SDN スイッチにおけるペイロードを利用したパケット処理の提案と実装

Proposal and Implementation of Packet Processing Using Payload on SDN Switch

長瀬 和也 †, 福田 浩章 ‡
Kazuya Nagase†, and Hiroyuki Fukuda‡
† 芝浦工業大学 工学部
†Faculty of Engineering, Shibaura Institute of Technology

要旨

Software Defined Network(SDN) とは, ネットワークをソフトウェアで制御する仕組みを指す. SDN を実現する技術として Open Flow がある. Open Flow スイッチでは Open Flow プロトコルの規定により, OSI 参照モデルにおけるレイヤ 4 以下しか指定できないため, HTTP プロトコルヘッダなど, ペイロード情報を利用したネットワーク制御はできない. そのため, レイヤ 5 以上の情報を参照するアプリケーションファイアウォール等を利用する場合には, 従来のネットワーク機器を使用する必要がある. この問題に対してスイッチ内部でパケットの再構成をしてペイロードの参照機構を提案する Packet Reassembly on SDN Switch(PRS) という先行研究がある. PRS の機構を利用してペイロードを利用したパケット処理をするには, 分割されたペイロードの再構成と, 再構成されたペイロードを利用したパケット処理が必要になるが, 現状の PRS では後者は考慮されていない. 本研究では PRS をさらに拡張し, 再構成されたペイロードを利用したパケット処理を実現する.

1. はじめに

クラウドサービスを一部でも利用している企業は, 2013 年の 33.1%から 2017 年では 56.9%と 4 年間で 23.1%上昇している [1]. 情報システムを企業設備内で完結して管理・運用するオンプレミスな環境で構築するのが当たり前であった数年前から, 現在ではクラウドサービスを活用したインフラストラクチャ構築は主流になりつつある. オンプレミスな環境で運用するネットワークでは, 基本的に一度設定したネットワークを基本的にそのまま利用するが, クラウドサービスではサービス利用者ごとに必要なときに必要な分だけ動的にリソースを割り振る必要があるため, ネットワーク構成が動的に変化する.

従来のネットワークの構築の例として, ネットワーク技術者がルータやブリッジなど個々のネットワーク機器に対し, 個別の設定を繰り返す方法が考えられる. しかし, この方法で動的に変化するクラウドサービスのネットワークを構成するのは, 処理にかかる時間・コストの両面から現実的でない.

このような課題に対しネットワークをソフトウェアから動的に制御する仕組みとして, Software Defined Network(SDN)[2] がある. SDN では従来のネットワーク機器で一つの機器に組み込まれていた, パケットを物理的に転送する機能とネットワーク経路を制御する機能を分離する.

SDN を実現する技術として Open Flow[3] がある. Open Flow スイッチでは Open Flow プロトコルの規定により, OSI 参照モデルにおけるレイヤ 4 以下しか指定できないため, HTTP プロトコルヘッダなど, ペイロード情報を利用したネットワーク制御はできない. そのため, レイヤ 5 以上の情報を参照するアプリケーションファイアウォール等を利用する場合には, 従来のネットワーク機器を使用する必要がある. この問題に対してスイッチ内部でパケットの再構成をしてペイロードの参照機構を提案する Packet Reassembly on SDN Switch(PRS)[4] という先行研究がある. PRS の機構を利用してペイロードを利用したパケット処理をするには, 分割されたペイロードの再構成と, 再構成されたペイロードを利用したパケット処理が必要になるが, 現状の PRS では後者は考慮されていない. 本研究では PRS をさらに拡張し, 再構成されたペイロードを利用したパケット処理を実現する.

以下, 2 節では SDN と Open Flow についての概要を述べ, 3 節では先行研究である Packet Reassembly on SDN Switch について, 4 節では Open Flow に対応したソフトウェアスイッチである Open vSwitch について, 5 節では本研究の提案について述べる.

2.SDN / Open Flow

2.1.SDN

Software Defined Network(SDN) とは, ネットワークをソフトウェアから動的に制御する仕組みである. 従来のネットワーク機器ではパケットを物理的に転送する機能とネットワーク経路を制御する機能

が一つの機器に組み込まれていた. SDN ではこの 2 つの機能を, ネットワークの経路を制御するコントロール部分とパケットを転送するインフラストラクチャ部分に分離する. コントロール部分はソフトウェアを用いて経路を制御し, 決定されたパケット転送のルールを一元的にインフラストラクチャ部分に伝える. このアーキテクチャにより SDN では設定後のネットワーク経路の変更など, 動的なネットワークの変更を実現する.

2.2.Open Flow

Open Flow とは SDN を実現する技術である. Open Flow ではデータの経路を汎用化するために図 1 に示した「Flow Table」が用いられる.

Match Field	Priority	Counters	Instruction	Timeouts	Cookie
送信元IP アドレス			Forward		
送信先MACアドレス			Drop		
⋮			⋮		

図 1: Flow Table

「Flow Table」には「Match Field(条件)」に対する「Instruction(アクション)」を登録する. Match Field には物理ポートや送信元 IP アドレス等のパケットを処理する条件を記述する. Instruction では Match Field に対する具体的な命令として, 物理ポート 0 番から出力, 送信先 MAC アドレスを更新などの具体的なアクションを記述する. 「Flow Table」は, 上記の 2 つに加えて, 柔軟な制御のためのオプションとして「Priority(優先度)」, 「Counters(統計情報)」, 「Timeouts(保持時間)」, 「Cookie(コントローラの使用する情報)」これらを 4 つ加えた要素で構成され, これら一つの組み合わせはフローエントリと呼ばれる. 「Open Flow」の特徴の一つは柔軟に制御可能な「Flow Table」を用いて, 機器ごとに異なっていたデータの流れを汎用的なものに規定していることである.

そして Open Flow では SDN のアーキテクチャに従い, パケットの制御機能と転送機能を分離する. ネットワークの経路制御機能はトポロジから切り離し「Open Flow コントローラ」へと集約される. そして, 実際にパケット転送の役割をする「Open Flow スイッチ」へコントローラからの集中的な制御によりネットワークトポロジを構成する.

コントローラとスイッチの相互通信は「Open Flow プロトコル」により規定されている. パケットの処理は「Flow Table」で共通化されているので, Open Flow ではコントローラは機器ごとの差異を考えることなく指示を送ることができる.

3.Packet Reassembly on SDN Switch

3.1.Open Flow の問題点

Open Flow ではコントローラからスイッチに送られたフローエントリに従ってパケットを制御するが, マッチフィールドに条件として指定可能なのは OSI 参照モデルにおけるレイヤ 4 以下の情報である. そのため, HTTP プロトコルヘッダなど, ペイロード情報を利用したネットワーク制御はできない. そのため, レイヤ 5 以上の参照を目的としたアプリケーションファイアウォール等を利用する場合には, 従来のネットワーク機器を使用する必要がある. この場合, 従来の機器を用いた部分だけが SDN で制御できないため, SDN のソフトウェアを用いた柔軟な経路制御の妨げになる. 代替手段としてコントローラに逐次処理を移譲する方法があるが, この場合も関連するパケットを全てコントローラに対して問い合わせる必要があるため, ネットワークの遅延に繋がる.

3.2.Packet Reassembly on SDN Switch

3.1 節で述べた課題に対し, Packet Reassembly on SDN Switch(以下 PRS) という先行研究がある. PRS ではスイッチ内部でのペイロード参照を目的とした機構を提案している. TCP/IP を利用した通信では, 一般にペイロードは IP セグメンテーション, および TCP セグメンテーションによって分割されて

ネットワークを流れている．PRS では Open Flow スイッチの内部で分割されたペイロードを一度保持し, 全てのパケットの到着後に個々のヘッダ情報からペイロードを再構成する．この手法を用いることでスイッチ内部に完結した処理でペイロードに含まれる OSI 参照モデル 5 以上の情報を参照することが可能になる．

4.Open vSwitch

Oepn vSwitch は Open Flow に対応したソフトウェアスイッチである．

Open vSwitch の Flow Table から合致するフローエントリを探索する手順を次に述べる．コントーラから受信したフローエントリは最初にユーザモジュールの Flow Table に登録される．コントローラから Open vSwitch にパケットが到着した場合には, カーネルモジュールの Flow Table から探索する．存在しない場合はカーネルモジュールはパケットをユーザモジュールへと転送し, 存在が確認された場合には, そのフローエントリをカーネルモジュールへキャッシュして登録する．カーネルモジュール, ユーザモジュールどちらにも存在しない場合はパケットをカプセル化してコントローラへ転送する．以上が Open vSwitch の基本的な Match Field からフローエントリの探索になる．条件に当てはまるフローの存在を確認すると, フローエントリに従いパケット処理命令が実行される．

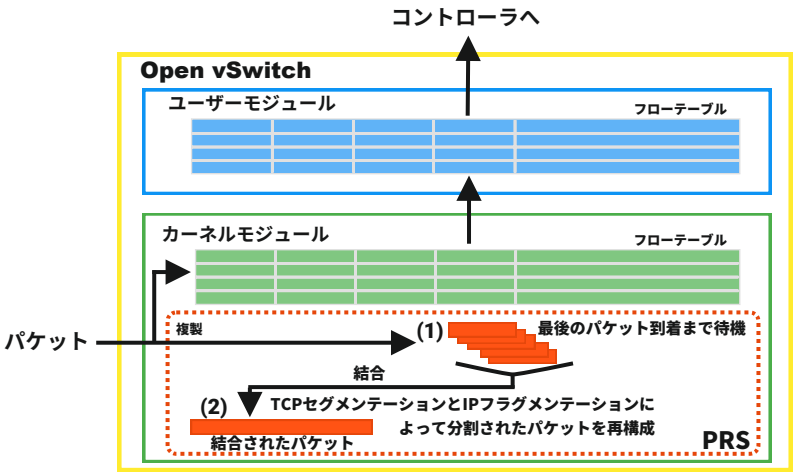

図 2: PRS のアーキテクチャ

図 2 に示すように PRS は Open vSwitch のカーネルモジュールを拡張し, スイッチ内部でのパケット結合処理を実現している．これは逐一ユーザ空間にパケットを転送する処理による遅延を防ぐためである．

PRS の機構を利用したペイロード参照にしたパケット処理には 2 つの機能が必要になる．

1. 分割されたペイロードの再構成
2. 再構成されたペイロードを利用したパケット処理

しかし, 現状の PRS では分割されたパケットの再構成は実装されているものの, ペイロードを参照したパケット処理を実現する機構は存在しない．そこで本研究では PRS を拡張し, ペイロードを参照したパケット処理の提案と実装を行う．

5. ペイロードを利用したパケット処理の提案と実装

Flow Table を拡張してペイロードを利用したパケット処理を実現する．本研究で提案する PRS を拡張したアーキテクチャを図 3 に示す．

5.1.PRS Table

図 3(1) に示す PRS Table には通常のフローエントリに加えて, OSI 参照モデルにおけるレイヤ 5 以上の Match Field(以下, PRS Match Field) と, それに対するアクション (以下, PRS Institute) の要素を登録する列が追加されている．PRS Table を利用しておこなうのは

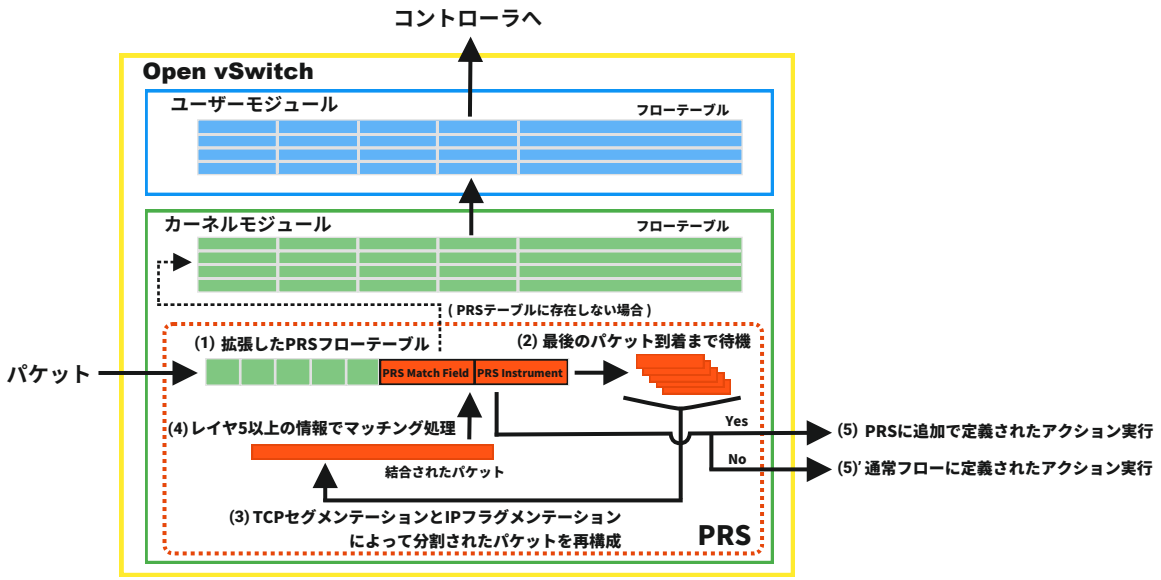

図 3: 本研究で提案するアーキテクチャ

1. スイッチに流入したパケットに対する結合処理の有無の振り分け
2. ペイロードの検査, と対応するアクションの実行

である.

5.2. ペイロードを利用したパケット処理の流れ

次に提案するペイロードを利用したパケット処理の流れを記述する. (1) では先ほど述べたように PRS の処理の有無を振り分けている. PRS Table の通常のフローエントリの部分を参照し, Match Field に合致した場合は PRS の処理へ, そうでない場合は通常のカーネルモジュールのテーブルに処理を流す. (2), (3) ではすでに PRS に実装されているパケット結合処理を行う. IP フラグメンテーションと TCP セグメンテーションを分割されたパケットの再構成をおこなうことで, ペイロードの参照な可能な状態にする. (4) では, (1) でマッチしたフローエントリを再び参照し, 追加した PRS Match Field でペイロードを利用したマッチング処理をおこなう. 最後の (5) では, PRS Match Field のマッチの有無で処理が別れる. マッチした場合には, 追加した PRS Institut が実行され, そうでない場合には PRS Table のフローエントリにある通常の Institute が実行される.

参考文献

[1] 総務省 (2018), “平成 30 年版 情報通信白書 | 企業におけるクラウドサービスの利用動向, ” , http://www.soumu.go.jp/johotsusintokei/whitepaper/ja/h30/html/nd252140.html, ”, 2018 年 11 月 12 日アクセス,

[2] OPEN NETWORKING FOUNDATION, “Software-DefinedNetworking: The New Norm for Networks White Paper, ”, April 13, 2012,

[3] OPEN NETWORKING FOUNDATION(2016), “Open Datapath - Open Networking Foundation, ” , https://www.opennetworking.org/technical-communities/areas/specification/open-datapath/, ”, 2018 年 11 月 12 日アクセス,

[4] 小嶋奨, 福田浩章, “PRS:SDN スイッチでの ペイロード検査機構, ” SIG Technical Reports, 2018/02/20.