由度の高いものとなっており、これらの仕組みによって、高速で柔軟性の高いネットワークを実現することができる。

3. スループット予測

静的、動的なネットワーク情報を用いることでネットワークのスループットの予測を行うスループット予測は、グリッドでのタスクスケジューリングやマルチパス転送でのパス選択などの効率的なデータ転送に用いることができるため、重要な課題である。

特に、TCP 転送のスループット予測の手法として、動的なネットワーク情報を用いた手法として Wolski の手法が提案されている [4]。Wolski の手法では、小さいデータである probe と大きい転送データのペアのスループットから予測モデルを作成し、probe のスループットから実際の転送データのスループットを予測する。予測モデルとして、累積分布関数 (CDF : Cumulative Distribution Function) を用いた予測手法が提案されている [5]。さらに、近年 CDF での手法よりも優れた手法として SVR (Support Vector Regression) を用いた手法が提案された [3]。SVR とは SVM (Support Vector Machine) を用いて回帰問題を解く手法であり、スループット予測においては、probe のスループットとデータのスループットのペアを用いて学習を行う。図 3 は CDF を用いた手法と SVR を用いた手法の実験結果であり、SVR を用いた手法の方がより予測性能が高いことが示されている。

したがって、本研究ではスループット予測の際に SVR を用いたスループット予測の手法を用いる。

4. MPTCP 転送

MPTCP 転送は、TCP 転送の高速化のために Internet Engineering Task Force(IETF) によって改良が進められている手法であり、TCP 転送を複数の経路を用いて行うことでリソースの利用効率や冗長性が増しスループットが向上する。MPTCP 転送の実装として Linux Kernel 実装 [1] が行われているが、特に最近では iOS7 の Siri が MPTCP 転送に対応しており、初めて実用化されるなど注目を集めている。

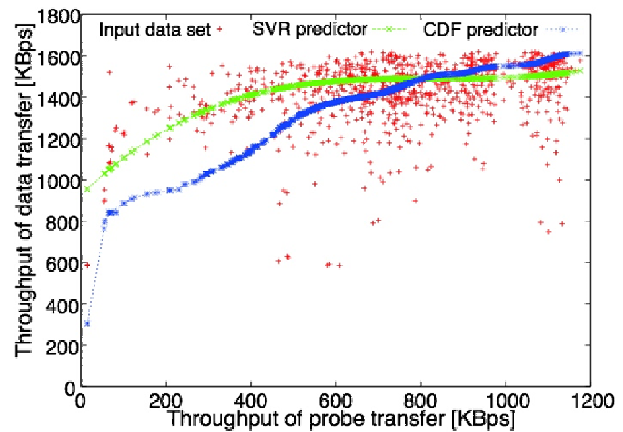


図 3 CDF 手法と SVR 手法の比較 [3]

4.1 Linux Kernel 実装

Linux Kernel 実装では、TCP に対し大まかに以下のような変更を加えて MPTCP 転送を実現している。

- (1) コネクション全体でどのノードと通信しているかを特定するためにノードごとに決められた ID をオプションとして付与する。
- (2) DPI (Deep Packet Inspection) のフィルタリングをパスするため、各コネクションごとの seq/ack とは別に全コネクションで共通の Data seq/ack をオプションとして付与する。
- (3) ある経路が切断されたときに再送できない状態に陥ることを避けるため、全体で共通の Window サイズを用いる。

これらの変更から、全体として socket から上のレイヤーでは通常の TCP 転送の様に振る舞う。

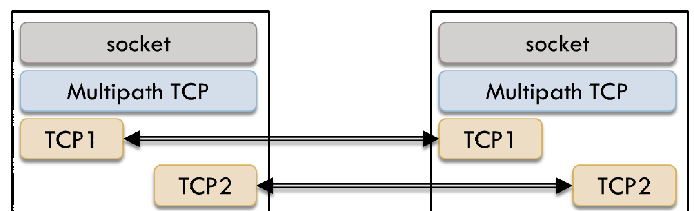


図 4 MPTCP 転送の Linux Kernel 実装

5. 提案手法

これまで述べてきたように、本研究では、OpenFlow ネットワーク環境で、SVR を用いたスループット予測手法での経路選択による MPTCP 転送を提案する。

図 5 が提案手法の概要図となる。提案手法は、経路選択に必要なトポロジ情報を定期的に収集する処理と、ホストからデータを送信する際にトポロジ情報とスループット予測からスループットが最大となる経路を選択し、OpenFlow スwitchへ転送ルールをインストールする二つの処理からなる。

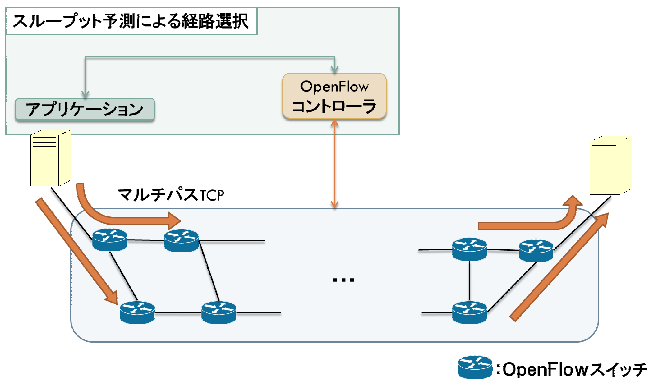


図 5 提案手法概要図

5.1 トポロジ情報の収集

経路選択に必要なトポロジ情報を定期的に収集する。OpenFlow コントローラは全ての OpenFlow スイッチに対し、自身と隣接接続している OpenFlow スイッチに LLDP (Link Layer Discovery Protocol) パケットを送るように指示する。パケットを受信した OpenFlow スイッチは、受信したパケットを OpenFlow コントローラに送る (packet-in)。OpenFlow コントローラに送られたパケットの送信元の OpenFlow スイッチの情報によって、ネットワークのトポロジがわかる。

5.2 スループット予測による経路選択

経路選択の処理では、送信ホスト上のアプリケーションと OpenFlow コントローラが通信し、以下の順にスループット予測による経路選択と OpenFlow スイッチへの転送ルールのインストールを行う。

- (1) 送信ホストは、データの転送を開始する前に宛先のアドレスと転送するデータの大きさを OpenFlow コントローラに伝える。
- (2) OpenFlow コントローラは、得られたトポロジ情報から送信ホストから宛先までの全ての経路を算出する。さらに、各経路毎に OpenFlow スイッチに転送ルールのインストールし、送信ホストに probe を送りスループットを返すように伝える。そして、送信ホストから集められた各経路毎の probe のスループットから事前に学習したモデルを用いてスループット予測を行い、スループットが最大となるように経路を選択し OpenFlow スイッチに転送ルールのインストールする。
- (3) 送信ノードは転送ルールのインストールの完了後、データの転送を行う。

5.3 実装

実装は、OpenFlow コントローラと送信ノードのアプリケーションに対して行う。

OpenFlow ネットワークでの MPTCP 転送は、送信ノ

ードから送られたパケットを OpenFlow コントローラによって複数の転送経路に分散して送るよう OpenFlow スイッチに転送ルールをインストールすることで実現できる。本研究では、OpenFlow のネットワーク上で Linux Kernel の実装を用いて MPTCP 転送を行う。

6. 動作検証

以下の環境に実装を行い、動作検証を行った。

- OpenFlow コントローラ : Trema v.0.4.5
- OpenFlow スイッチ : Open vSwitch v.1.4.0
- ホスト/ゲスト OS : Linux Mint 14
- MPTCP Linux Kernel Implementation v.0.86

OpenFlow コントローラは Ruby でプログラムを行うことができる Trema[6] を使用した。Open vSwitch[7] は仮想的な OpenFlow スイッチである。

ホスト OS 上に KVM で送受信のためのゲスト OS を 2 つ起動し、複数経路を持つように Open vSwitch で接続を行った環境で、ゲスト OS に MPTCP の Linux Kernel 実装を適用し、MPTCP での通信を行える事を確認した。

7. 関連研究

関連研究として、OpenFlow 環境で MPTCP 転送を実現した研究 [8] があげられる。この研究では、MPTCP 転送を実現するために OpenFlow v.1.1 で追加されたポートグループによるラウンドロビンでのロードバランスを用いている。本研究とは現在最も広く利用されている OpenFlow v.1.0 を対象としているという点でも異なるが、特に本研究では SDN の自由な経路選択が可能な特性を生かし、スループット予測による経路選択を用いている点で大きく異なっている。

また [9] では、ビッグデータの様な大きなデータの転送に耐えるために、実際に 10Gbps, 100Gbps の広域回線と 40GE NIC のサーバを使ってマーストリヒト、シカゴ間で OpenFlow ネットワークでの MPTCP 転送を行っている。この研究では転送に用いる経路は固定されており、実際に大きなデータの MPTCP 転送を行うことを目的としているため、適切な経路選択によってスループット向上を図る本研究とは異なる。

8. まとめと今後の課題

本研究では、ネットワークのスループット向上のために、OpenFlow での SVR を用いたスループット予測手法による経路選択での MPTCP 転送を提案した。OpenFlow コントローラと呼ばれるソフトウェアによってネットワークを制御する OpenFlow を用いることで自由度の高い経路選択が可能となり、複数の経路を用いて転送を行う MPTCP 転送によってスループットが向上する。また、経路選択の際にスループット予測を用いることでスループットが最大と

なる経路を選択できる。

今後は実装を進め、仮想スイッチの Open vSwitch、実機の OpenFlow スイッチ、広域の OpenFlow テストベッドの JGN-X RISE[10] で提案手法の性能の測定を行う。また、提案手法には以下の 2 つの問題が挙げられるためそれらの解決手法を考える必要がある。

- (1) 本稿で提案した手法では広域環境の OpenFlow ネットワークを想定しており、OpenFlow スイッチの数は十数程度と考えているため宛先までの全経路に対してスループロット予測を行っている。そのため、OpenFlow スイッチの数が多くなった場合に問題となる。
- (2) 送信する直前にトポロジ情報から宛先までの経路の算出や probe を送る処理を行う必要があり、長時間同じホストと通信する場合は問題とならないが、短時間の通信の場合は経路選択にかかる処理のオーバーヘッドが大きくなることが考えられるため、予測の精度を保ちつつ事前に計算を行うような手法を考える必要がある。

参考文献

- [1] MultiPath TCP - Linux Kernel implementation: <http://mptcp.info.ucl.ac.be/>.
- [2] McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S. and Turner, J.: OpenFlow: enabling innovation in campus networks, *SIGCOMM Comput. Commun. Rev.*, Vol. 38, No. 2, pp. 69–74 (2008).
- [3] Chungha Lee, Hirotake Abe, T. H. and Umemura, K.: Analytical Modeling of Network Throughput Prediction on the Internet., *IEICE TRANSACTIONS on Information and Systems*, Vol. E95-D, No. 12, pp. 2870–2878 (2012).
- [4] Wolski, R., Spring, N. T. and Hayes, J.: The network weather service: a distributed resource performance forecasting service for metacomputing, *Future Generation Computer Systems*, Vol. 15, No. 5-6, pp. 757–768 (1999).
- [5] Swamy, M. and Wolski, R.: Multivariate Resource Performance Forecasting in the Network Weather Service, *Supercomputing, ACM/IEEE 2002 Conference*, p. 11 (2002).
- [6] Trema: <http://trema.github.io/trema/>.
- [7] Open vSwitch: <http://openvswitch.org/>.
- [8] van der Pol, R., Boele, S., Dijkstra, F., Barczyk, A., van Malenstein, G., Chen, J. and Mambretti, J.: Multipathing with MPTCP and OpenFlow, *High Performance Computing, Networking, Storage and Analysis (SCC), 2012 SC Companion*, pp. 1617–1624 (2012).
- [9] van der Pol, R., Bredel, M., Barczyk, A., Overeinder, B., van Adrichem, N. and Kuipers, F.: Experiences with MPTCP in an intercontinental OpenFlow network, *TNC 2013* (2013).
- [10] JGN-X: <http://www.jgn.nict.go.jp/>.