

SDN 技術理解のためのアンプラグドな演習の設計

Design of Unplugged Exercise for SDN Technology

新村 正明^{*1}, 野瀬 裕昭^{*2}, 國宗 永佳^{*3}
Masaaki NIIMURA^{*1}, Hiroaki NOSE^{*2}, Hisayoshi KUNIMUNE^{*2}

^{*1}信州大学

^{*1}Shinshu University

^{*2}長野県工科短期大学校

^{*2}Nagano Prefectural Institute of Technology

^{*3}千葉工業大学

^{*3}Chiba Institute of Technology

Email: niimura@cs.shinshu-u.ac.jp

あらまし: 新しいネットワーク技術である SDN(Software Defined Networking)技術では, パケットヘッダによる条件処理が中心であることから, 制御ソフトウェアにおいてデータフロー的な要素を考慮する必要がある. 本研究では, パケットをデータフローと扱うことを体験的に学ぶため, SDN によるパケット制御を人間が処理するアンプラグドな演習を設計・実践した.

キーワード: 演習, アンプラグド, ネットワーク技術, SDN

1. はじめに

SDN(Software Defined Networking)技術は, 2008 年頃に提唱された新しいネットワーク技術である. 従来のネットワーク技術が, OSI 参照モデルに代表されるよう, 通信機能を階層構造に分割して実装していたのに対し, ネットワークのトラフィックをすべてソフトウェアで制御する技術であり, Google のバックエンドネットワークの制御に用いられる⁽¹⁾など, 実ネットワークへの導入も進んでいる.

我々も, SDN 技術習得のための演習システム⁽¹⁾を開発し, 実際の授業への展開を行ってきた. しかし, SDN は, 従来のネットワーク技術とソフトウェア技術が融合した技術であり, その理解には, 2つの技術の違いについて理解する必要がある.

そこで, SDN の制御ソフトウェア開発を行う前段階として, ソフトウェアによる制御と, パケット転送がどのように分離されているかを理解するための演習の設計を行った. 今回は, SDN 技術に実装例である OpenFlow によるパケット制御を対象とした.

2. OpenFlow の学習における問題点

2.1 OpenFlow の動作

OpenFlow によるネットワーク制御は, コントローラとスイッチの2つの機能により行われる.

まずスイッチは, 従来のネットワーク技術におけるスイッチが, 各々の通信階層における制御に基づいてパケット転送を行っていたのに対し, OpenFlow のスイッチは, パケットヘッダにおける部分一致による条件設定(Match)と, 条件に一致した際に行う動作(Action)及び優先順位からなるフローエントリを複数持つ. この複数のエントリを保持するものをフローテーブルと呼ぶ. スイッチは, 受信したパケットのヘッダに一致する Match を持つフローエントリ

りを優先順位順に検索し, 最初に一致したエントリの Action を実行するという方法でパケット転送を行う.

フローテーブルに一致するエントリがない場合には, そのパケットをコントローラに送り, コントローラは制御ソフトに従って新たなフローエントリを生成し, スイッチに送信する.

2.2 問題点

前節で述べたとおり, スイッチ内における処理は, 指定された条件に一致した場合にパケットに対する処理を行うものであり, データフローとして処理を考案する必要がある. これに対し, コントローラでは, 制御構造をもつソフトウェアによりフローエントリが生成される.

このように, Openflow におけるパケット処理は, 「制御構造をもつソフトウェアによるフローエントリ生成」と「フローエントリ群によるデータフロー的なパケット処理」の, パラダイムの異なる2つの処理方式により実現されている.

このため, 学習者が作成するソフトウェアは, 制御構造を持つプログラミング言語により作成されるが, このソフトウェアはフローエントリの生成のみで実際のパケットの制御は行わないことから, 作成するソフトウェアと実際のパケット処理に乖離が生ずることになる.

3. 演習設計

3.1 方針

問題点で述べたとおり, コントローラのソフトウェアがパケットを直接操作するのではなく, フローテーブル経由での制御となる. しかし, 演習においてはコントローラのソフトウェア開発が中心となりパケットの処理をイメージすることが困難である.

そこで、まずパケットの流れに着目させる演習を検討した。このような演習のためのパケットの可視化については先行研究があるが、従来のネットワーク技術が対象であることや、SDN 向けに改変するにはコストがかかることから、実現が容易で、かつ、体験的に学ぶことが可能なアンプラグド教材として設計することとした。

3.2 演習構成

演習は、図1に示すようなループ状のスイッチ接続をもつネットワークを想定する。従来のネットワーク技術では実現困難であり、SDN 技術の理解に適していると判断した。また、各機能の分離が明確になるよう、少なくとも1人が1つのスイッチを担当し、コントローラ役1名、端末役2-3名を含めて7名程度でのグループワークとする。

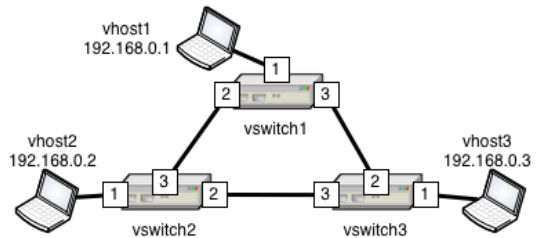


図1 演習対象のネットワーク

3.3 端末

演習は、端末役の学習者間で ping を打つ場合のパケットの流れを再現することとする。ping では ARP 要求, ARP 応答, ICMP echo 要求, ICMP echo 応答の4パケットのやり取りがなされる。そこで、図2に示すような4パケットのヘッダ部だけ記入するテンプレートを用意し、このパケットシートのやり取りで、パケットの流れを体験することとした。

ARP 要求		ARP 応答	
送信先MAC : destination mac	ff:ff:ff	送信先MAC : destination mac	
送信元MAC : source mac		送信元MAC : source mac	
送信先 IP : destination ip		送信先 IP : destination ip	
送信元IP : source ip		送信元IP : source ip	
IP 192.168.0.□ を使っているのは？		IP 192.168.0.□ を使っています	

ICMP 要求		ICMP 応答	
送信先MAC : destination mac		送信先MAC : destination mac	
送信元MAC : source mac		送信元MAC : source mac	
送信先 IP : destination ip		送信先 IP : destination ip	
送信元IP : source ip		送信元IP : source ip	

図2 パケットシート

3.4 スイッチ

スイッチ役の学習者は、図3に示すようなフローテーブルシートを保持し、端末役から受け取ったパケットシートを、フローテーブルに従って処理する。また該当するフローエントリがない場合、後述のコントローラにパケットシートを送り、指示されたフローエントリを手書きで記入する。

Priority	Match	Action
	source_mac_address : destination_mac_address : in_port :	
	source_mac_address : destination_mac_address : in_port :	
	source_mac_address : destination_mac_address : in_port :	
	source_mac_address : destination_mac_address : in_port :	

図3 フローテーブルシート

3.5 コントローラ

今回は、パケットの流れに着目することを目的とし、制御ソフトウェアはこちらで提供することとする。そこで、コントローラ役の学習者には、図4に示す、コントローラのソフトウェアを示したシートを配布する。これには、スイッチ役の学習者への指示が容易になるよう、図3のフローテーブルシートと同様な形式で制御ソフトウェアが記述されている。

条件	Priority	Match	Action
message_source_mac : message_destination_mac : in_port :	Switch Ready	source_mac_address : destination_mac_address : in_port : 3	Port 1 へ Port 2 へ
message_source_mac : message_destination_mac : in_port :	1	source_mac_address : destination_mac_address : in_port : 3	Drop
	400	source_mac_address : destination_mac_address : in_port : 1	Port 2 へ

図4 コントローラシート

4. 試行

本提案の有効性を確認するために、長野県工科短期大学の演習にて提案手法の試行を行った。

演習の結果、全グループで、ping を打つ場合のパケットの流れが再現でき、フローテーブルの情報も正しく記載されたことから、SDN の動作原理については、概ね、理解が進んだと思われる。

しかし、最初にコントローラの制御ソフトの解説をしたことから、スイッチ役の学習者がコントローラの指示なくフローエントリを記入してしまうことや、グループワークとしたため、端末役とスイッチ役が相談しながら処理を進めてしまい、役割分担が不明瞭になってしまう点が見受けられた。

5. まとめ

本提案により、SDN 技術の動作原理の理解が進むであろうことが確認できた。今後は、パケット制御部分だけの演習にとどまっていることや、役割分担が不明瞭である等の問題点の解決を図っていく。

参考文献

- (1) B Koley: “Software Defined Networking at Scale”, Google Research, (2014)
<https://research.google.com/pubs/archive/42948.pdf>
- (2) 百瀬拓也, 横山貴志, 國宗永佳, 新村正明: “Web ベース統合開発環境による SDN 実習支援システムの提案”, 電子情報通信学会技術研究報告(教育工学), 115, 492, pp.7-10, (2016)
- (3) 大脇裕也, 立岩佑一郎, 片山喜章, 高橋直久: “ネットワーク学習者支援のためのパケット可視化システムについて”, 電子情報通信学会総合大会講演論文集, 2, 82 (2015)