프로그래머블 데이터평면 연구 동향

장석원, 백호성, 한솔, 김영호, 백상헌, 임용재
고려대학교 전기전자공학과
e-mail : { imsoboy2, gh1emd, hs1087, drkim05, shpack} @korea.ac.kr , yongjae.rim@gmail.com

A Study on Programmable Data Plane

Seokwon Jang, Hosung Baek, Sol Han, Youngho Kim, Sangheon Pack, Yongjae Rim
School of Electrical Engineering, Korea University

요 약

소프트웨어 정의 네트워킹 기술은 OpenFlow 프로토콜을 중심으로 제어평면과 전달평면의 인터페이스에 대한 연구 개발을 진행해왔다. 하지만, 새로운 헤더 포맷 또는 프로토콜에 대한 처리를 위해서는 OpenFlow Specification 의 버전이 갱신되어야 하는 문제점이 있었다. 이에 대해 트래픽 처리가 특정 포맷에 한정되지 않도록 하기 위해, 네트워크 장비를 프로그래밍 하여 네트워크 관리자가 원하는 방식으로 트래픽을 처리할 수 있도록 하는 네트워크 장비 프로그래밍에 대한 연구가 활발하다. 본 논문에서는 데이터평면 프로그래밍에 대한 개략적인 연구 동향을 살펴본다.

1. 서론

소프트웨어 정의 네트워킹(SDN: Software Defined Networking)은 네트워크 운영자에게 네트워크를 프로그램적으로 컨트롤 할 수 있는 능력을 주는 차세대 네트워킹 기술이다. SDN 에서는 제어평면과 데이터평면이 물리적으로 분리되어 있으며, 하나의 제어평면은 데이터평면의 여러 네트워크 장비들을 관리하게 된다. 한편, 데이터 평면의 네트워크 장비들은 제어평면과의 연결을 OpenFlow 와 같은 공통되고 vendor-agnostic 한 특성의 인터페이스를 이용하여 구축한다. 이를 통해, 서로 다른 제조사의 물리적 장비나 소프트웨어들이 공존하는 데이터평면에 대해 하나의 제어평면이 일관적으로 관리를 할 수 있게 된다. OpenFlow 의 인터페이스는 초창기 Ethernet, TCP/IPv4 정도의 간단한 헤더포맷에서 최근까지 40 개 이상의 헤더포맷을 지원하도록 발전해왔다. 하지만, STT 또는 NVGRE 등의 packet encapsulation 에 대한 새로운 포맷이 요구됨에 따라, 요구에 맞게 OpenFlow 의 헤더포맷을 늘리는 방식은 유연하지 못하며, specification 이 갱신 될 때까지 기다려야 한다는 단점이 존재했다. 이에 대한 대응 방안으로 네트워크 장비를 프로그램적으로 재 설정할 수 있도록 변화시켜 원하는 헤더포맷을 추출하고 추출한 헤더를 처리할 수 있도록 하는 Programmable Data Plane 연구를 진행하고 있다. Programmable Data plane 연구는 다음과 같이 3 가지 사항에 대해 고려한다. 1) 프로그래밍이 가능한 네트워크 장비. 2) 네트워크 장비의 low-level interface 에 대한 high-level abstraction 을 제공할 프로그래밍 언어. 3) 프로그래밍 언어를 네트워크 장비에 매핑하는 컴파일러.

본 논문에서는 위의 세가지 사항에 대한 관련 연구 동향에 대하여 설명한다. 2 장에서는 프로그래밍이 가능한 네트워크 장비에 대한 연구를 설명하고 3 장에서는 프로그래밍 언어와 컴파일러에 대해 개략적으로 설명을 한다.

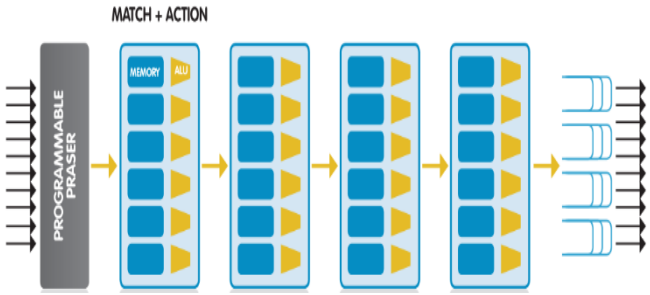

그림 1. Barefoot Networks 의 PISA[1]

2. 프로그래밍 가능한 네트워크 장비

데이터평면을 프로그래밍하기 위해서는 데이터평면을 구성하고 있는 네트워크 장비들이 프로그램적으로 재설정이 가능하여야 한다. 이에 대한 네트워크 장비 모델은, Barefoot Networks 의 Protocol Independent Switch Architecture(PISA)[1]와 Intel 의 FlexFipe[2]가 대표적이다. 최근까지 개발중인 이러한 프로그래밍 가능한 네트워크 장비는 fixed-function switch 와 동급의 성능을 지원함과 동시에 네트워크 장비자체의 프로그래밍 인터페이스를 지원하고 있다. 그림 1 은 Barefoot Networks 의 PISA 이다. PISA[1]에서는 P4[3]언어를 이용하여 개발자에게 Memory/ALU 를 통한 packet parser 와 table match/action 에 대한 인터페이스를 제공하여 packet-processing 에 대한 하드웨어 상에 프로그래밍 능력을 지원하고 있다. 한편, 위에서 언급한 것과 같은 대부분의 프로그래밍이 가능한 네트워크 장비들에 대한 연구는 packet parser 와 memory match/action 에

<표 1> 각 프로그래밍 언어의 특징

	P4	Domino
Syntax & Semantics	Field definition / packet processing	C-like syntax / packet transaction
Processing Field	Header field	Header field
Field Reconfigurability	Support	Support
Stateful Processing	Support (lower performance)	Support (higher performance)
Target Device	Switch, middlebox, router	Banzai machine model
Target applications	Flow-driven SDN service	Active queue management, stateful data plane algorithms

대한 인터페이스만 제공하며, instruction set architecture 등의 machine model 에 대한 인터페이스는 제공하지 않는다. 따라서, Active Queue Management(AQM) 등의 stateful 데이터평면 알고리즘을 해당 장비에 프로그래밍하기에는 한계가 있다는 단점이 있다. 이에 대한 보완으로, Domino[4]가 제안되었다. Domino[4]는 데이터평면 알고리즘에 대한 프로그래밍 능력을 제공해주기 위한 프레임워크로써, Atom 이라는 형식으로 스위치에 적합한 형태의 instruction set architecture 및 해당 구조가 설계된 machine model 인 Banzai switch 를 정의하여 스위치 장비의 line-rate 를 보장하며 stateful 데이터평면 알고리즘을 프로그래밍하기 위한 연구를 진행하고 있다.

3. 데이터평면 프로그래밍 언어와 컴파일러

본 장에서는 데이터평면 언어 중 가장 활발하게 연구가 진행되고 있는 P4[3]와 Domino[4] 언어에 대하여 알아보도록 한다. 표 1 은 두 언어의 개략적인 특징을 나타낸다.

먼저, P4[3]는 오픈 소스 커뮤니티를 기반으로 연구되고 있는 high-level 데이터평면 언어이다. P4[3]의 설계 목표는 프로토콜에 독립적임과 동시에 네트워크 장비에 대한 독립성을 얻는 것이다. P4[3]는 2 장에서 언급한 PISA[1]와 같은 프로그래밍 가능한 장비가 제공하는 프로그래밍 인터페이스인 memory/ALU 에 대해서 header parsing field 와 table match/action filed 를 P4-specification[5]에 맞게 정의하여 프로그래밍하는 과정을 거친다. 이를 통해 개발자가 원하는 헤더 포맷을 처리할 수 있도록 하여 프로토콜로부터 독립성을 얻고 있다. 또한, 병렬처리를 위한 pipelining, table 간의 종속성 계산 그리고 프로그래밍 가능한 장비의 구조적 모델에 대한 구체적 특성 등의 복잡한 연산처리 과정은 컴파일러에서 처리함으로써, 언어자체로는 네트워크 장비에 대한 독립성을 얻고 있다. 하지만, 제한적일 수 있는 네트워크 장비에 대한 자세한 정보가 필요하며, 복잡한 연산 처리의 필요성으로 컴파일러의 설계과정이 코드개발에 비해 복잡해 지게 된다. 한편, P4[3]언어는 지속적인 언어의 개발로 스위치뿐만 아니라 firewall 이나 NIC 와 같은 모든 형태의 네트워크 장비에 대한 프로그래밍 언어로 자리잡는 것을 목표로 하고 있지만, P4[3]언어의 메모리 자원의 공유 범위를 각 테이블로 한정한다는 특성상 stateful 데이터평면 알고리즘 구현에는 한계가 있다. 이에 대한 솔루션이자 연구의 시발점이 될 수 있는 것이 Domino[4]이다.

Domino[4]는 가장 최근에 제안된 데이터평면 프로그래밍 프레임워크로써, P4[3]의 단점을 보완하고자 연구가 진행되고 있다. Domino[4]는 C 언어와 유사한 형태의 문법구조를 갖는 언어와 machine model 그리고 컴파일러를 제안하였으며, P4[3]와 마찬가지로 packet 의 header field 만을 추출하여 처리한다. Domino[4]의 설계 목표는 네트워크 디바이스의 line-rate 처리율을 보장하는 stateful 데이터평면 알고리즘에 대한 프로그래밍 능력 제공이다. 현재, Domino[4]는 대다수의 stateful 데이터평면 알고리즘에 대한 line-rate 처리율을 보장하는 프로그래밍 능력을 제공하지만 payload 를 탐색해야 하는 DPI 또는 CoDel[6]과 같은 큰 헤더 포맷을 갖는 알고리즘에 대해서는 제한적이다.

4. 결론

본 논문에서는 데이터평면을 프로그래밍하기 위한 프로그래밍 가능한 네트워크 장비와 프로그래밍 언어에 대한 연구 동향을 살펴보았다. 데이터평면 프로그래밍은 데이터평면의 기존 fixed-function 네트워크 장비의 기능과 성능을 완전히 대체하기 위해서는 좀 더 많은 연구가 지속적으로 이루어져야 한다. 하지만, 프로그래밍에 대한 가능성을 통해 다양한 애플리케이션들에 대한 연구가 진행되고 있는 만큼 연구 주제로 주목할 만하다.

5. ACKNOWLEGEMENT

본 연구는 미래창조과학부 및 정보통신기술연구진흥센터의 정보통신·방송 연구개발사업의 일환으로 수행하였음. [B0101-16-0233, 스마트 네트워킹 핵심 기술 개발]

참고문헌

[1] Protocol Independent Switch Architecture, Barefoot Networks.
[Online]. Available: https://barefootnetworks.com/white-paper/the-worlds-fastest-most-programmable-networks/

[2] Ozdag, R. Intel® Ethernet Switch FM6000 Series-Software Defined Networking. Intel Corporation 2012.

[3] P. Bosshart, D. Daly, G. Gibb, M. Izzard, N. McKeown, J. Rexford, C. Schlesinger, D. Talayco, A. Vahdat, G. Varghese and D. Walker, "P4: Programming protocol-independent packet processors," *SIGCOMM Computer Communication Review,* Vol. 44, No. 3, pp. 87–95, July 2014.

[4] A. Sivaraman, M. Budiu, A. Cheung, C. Kim, S. Licking, G. Varghese, H. Balakrishnan, M. Alizadeh, and N. McKeown, "Packet transactions: High-level programming for line-rate switches," in *Proc. SIGCOMM 2016*, to appear.

[5] P4 Specification. [Online]. Available: http://p4.org/wp-content/uploads/2016/03/p4_v1.1.pdf

[6] K. Nichols and V. Jacobson, "Controlling Queue Delay," *Communications of the ACM,* 2012.