

# 프로그래밍 가능한 데이터 평면의 연구 동향

김수연, 박태준\*

전남대학교(학부생), \*전남대학교(교수)

rlatndus6205@jnu.ac.kr, \*taejune.park@jnu.ac.kr

## A survey on Programmable Data Plane

Sooyeon Kim, Taejune Park\*

Chonnam National Univ., \*Chonnam National Univ.

### 요 약

현재 사용하고 있는 SDN(Software Defined Networking) 기술은 한 개의 제어 평면으로 여러 개의 데이터 평면을 관리할 수 있어 네트워크 운영을 보다 쉽게 할 수 있다. 그러나, 네트워크 장비의 기능이 고정되어 있다는 문제가 있어 이러한 문제를 해결할 수 있는 프로그래밍 가능한 데이터 평면이라는 방안이 제시되었다. 본 논문에서는 프로그래밍 가능한 데이터 평면이 무엇인지 알아보고 이것의 향후 연구 방향에 대해 알아보고자 한다. 서론에서는 프로그래밍 가능한 데이터 평면의 등장 배경에 대해 설명한다. 본론에서는 이에 필요한 장비/언어뿐만 아니라 이를 사용하는 분야와 실제로 사용하고 있는 예시에 대해 알아보고 마지막으로 결론에서는 아직 해결되지 않은 연구과제와 향후 연구 방향에 대해 살펴본다.

### I. 서 론

#### 1-1. 등장 배경

네트워크의 데이터 평면은 스위치, 라우터, 방화벽 등 여러 네트워크 장비로 구성되고 이러한 장비는 장비 공급업체에 따라 기능이 고정된다 [1]. 그러나, 앞으로 개발될 애플리케이션은 서로 다른 프로토콜이나 함수들의 집합으로 동작하며, 여러 동작을 지원하기 위해서는 VPN, 로드 밸런싱, IPS 등 다양한 기능을 가진 네트워크 장비가 필요하다. 허나, 이러한 다양한 네트워크 장비들은 각자 저마다의 관리 및 제어 프로토콜이 있어, 이들을 통합적으로 관리하는 것은 쉽지 않다.

이를 해결하기 위해 SDN(Software Defined Networking)을 활용하여 네트워크 기능을 프로그래밍 가능한 형태로 운영하는 방안이 제시되었다. SDN에서는 네트워크를 제어 평면과 데이터 평면으로 분리하고 하나의 제어 평면을 통해 여러 데이터 평면을 관리/제어할 수 있어 네트워크 전체 상황을 쉽게 파악해 네트워크 설계와 관리 및 운영을 보다 쉽게 할 수 있다 [2].

그러나, 데이터 평면의 기능이 고정되어 있다는 부분은 여전히 문제로 남아있어 네트워크 운영의 비효율성을 크게 개선하지는 못했다. 오늘날 네트워크는 가상화 환경, 사물인터넷(Internet of Things, IoT), 에지 컴퓨팅(Edge Computing) 등 네트워크 환경과 구성에 변화를 요구하는 다양한 기술들이 등장했고 이러한 기술들을 지원하기 위해서는 데이터 평면에 새로운 기능들을 빈번하게 추가되어야 한다. 허나, 일반적으로 하드웨어 기반으로 구성된 데이터 평면이 다양한 동작들을 효율적으로 수행하려면 시간 및 비용뿐만 아니라 공급업체와 엔지니어의 노력이 많이 필요하고 새로운 기능을 추가하기 위해서는 업데이트를 주기적으로 해줘야 한다. 이는 비효율성을 초래해 네트워크 구현에 큰 걸림돌이 된다 [3].

이러한 문제를 해결하기 위해 프로그래밍 가능한 데이터 평면(Programmable Data Plane, PDP)이 등장했다. PDP에서는 제어 평면에 변화가 생기면 데이터 평면에서 바로 프로그래밍을 해 즉시 대응이 가능하다. 데이터 평면에서 프로그래밍이 가능하려면 프로그래밍 가능한 네트워크 장비와 장비 수준의 프로그래밍 가능 언어가 필요하다.

본 논문에서는 프로그래밍 가능한 장비와 데이터 평면에서 사용하는 프로그래밍 언어, 그리고 PDP를 활용하는 분야와 실제 사례에 대해 설명한 후 향후 연구 방향을 살펴보고자 한다.

### II. 본론

PDP는 데이터 평면의 프로그래밍 언어를 통해 네트워크 장비에서 필요한 기능이나 규칙을 바로 설정할 수 있다. 이로 인해 데이터를 정교하게 처리해 전송할 수 있고 데이터 경로에 있는 네트워크 장비를 직접 설정할 수 있다. 이는 추가 데이터 이동을 최소화해 비용과 시간을 절약할 수 있다는 장점으로 이어져 현재 다양한 분야에서 활용되고 있다. 본 장에서는 프로그래밍 가능한 네트워크 장비와 장비 수준의 프로그래밍 언어, 그리고 어떠한 분야에서 PDP를 사용하는지 설명하고자 한다.

#### 2-1. 프로그래밍 가능한 네트워크 장비

네트워크 장비에는 칩이 들어가는데 기존의 네트워크 장비들은 공급업체에 따라서 칩의 기능이 고정되어 있고 내부에서 패킷을 처리하는 파이프라인은 런타임 도중에 쉽게 재구성할 수 없다. 새로운 기능을 추가하기 위해서는 칩을 새로 만들거나 파이프라인을 재구성해야 하는데 이 과정에서 시간과 비용이 많이 들어 비효율적이고 애플리케이션 동작에 필요한 유연성을 제공하지 못한다. 이를 해결하기 위해 네트워크 장비를 프로그래밍 가능하도록 해 오류가 있거나 새로운 기능을 추가할 필요가 있을 때 장비를 직접 수정함으로써 유연성을 제공하고 네트워크를 좀 더 유동적이고 효율적으로 운영할 수 있다.

이러한 네트워크 장비로는 FPGA, SmartNIC, PISA 등이 있다. FPGA(Field Programmable Gate Array)는 설계 가능 논리 소자가 포함된 네트워크 칩으로 AND, OR, XOR, NOT과 같이 기본적인 논리 연산이 가능해, 이를 이용해 프로그래밍이 가능한 하드웨어 장비이다. SmartNIC(Smart Network Interface Card)는 데이터 센터의 네트워킹, 보안, 스토리지를 효율적이고 유연하게 만드는 프로그래밍이 가능한 네트워크 인터페이스 가속기로 각종 소프트웨어 기반 네트워크 프로세싱에 대한 오프로딩(offloading)이 가능하다. PISA(Protocol Independent Switch Architecture)는 단일 파이프라인 포워딩 아키텍처로 프로그래밍 가능한 parser를 통해 프로그래밍 가능하다. FPGA는 게이트 수준에서의 네트워크 장비를 구성할 수 있고, SmartNIC, PISA로 갈수록 프로그래밍 언어를 통한 네트워크 구현이 더욱 용이하다.

## 2-2. 데이터 평면 프로그래밍 언어: P4

데이터 평면을 프로그래밍하기 위해서는 적절한 프로그래밍 언어가 필요한데 대표적으로 P4 [4,5]가 있다. P4는 프로토콜과 독립적인 패킷 언어로 FPGA, SmartNIC, PISA 등 다양한 네트워크 장비의 종류와도 독립적으로 사용 가능하다. P4는 프로그래밍 가능성, 프로토콜 및 장비로부터 완전한 독립성이라는 목표가 있다. 프로그래머는 P4를 통해 필요한 기능을 직접 프로그래밍하며 이는 IP, TCP와 같이 표준화되어 있는 프로토콜뿐만 아니라 사용자가 직접 정의한 프로토콜 또한 지원하고, 이렇게 정의한 헤더나 프로토콜은 스위치나 라우터와 같은 장비의 종류에 크게 구애받지 않고 실행할 수 있도록 P4를 개발하고 있다. 현재는 프로그래머가 작성한 헤더나 프로토콜을 매치-액션 테이블(match-action table)을 거쳐 프로토콜부터 독립성을 얻고 있고, 또한, 언어들 사이의 공통된 특성을 이용하고 컴파일에서 연산을 수행하는 것뿐만 아니라 파이프라인과 테이블 간의 종속성을 활용해 병렬처리를 함으로써 언어로부터의 독립성을 얻었다.

## 2-2. 활용

1) **모니터링, 원격 측정 및 진단:** 모니터링은 모니터링되는 트래픽을 중간 미들 박스로 복제한 후 측정 및 진단하는 방식으로 실행되는데 미들 박스로 트래픽을 복사하는 과정에서 비용이 많이 든다. 미들 박스 대신 PDP를 사용할 경우 효율성을 향상시킬 수 있고 뿐만 아니라 네트워크 운영자가 PDP를 통해 네트워크 쿼리를 직접 작성할 수 있어 광범위한 네트워크 모니터링이 가능하고 네트워크에 따른 맞춤형 분석이 가능하다.

2) **네트워크 내 계산:** 네트워크 내 계산은 데이터 센터의 대규모 네트워크 데이터 처리의 성능 병목 현상과 확장성 한계를 해결하는 방법 중 하나다. 네트워크 내 계산은 데이터를 처리하는 과정에서 데이터를 간단히 변환한 후 처리하므로 네트워크 트래픽 양을 줄일 수 있다. 또한, 복잡한 계산은 간단한 산술/논리 연산으로 바꿀 수 있기 때문에 프로그래밍 가능한 하드웨어에서 병렬화 및 실행이 가능하다. 이는 PDP를 사용한다면 계산을 훨씬 더 효율적으로 수행할 수 있다는 것을 의미한다.

3) **오프로딩(offloading):** 오프로딩은 소프트웨어를 기반으로 실행되는 애플리케이션의 일부를 다른 성능이 좋은 하드웨어에 전달해 실행한 후 결과를 반환받는 방식으로, CPU와 같은 하드웨어의 부담을 줄이고 성능을 높일 수 있다. PDP를 사용할 경우 미리 SmartNIC과 같은

네트워크 장비를 프로그래밍한 후 다른 하드웨어로부터 복잡한 일을 받아 빠르게 처리할 수 있어 성능을 향상시킬 수 있다.

이러한 활용성에 주목하여, Microsoft의 'AccelNet', META의 'FBOSS', 그리고 Amazon 'FPGA 기반 가속 서비스' 등 실제 여러 기업들이 자사의 네트워크를 PDP를 이용해 구축하고 있으며, 점차 더 많은 곳에서 PDP 사용할 것이라고 생각한다 [7,8,9].

## III. 결론

PDP를 사용한다는 것은 데이터가 이동하는 중간 경로에 네트워크 장비가 있으며 이를 프로그래밍할 수 있다는 것을 의미하고 결국 네트워크 운영에 드는 자원을 절약할 수 있다는 것을 의미한다. 그러나, 아직 제어 평면에서 데이터 평면을 간단히 제어하기 위한 추상화 문제나 패킷 처리 방식을 훨씬 효율적으로 프로그래밍하는 문제 등 아직 해결되지 않은 연구 과제들도 남아있다 [6]. 이를 해결하기 위해서는 PDP에 대한 지속적인 연구가 필요하고 만약 이러한 과제들이 해결된다면 네트워크 설계에 긍정적인 영향을 미칠 것으로 기대된다.

## ACKNOWLEDGMENT

이 성과는 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (No. 2022R1C1C1006967).

## 참 고 문 헌

- [1] Parol P, "P4 Network Programming Language-what is it all about?", 2020, (<https://codilime.com/blog/p4-network-programming-language-what-is-it-all-about/>)
- [2] 장석원, 백호성, 한솔, 김영호, 백상현, 임용재, "프로그래머블 데이터평면 연구 동향", *추계학술발표대회 논문집*, 23(2), 2016
- [3] Hauser F. et al., "A survey on data plane programming with p4: Fundamentals, advances, and applied research.", *Journal of Network and Computer Applications* (2022), 103561, December 2022
- [4] Bosshart P. et al., "P4: Programming Protocol-Independent Packet Processors", *SIGCOMM Computer Communication Reviews*, Vol.44, No.3, pp.88-95, July 2014
- [5] P4, "P4 Open Source Programming Language", (<https://p4.org/>)
- [6] Michel . et al., "The Programmable Data Plane: Abstractions, Architectures, Algorithms, and Applications", *ACM Computing Surveys*, Vol.54, No.4, pp.1-36, May 2022
- [7] Amazon EC2 F1 instance (<https://aws.amazon.com/ko/ec2/instance-types/f1/>)
- [8] Daniel F. et al., "Azure Accelerated Networking:{SmartNICs} in the Public Cloud", 15<sup>th</sup> USENIX Symposium on Networked Systems Design and Implementation (NSDI 18), 2018
- [9] Choi S. et al., "FBOSS: Building Switch Software at Scale", *SIGCOMM '18*, August 2018