

김도현 - An eBPF-XDP hardware-based network slicing architecture for future 6G front to back haul networks

소유자	김도현
태그	

원본

[An eBPF-XDP hardware-based network slicing architecture for future 6G front to back haul networks.pdf](#)

번역본

[An eBPF-XDP hardware-based network slicing architecture for future 6G front to back haul networks 번역본.docx](#)

요약

5G 와 6G 네트워크에서는 높은 데이터 전송 속도와 신뢰성, 그리고 유연한 네트워크 관리가 필수적이다. 이 논문에서는 **네트워크 슬라이싱**, 프로그래밍 가능한 **데이터 플레인**과 같은 기술을 활용한다. 또한 **eXpress Data Path(XDP)**, **extended Berkeley Packet Filter(eBPF)**, **smart network interface card(SmartNICs)**를 포함하는 하드웨어 기반 기술을 사용하여 네트워크의 성능과 효율성을 극대화하는 것에 대해 다룬다.

XDP와 eBPF는 데이터 플레인에서의 높은 처리 속도와 낮은 지연 시간을 가능하게 하며, 데이터 패킷의 실시간 처리와 분류를 개선한다. SmartNICs를 사용하여 네트워크 트래픽 처리를 네

트위크 카드로 오프로드함으로써 CPU 부하를 감소시키고, 전체 시스템의 성능을 향상시킨다. 또한, LINUX 커널의 오버헤드를 줄이고 네트워크 슬라이스를 효과적으로 제어하고 모니터링 할 수 있는 방법에 대해 소개한다.

6G 네트워크에서 네트워크 통신을 가속화하고 CPU에 부담을 주지 않고 시스템의 리소스를 최대화하여 네트워크 처리 작업을 최적화하는 것을 목표로 한다.

본문

▼ 커널 바이패스 기술이란?

커널 바이패스 기술은 컴퓨터에서 데이터를 처리할 때 운영 체제의 중심부의 커널을 우회하는 기술이다. 시스템 커널과 사용자 공간 애플리케이션 간의 인터페이스는 성능 병목 현상의 주요 원인인데 커널을 우회함으로써, 데이터 처리 과정에서 발생할 수 있는 지연을 크게 줄일 수 있다. 주요 목적으로는 고성능 네트워킹을 달성하기 위해 시스템 호출과 컨텍스트 스위치의 오버헤드를 최소화하는 것이다. DPDK, SNABBSWICHTH, NETMAP, PACKET MMAP, PF RING, EF VI, AF XDP라는 주요 기술이 있다.

DPDK (Data Plane Development kit) : 광범위한 CPU 아키텍처에서 사용할 수 있는 패킷 처리 작업을 가속화하기 위해 설계된 공개 소스 소프트웨어 라이브러리 모음으로, 사용자 공간에서 직접 네트워크 디바이스를 조작하여 커널 네트워크 스택을 우회한다.

SNABBSWITCH : L2 애플리케이션을 위해 설계되었으면 UIO를 활용하여 완전한 프레임워크를 제공한다.

NETMAP : 커널 모듈을 사용하여 네트워크 인터페이스와 통신하고, 필요한 경우 드라이버의 맞춤형 구성을 통해 네트워크 패킷 처리를 최적화한다.

PF_RING : 네트워크 소켓을 통해 고성능 데이터 캡처를 가능하게 하며, 시스템의 인터럽트 부하를 줄이는 기능을 제공한다.

AF_XDP : 네트워크 패킷을 직접적으로 사용자 애플리케이션으로 전송할 수 있는 구조를 제공하여, 메모리 버퍼 관리를 통해 매우 높은 데이터 처리율을 달성하며, 이 과정에서 RX 및 TX 링을 효율적으로 활용한다.

▼ 하드웨어 오프로드 기술이란?

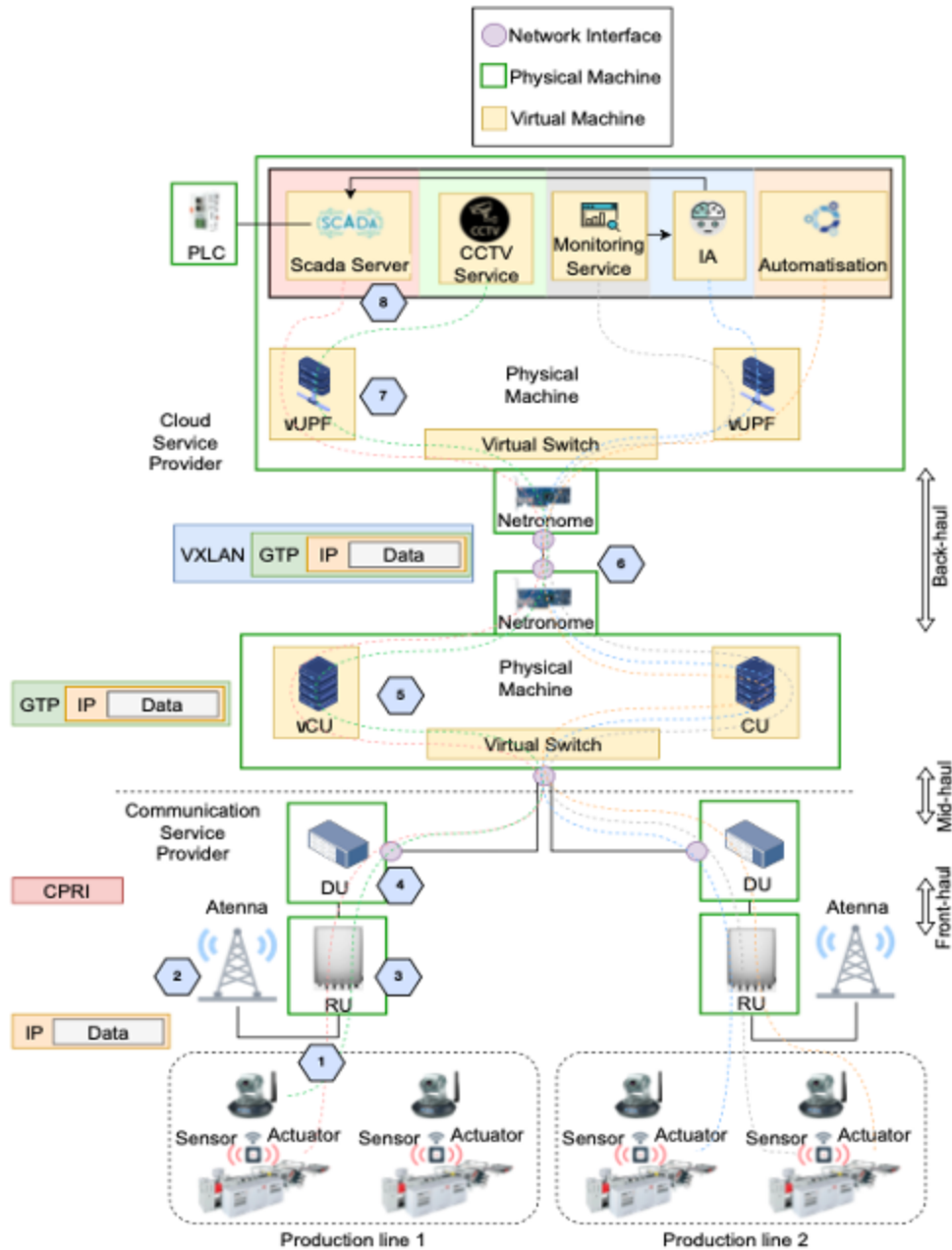
프로그래밍 가능한 하드웨어는 메모리 접근성이 지속적이어서, 필터링 규칙의 수가 증가하더라도 추가 메모리 액세스 시간의 부담이 거의 없어지며, 이로써 보다 정교한 대기 시간 제

어가 가능해진다. 이와 더불어 하드웨어 장치는 소프트웨어 데이터 플레인에 비해 뛰어난 처리 속도를 자랑하며, 이는 빠른 데이터 전송 속도를 암시한다.

FPGA 및 SMARTNIC는 30TB/S의 내부 파이프라인 SRAM 대역폭과 최대 100GBE의 물리적 네트워크 인터페이스를 갖춘 최신 데이터 센터의 까다로운 충족하도록 설계가 되어있다. 하지만 높은 가격과 SDK 및 드라이버의 사유권 문제를 가지고 있다. INTEL은 NFV 인프라를 위한 SMARTNIC 솔루션을 통해 고성능 및 저지연 연결을 제공하고 있지만, 높은 비용과 독점적인 코드로 인해 선호하지 않는다. 또한 NETFPGA와 같은 오픈 소스 솔루션은 100GBPS의 네트워크 전송을 제공하지만 SUSE RIFFA 드라이버 단점으로 인해 NETFPGA는 6G 네트워크 KPI를 충족하지 못한다.

이러한 문제를 해결하기 위해 NETRONOME과 CORIGINE은 AGILO SMARTNIC 및 소프트웨어를 제공한다. P4나 강제적인 C 언어를 이용하여 SMARTNIC를 프로그래밍할 수 있는 SDK를 제공하고 있다. SMARTNIC은 하드웨어 오프로드를 지원하여 하드웨어 프로그래밍 가능성과 유연성을 갖추고 있으며 AF XDP 지원을 하며 하드웨어 오프로드와 커널 바이패스 기술을 결합하여 복잡한 네트워크 처리를 가능하게 한다.

▼ 다중 테넌트 pre-6G 아키텍처 개요



이 그림은 인더스트리 4.0 환경에서 5G 이후 및 프리-6G 아키텍처를 보여준다.

- 1 CSP(통신 서비스 제공업체)는, 데이터를 처리하고 다양한 네트워크 서비스에 연결해준다.
- 2 MIMO 안테나는 데이터 신호를 강화하여 더 멀리, 정확하게 전송한다.
- 3 RU는 디지털 데이터를 무선 신호로 변환해 안테나로 보낸다.

- 4 DU는 기지국의 부분적인 역할을 담당하며 CU가 제어한다.
- 5 vCU는 데이터를 코어 네트워크로 이동하며, 네트워크 효율성과 자원 사용을 최적화 하는데 중요한 역할을 한다.
- 6 vCU와 vUPF 간의 네트워크 트래픽 전달을 담당하는 SmartNIC 식별하며 네트워크 데이터 경로 처리, 제어 및 모니터링 기능을 오프로드한다. 즉, 유연성을 유지하며 네트워크 통신 성능을 최적화하는게 목표이다.
- 7 가상 네트워크 기능이다. vUPF는 사용자 이동성, 인증, 핸드오버 관리 및 사용자 등록 등을 담당한다.
- 8 네트워크 아키텍처 내에서 특정 산업 또는 응용 분야에 맞춤형 서비스를 제공한다.

차세대 네트워크 요구사항을 충족하기 위해 중요하며 낮은 지연 시간, 높은 데이터 전송 속도, 뛰어난 신뢰성을 제공하며 네트워크 슬라이싱 기능을 통해 다양한 애플리케이션과 서비스 등 네트워크 자원을 효과적으로 공유하고 관리할 수 있다.

Agilio CX SmartNIC : 고성능 네트워크 인터페이스 카드는 데이터 경로 최적화하여 까다로운 KPI 충족하는 것을 목표로 한다.

네트워크 세그먼트에서 네트워크 패킷은 센서와 액추에이터에서 수신 및 전송이 되는데 각 CU에 의해 GRPS 터널링 프로토콜 (GTP)를 사용하여 캡슐화된다. 다양한 위치를 따라 네트워크 이동성을 허용한다.

또한 가상 스위치는 네트워크 트래픽을 가상 네트워크 운영자 간에 격리하고 네트워크 리소스를 분리할 수 있도록 두 번째 터널을 생성하는데 VXLAN이 사용된다.

▼ Pre-6G 사용 사례

이 연구는 동적 제어, 모니터링 및 네트워크 슬라이싱이 가능한 고성능 프로그래밍 가능한 데이터 플레인 개발하는 것에 목표를 둔다.

미래의 6G 네트워크의 데이터 플레인(프론트홀에서 백홀까지)이 필요한 성능과 효율성으로 고급 사용 사례를 지원할 수 있도록 하는 것이 목적이다.

TABLE I: Use cases key performance requirements

Use case	E2E Latency	Reliability	Bandwidth	Packet Size	Slice config
(1)	<0.1ms	10^{-8}	<100Mbps	<1500B	<1s
(2)	<33ms	10^{-6}	<3Gbps	<1500B	<1s
(3)	<0.15ms	10^{-6}	<10Gbps	<300B	<1s
(4)	<0.5ms	10^{-5}	<10Gbps	<132B	<1s
(5)	<0.1ms	10^{-6}	<10Gbps	>200B	<1s

(1) PLC 제어 기능을 엣지로 오프로드

새로운 가상화된 구성 요소와 기계 구성 요소 간의 통신 네트워크는 6G 시나리오와 관련된 성공적인 트래픽 전송 및 제어 방법을 위한 중요한 요소이다.

컨테이너 또는 가상 머신으로 실행하여 엣지에 PLC 제어 기능을 오프로드하는 것은 생산 프로세스를 제어하는 데 더 큰 유연성을 제공한다.

네트워크 슬라이싱 전략을 적용하여 낮은 산업용 애플리케이션 주기 시간과 매우 정밀한 동기화를 지원해야 하는 저지연 및 고신뢰성이 필요하다.

(2) 스마트 운송 차량 : 위치 추적 및 비디오 처리 오프로드

1 AGV 내장 카메라를 사용하여 비디오를 캡처, 객체 감지, 의사 결정용으로 엣지 컴퓨팅 노드로 비디오를 스트리밍

2 실시간 의사 결정 알고리즘을 기반으로 실내 및 실외 자율 네비게이션

비디오 전송에 높은 데이터 속도를 요구하기 때문에 비디오 분석하고 의사 결정을 내리며 최적의 통신을 보장하는 효율적인 네트워크 아키텍처가 필요하다.

즉, 이 사례에서는 데이터 속도 요구 사항, 초고정밀 및 의사 결정 등 첨단 기술과 관련하여 분석된다.

(3) Advanced Network Slicing

다양한 고객 요구에 대해 유연하고 맞춤형 E2E 보장된 QoS를 제공한다.

네트워크 데이터 통신을 기반으로 여러 가지 세분화 수준의 온디맨드 유연한 네트워크 슬라이스 정의 가능하다.

네트워크 슬라이스 인스턴스와 시간이 1초미만이어야 하고 복잡한 산업 환경에서 SLA를 준수하고 QoS를 보장하기 위한 의도 기반 네트워크 슬라이싱 관리를 지원하여 효율적인 네트워크 트래픽 통신 보장한다.

(4) 실내 농업 시나리오에서 동물 추적

동물의 활동, 복지 및 건강을 모니터링하는 목걸이를 착용하며 동물 복지에 영향을 미칠 수 있는 것이 감지되면 즉시 알린다.

모니터링 데이터는 엣지 컴퓨터 노드로 전송되어 AI 에이전트가 동물의 이동성과 건강 데이터를 처리하여 동물의 복지 건강을 보장한다.

(5) 공항 서비스 및 수하물 처리 로봇

항공 수하물 서비스 및 수하물 처리 로봇을 사용하여 자동화된 수하물 처리 시스템을 구현한다.

AGV 로봇을 활용하여 여행객의 수하물을 수하물 처리 컨베이어 벨트 간 이동시킨다.

즉, 비디오 라이브 스트리밍을 통해 로봇과 AGV의 경로 상 물체 인식 및 충돌 회피 수행하여 자동화 및 지속 가능한 수하물 분류 시스템 구현, 높은 신뢰성 및 낮은 통신 요구 필요하다.

그냥 이 사례는 미래 6G 네트워크에서의 성능과 효율성을 검증과 SLA, KPI 요구 사항을 충족하기 위한 기준을 나타내며 신뢰성과 실행 가능성을 평가하고자 하는 것이다.

▼ 프로토타이밍 플랫폼 및 구현

6G 네트워크를 위한 고성능 및 고신뢰성을 지원하기 위해 개발된 프로토타입 플랫폼의 개발과 테스트에 대해 설명한다.

▼ A. XDP 기반 Agilio CX SmartNIC 참조 데이터 경로

▼ 실험환경

최대 처리 속도 25Gbps 의 Metronome Flow Processor(NFP) 드라이버를 통해 XDP 및 DPDK와 SmartNIC을 제공하는 Agilio CX SmartNIC 플랫폼을 기반으로 한다.

Agile CX SmartNIC에는 NFP-4000 프로세서가 있으며 60개의 프로그래밍 가능한 흐름 처리 코어, 48개의 패킷 처리 코어, RX 및 TX 채널이 있는 PCIe Gen3 2x8 인터페이스 및 2xSFP28 25GbE 물리적 인터페이스도 포함된다.

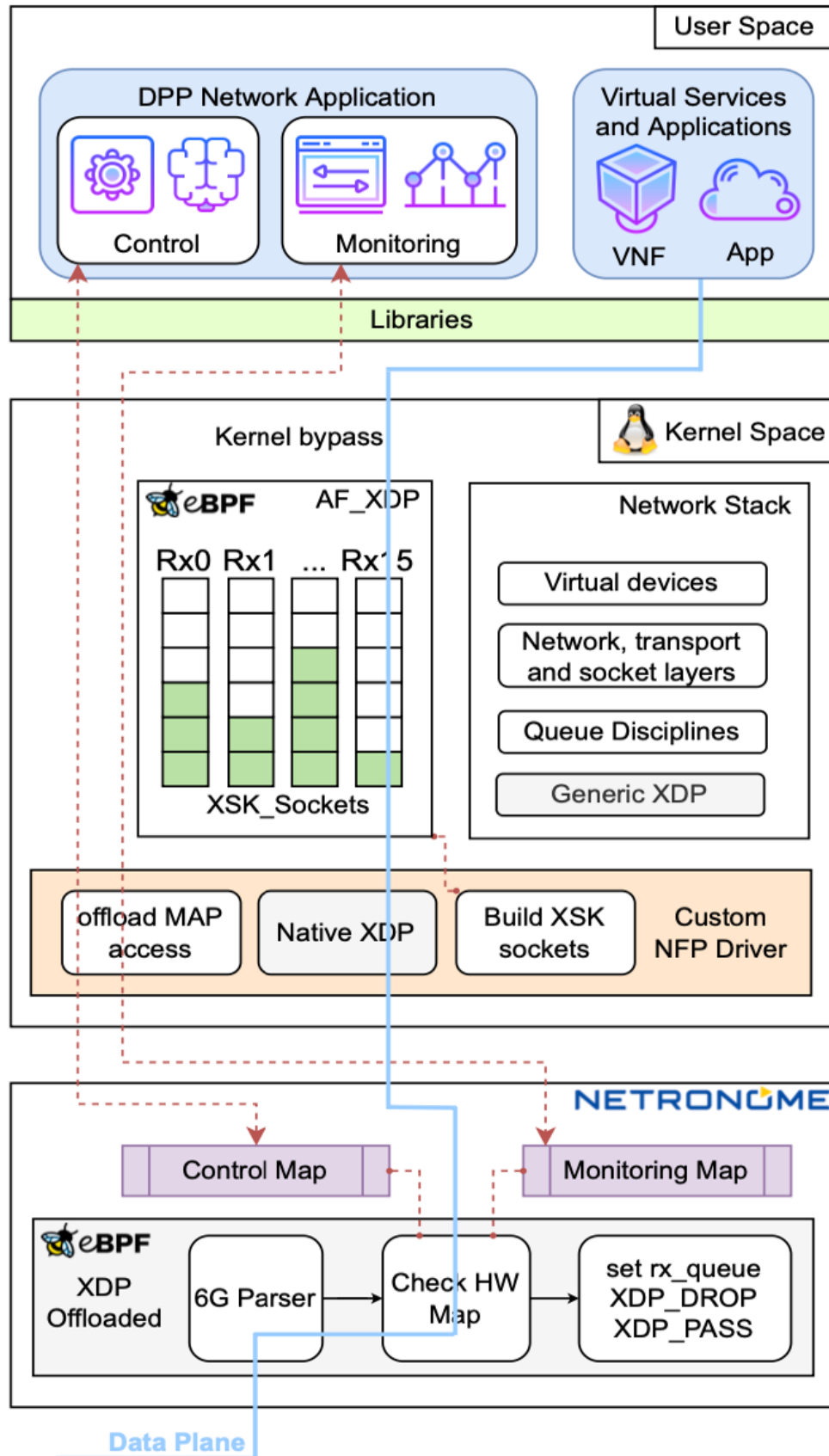
SmartNIC는 Agilio eBPF 펌웨어를 사용하여 XDP를 오프로드 및 드라이버 수준에서 구현한다.

SmartNIC은 최대 16개의 하드웨어 큐를 통해 각 큐를 사용자 공간으로 직접 연결하는 XSK 소켓에 연결하고 고성능 패킷 처리를 지원한다. 또한, AF XDP를 통해 고성능 패킷 처리를 위해 XSK 소켓을 생성한다.

- (1) 다중 테넌시 지원을 통해 구현된 오프로드 및 프로그래밍 가능한 pre-6G 데이터 경로
- (2) 오프로드된 고성능 실시간 모니터링 도구
- (3) 고급 네트워크 슬라이싱 및 트래픽 필터링을 제공하는 오프로드된 제어 기능
- (4) XSK 소켓 지원을 제공하고 따라서 Linux 커널 기반 구현의 성능을 향상시키는 NFP 드라이버의 맞춤형 버전

pre-6G 네트워크 아키텍처의 전송 네트워크 세그먼트에서 고성능 통신을 향상시키기 위해 동일한 네트워크 카드를 사용하여 하드웨어 오프로드 기능과 커널 바이패스 기술을 결합한 솔루션을 접근한다.

▼ B. 제안된 프레임 워크 아키텍처



< 파란색 선 - 데이터 플레인 , 빨간색 점선 - 제어 플레인 >

위에서 부터 세가지 공간으로 나뉘어 진다.

▼ 최종 사용자에게 가장 가까운 부분인 사용자 공간

두 가지 핵심 구성요소가 있다.

- DPP 네트워크 애플리케이션 : 제어 및 모니터링 도구를 제공한다.
- 가상 서비스 및 애플리케이션 : 네트워크 데이터 플레인에 있는 다양한 서비스와 애플리케이션을 제공한다.
 - 제어 애플리케이션 : 하드웨어 제어 맵에 네트워크 규칙을 삽입 및 제거하는 역할을 한다.
 - 모니터링 애플리케이션 : 모니터링 맵에서 규칙을 삽입하고 제거할 뿐만 아니라 하드웨어 파이프라인에서 이 맵에 저장된 실시간 통계 및 유용한 정보를 수집한다.
 - Agilo CX SmartNIC에 의해 처리된 네트워크 트래픽의 실시간 지표를 제공한다.
 - 초당 패킷 수, 수신/전송된 패킷,

Libbpf와 ask 등의 라이브러리는 사용자 애플리케이션과 커널 공간 사이의 통신을 도와준다..

▼ 하드웨어 및 소프트웨어 인터럽트를 처리하는 Linux 커널 공간

네트워크 패킷이 커널 공간에 도달하면, 네트워크 스택에서 맞춤형 NFP 드라이버를 사용한다.

이 드라이버는 소켓을 통해 데이터를 사용자 공간으로 전달하게 되는데, 이전에 사용된 공개 NFP 드라이버는 스택 병목현상을 일으켜 네트워크 스택의 병목 현상 즉, 성능 저하의 원인으로 확인되었다.

이를 해결하기 위해 새로운 맞춤형 NFP 드라이버가 개발되었으며, AF XDP와 XSK 소켓을 활용해 네트워크 스택을 우회함으로써 통신 속도를 높였다. 이 NFP 드라이버는 최대 16개의 소켓을 지원하며 하드웨어 제어 및 모니터링 맵에 접근도 가능하다. Native XDP를 통해 유연한 데이터 경로도 제공해, 6G 네트워크의 엄격한 성능 요구사항도 충족할 수 있다.

▼ Agilio CX SmartNIC을 사용하는 하드웨어

SmartNIC에 탑재된 맞춤형 XDP 오프로드 펌웨어는 eBPF 기반 제어 및 모니터링 맵이 있다.

eBPF 맵

제어맵 : 각 규칙에는 32비트 키와 32비트 값이 포함되어 있다.

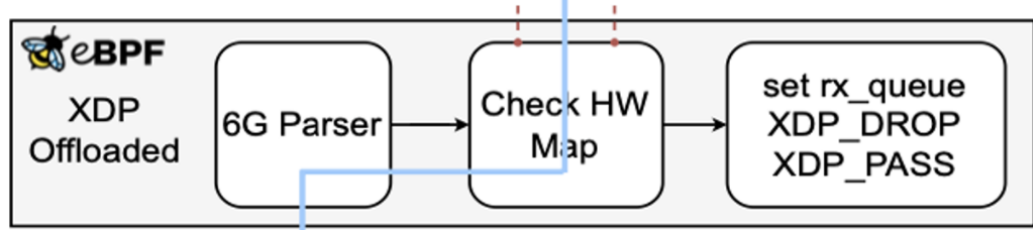
이를 동작을 나타내기 위해선 16비트와 필요한 출력 값을 나타내기 위해 16비트로 나뉜다. (ex, 특정 rx 큐에 설정)

모니터링 맵 : 32비트 키와 64비트 값을 포함되어 있다.

32비트키는 패킷 길이, 32비트는 이 특정 규칙에 일치하는 패킷 수를 나타낸다.

XDP 오프로드 펌웨어는 다른 구성 요소, pre-6G 파서, 하드웨어 맵이 확인되고 출력이 결정된다.

출력 결정이 이루어지는 논리로 나눌 수 있다.



프리-6G 파서 : 네트워크 플로우를 처리하고 필요한 메타 데이터를 추출하는 역할을 한다. 이 정보는 나중에 일치 모듈에서 사용할 해시 식별자를 계산하는데 사용된다.

일치모듈은 네트워크 패킷이 eBPF 제어 맵의 해시 식별자와 일치하는지 확인하고, 이에 따른 3가지 동작을 합니다.

- 특정 하드웨어 큐로 패킷 전달
- 패킷 삭제
- 기본 큐를 통해 커널 네트워크 스택으로 전달

▼ C. 슬라이스 정의: 최종 사용자 및 슬라이스 관리 개요

```

{
  "Resources": [{
    "resourceId": "F8A4C949",
    "encapsulationID1": "00000445",
    "encapsulationType1": "vxlan",
    "srcIP": "146.191.50.26",
    "dstPort": "4789",
  }, {
    "resourceId": "B6E6B2B3",
    "encapsulationID2": "8894D0D4",
    "encapsulationType2": "gtp",
    "srcIP": "10.100.0.19",
    "dstPort": "2152",
  }],
  "Intent": {
    "flowAgentName": "XDP-based"
    "actionType": "INSERT",
    "actionName": "CREATE_SLICE",
    "slice_id": "03E8",
    "priority": "1",
    "MAB" : "12000000",
    "MGB" : "10000000"
  },
  "Params": [{
    "paramName": "interfaceName",
    "paramValue": "eth0"
  }]
}

```

< 의도 기반 메시지 >

여기서는 유럽 6G Brains 프로젝트와의 연계를 통해 슬라이스 인터페이스화 및 서비스 인스턴스 생성의 실용적인 접근 방법을 제시한다.

DPP 애플리케이션을 통해 수신된 슬라이스 정의 메시지에 집중하여, 이 메시지에서는 우선순위, 최대 허용 대역폭(MAB) 및 최소 보장 대역폭(MGB)과 같은 슬라이스 구성 속성도 포함하며 필요한 패킷 메타 데이터 구조 및 식별자를 처리하고 생성하는지 설명한다.

▼ D. 슬라이스 정의: 제안된 모델, 정책 및 논리적 구현

```

struct pkt_meta {
    __u16 out_macsrc[3];
    __u16 out_macdst[3];
    __u16 in_macsrc[3];
    __u16 in_macdst[3];
    __u16 out_ethproto;
    __u16 in_ethproto;
    __be32 out_ip4src;
    __be32 out_ip4dst;
    __be32 in_ip4src;
    __be32 in_ip4dst;
    __u8 out_ip4proto;
    __u8 in_ip4proto;
    union {

        __u32 out_ports;
        __u16 out_port16[2];
    };
    union {
        __u32 in_ports;
        __u16 in_port16[2];
    };
    __u32 vni;
    __be32 teid;
    __u32 ethercatid
};

```

<패킷 메타데이터 구조>

DPP 네트워크 애플리케이션과 맞춤형 eBPF 알고리즘을 사용하여 smartNIC을 프로그래밍하고, eBPF 시스템 호출을 통해 사용자 공간, 커널, 하드웨어 간의 제어 통신을 가능하게 한다.

네트워크 플로우를 제어하고 관리하는데 필요한 데이터 구조는 BPF MAP TYPE HASH 맵을 사용하여 제어규칙을 저장하며 네트워크 플로우의 효율적인 제어를 촉진한다.

▼ 알고리즘

```
1: struct pkt_meta;
2: procedure XDP_PROG(pkt)
3:   pkt_meta ← parse_headers(pkt);
4:   hash ← calculate_hash(pkt_meta);
5:   control_entry ← control_map.lookup(hash);
6:   if control_entry != NULL then
7:     if control_entry.action == 0 then
8:       return XDP_DROP
9:     if control_entry.action == 1 then
10:      pkt.rx_queue ← control_entry.queue
11:   else
12:     return XDP_PASS
13:   monitor_entry ← monitor_map.lookup(hash);
14:   if monitor_entry != NULL then
15:     monitor.update(hash,pkt_meta,1)
16:   return XDP_PASS;
```

<6G 네트워크를 위한 eBPF-XDP 패킷 제어 알고리즘>

Agilio CX SmartNIC에서 구현된 XDP 오프로드 파이프라인 로직이다. 이 알고리즘은 네트워크 트래픽 분류, 필터링 및 제어를 가능하게 하고, 네트워크 카드와 사용자 공간 간의 트래픽을 가속화시킨다.

eBPF 프로그램이 실행되어 패킷 헤더를 추출하고 필요한 메타데이터(패킷의 내부 및 외부 MAC, IP, Port 등을 포함 및 해시 계산에 사용)를 수집된다.

▼ 알고리즘 동작 해석

3번째 줄 : SmartNIC 관점에서 패킷이 물리적 인터페이스에 도착하면 오프로드된 eBPF 프로그램이 패킷을 캡처하고 패킷 헤더 추출을 한다. 들어오는 패킷의 모든 비트를 구문 분석하고 분석하여 pkt 메타 구조를 완성하는데 필요한 정보를 추출한다. (내외부 MAC, IP 주소, 포트, 링크 프로토콜, 전송 프로토콜 및 VXLAN 및 GTP 식별자를 포함하여 pre-6G 네트워크 플로우를 식별하는 정보가 포함)

4번째 줄 : 3번째 줄의 데이터를 32비트 길이의 해시를 계산하는데 사용되며, 이는 들어오는 네트워크 패킷을 명확하고 개별적으로 식별한다.

5,6번째 줄 : 제어 맵에 일치하는 항목이 있는지 확인하는 용도로 사용된다. 규칙이 확인되면 액션 값이 얻어진다.

7,8번째 줄 : 동작 값이 0이면 패킷이 삭제된다.

9번째 줄 : 동작 값이 1이면 사용자 공간으로 전송되기 위해 XSK 소켓으로 전송된다는 것을 의미한다.

10번째 줄 : 제어 규칙에서 출력 큐/소켓의 값을 얻고 패킷 구성 메타데이터가 업데이트된다.

12번째 줄 : 해시 값과 일치하는 제어 규칙이 없는 경우 패킷은 기본 큐/소켓(XDP PASS)을 사용하여 사용자 공간으로 전송되며 이 값은 0되게 된다.

13번째 줄: 패킷이 삭제되지 않은 경우 모니터링 맵을 확인하여 들어오는 패킷의 해시와 일치하는 규칙이 있는지 확인한다.

14,15번째 줄 : 들어오는 패킷의 해시와 일치하는 규칙이 이 있는 경우, 현재 처리된 패킷 정보에 추가하고 패킷 카운터를 1씩 늘려 규칙을 업데이트한다.

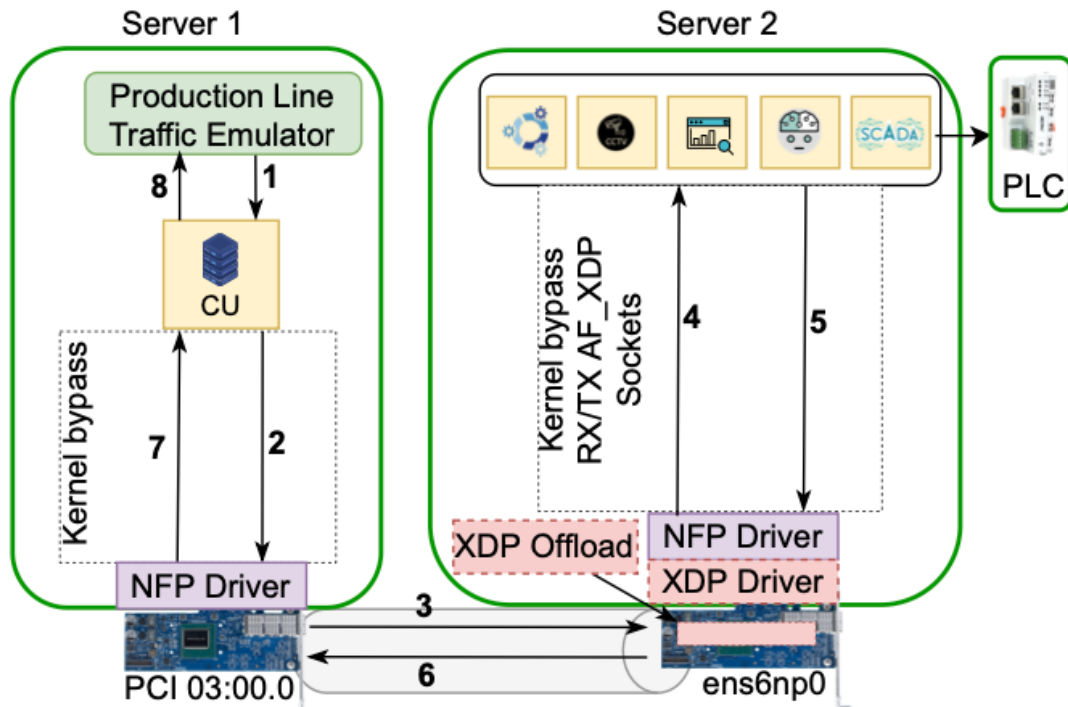
16번째 줄 : 패킷은 5번째 줄에서 이전에 선택한 출력 큐를 사용하여 사용자 공간으로 전송된다.

간략히 말해서, 제어 맵에서 일치하는 규칙이 있는지 검사하고, 일치하는 규칙이 있으면 해당 규칙에 따라 패킷이 삭제되거나 XSK 소켓을 통해 사용자 공간으로 전송됩니다. 일치하는 규칙이 없으면 패킷은 기본적으로 사용자 공간으로 전송됩니다. 또한, 들어오는 패킷의 해시와 일치하는 모니터링 규칙이 있으면 관련된 패킷 정보

가 업데이트되고 카운터가 증가합니다. 패킷은 이후 선택된 출력 큐를 통해 사용자 공간으로 최종 전송이 된다.

▼ 실증적 검증

▼ A. 실험설정



<성능 평가 테스트 베드 구현>

이 실험환경은 두 대의 물리적 서버를 사용하여 클라우드 서비스 제공자 아키텍처를 복제하였다.

두 서버 모두 AF XDP 소켓 지원을 포함하는 맞춤형 NFP 드라이버가 실행되고 있다.

한 서버는 리눅스 컨테이너에서 개발된 CU와 다른 서버는 SmartNIC에 구현된 맞춤형 XDP 오프로드 펌웨어가 있고 AF XDP은 커널을 우회, 다양한와 맞춤형 서비스를 리눅스 컨테이너를 실행한다.

1. 서버 1에서 트래픽생성

서버 1에는 트래픽 시뮬레이터가 설치되어 있으며, 물리적 장치에서 센서 역할을 하고 네트워크를 통해 비디오를 맞춤형 서비스로 전송한다.

Pktgen 도구가 CU 내부에 있어 송신기를 시뮬레이션하고, 이중 캡슐화 및 실시간 프로토콜(RTP)을 사용하여 비디오 프레임을 전송한다.

2. 서버1→SmartNIC으로 트래픽 전송

서버 1에서 생성된 네트워크 트래픽은 Linux 커널을 우회하여 DPDK를 사용하여 SmartNIC으로 직접 전송된다.

3. SmartNIC에서의 트래픽 처리 및 전송

서버 1의 SmartNIC은 PCI 슬롯을 통해 CU에서 생성된 트래픽을 수신한다.

그 후 물리적 인터페이스를 통해 서버 2로 전송한다.

4. 서버 2에서의 트래픽 처리

서버 2에서는 ens6np0 인터페이스를 통해 트래픽을 수신하고, SmartNIC은 오피로드된 파이프 라인을 사용하여 트래픽을 처리한다.

패킷 메타데이터를 제어 맵에 저장된 규칙과 대조하여 처리한다.

규칙에 부합하면 AF XDP 소켓을 사용하여 맞춤형 서비스로 전송된다.

5. 맞춤형 서비스에서의 트래픽 처리

맞춤형 서비스에서 네트워크 트래픽을 수신하고 목표 애플리케이션이 적절하게 처리한다.

또한 SmartNIC의 하드웨어 모니터링 맵에서 성능 지표를 수집하고 필요에 따라 제어 규칙을 업데이트한다.

6. 맞춤형 서비스에서 SmartNIC으로의 트래픽 재전송

네트워크 플로우가 수정되고, 패킷의 페이로드에 매개변수가 도입된 후 동일한 AF XDP 소켓을 사용하여 Linux 커널을 우회하고 SmartNIC으로 다시 전송된다

7. 서버 1로 트래픽 전달

네트워크 플로우는 서버 1로 전달되는데, ens6np0 인터페이스를 사용한다.

8. 서버 2에서 SmartNIC을 통한 트래픽 전송

서버 2에 할당된 SmartNIC은 네트워크 트래픽을 CU로 보낸다.

9. 최종 네트워크 플로우 처리

네트워크 플로우를 맞춤형 서비스에 의해 내려진 결정을 포함하여 관련된 작업을 수행한다.

▼ B. 결과

▼ 검증단계 전 설정

가속화된 데이터 경로를 실증적으로 검증하는 섹션이다.

실시간 통신이 안전하고 최적의 운영을 위해 혼잡한 시나리오에서도 높은 신뢰성과 저지연 통신을 제공한다.

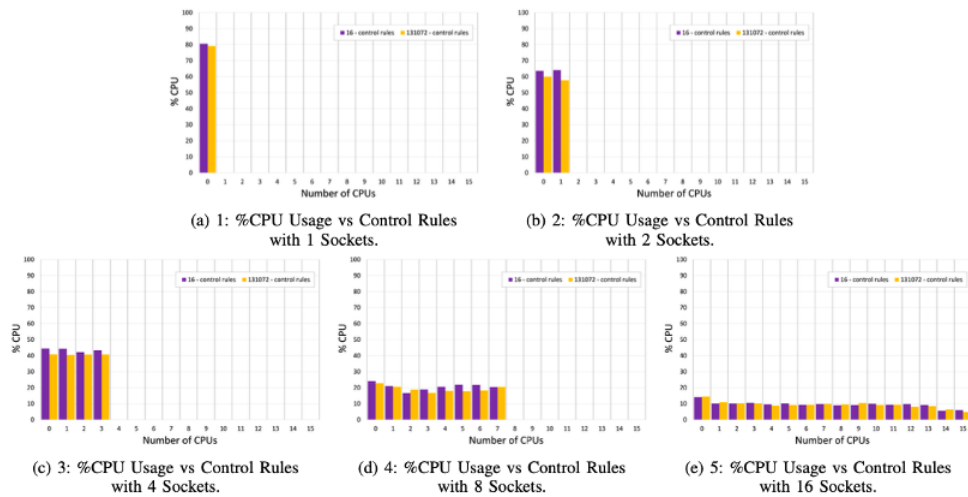
물리적 서버와 제품 라인의 통신을 보장하기 위해 격리된 AF XDP 소켓을 제안하고 . 각소켓에는 시스템의 최대 성능을 얻기 위해 CPU 리소스를 할당 받는다.

irq affinity 설정을 통해 특정 CPU가 네트워크 수신 큐를 처리한다. 시스템의 네트워크 수신 큐와 비례하여 CPU수가 설정된다.

패킷 크기는 132바이트 ~ 1500바이트까지 다양하며, 최대 25gpbs 속도로 초당 2000만개 패킷을 전송할 수 있는 실험을 포함한다.

1,2,4,8,16개의 소켓이 있으며 5회가 실행되고 5회의 평균치를 낸다.

▼ CPU 부하 분석



SmartNIC에 최대 131072개의 제어 규칙이 삽입된 상태에서 CPU 사용률을 분석한다.

실험은 1,2,4,8,16 에 대해 진행되었고, $16 * 2^n$ 회 반복 실행된다.

결과

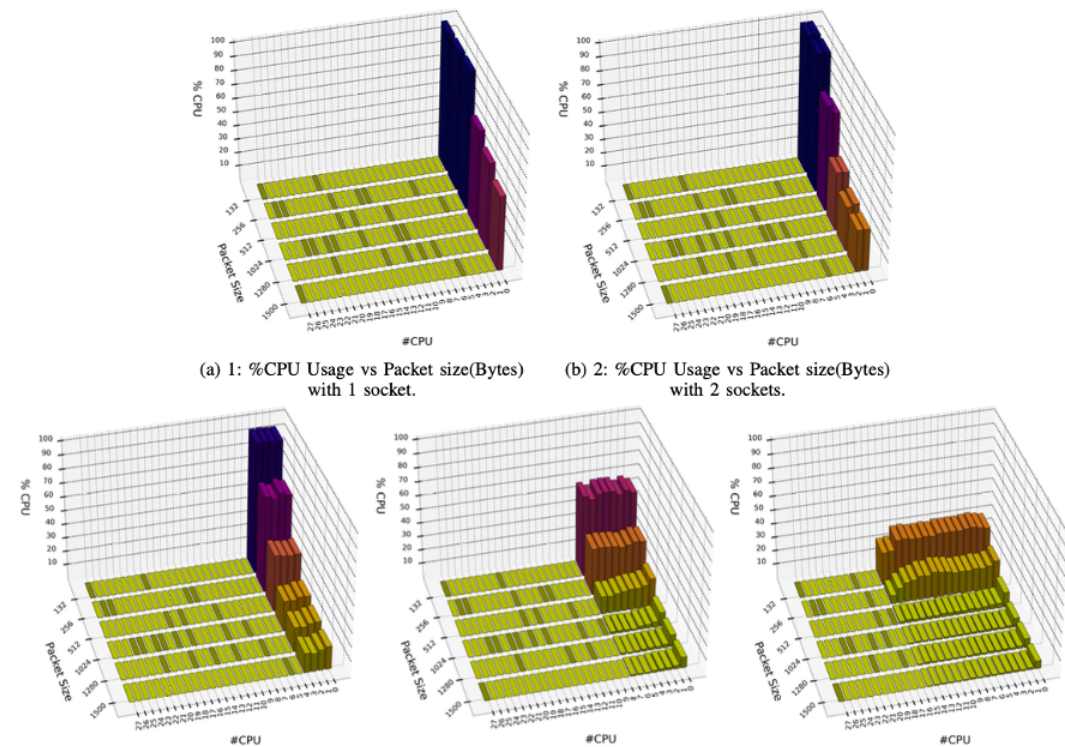
하나의 소켓을 사용하는 경우(CPU는 평균 70% 부하), 규칙 수가 증가해도 CPU 부하는 증가하지 않는다.

소켓 수가 증가함에 따라 CPU 부하가 감소하며, 16개의 소켓을 사용할 때 CPU 부하는 10% 미만이다.

효과

제어 규칙의 수와 독립적으로 CPU 부하가 일정하게 유지되어 시스템의 확장성을 입증한다. 최대 25Gbps의 네트워크 트래픽을 처리하면서 대부분의 CPU가 유휴 상태로 남아 다른 작업에 활용될 수 있다.

▼ 패킷 크기에 따른 CPU 부하



패킷 크기가 132바이트에서 1500바이트로 변경될 때 CPU 사용률 변화를 분석이 되어있다.

결과

소켓이 하나이고, 패킷 크기가 512바이트 이하 - CPU 사용률 100%에 임박

BUT 패킷 크기가 1500바이트로 증가하면 CPU 사용률 35%까지 감소

16 개 CPU에서 132 패킷 - CPU 사용률은 25%

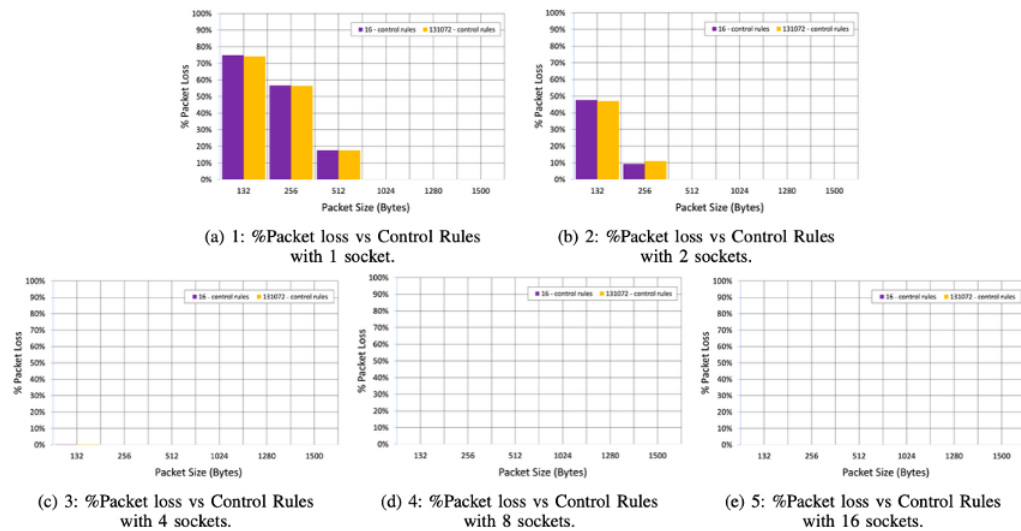
BUT 패킷 크기가 512바이트를 넘어서면 CPU 사용률은 5% 미만으로 감소

소켓 수가 증가할수록 CPU 부하는 감소하며, 패킷 크기가 커질수록 부하가 더욱 감소한다.

패킷 크기가 증가함에 따라 CPU 부하가 줄어드는 현상은 eBPF/XDP 기반 프레임워크의 확장성을 입증하며, 네트워크 트래픽을 효율적으로 분산 처리할 수 있는 시스템의 능력을 보여준다.

▼ 패킷 손실률의 분석

패킷 크기와 제어 규칙 수의 변화에 따른 패킷 손실률을 측정했다. 시스템의 신뢰성을 확인하는 중요한 요소이다.



<다른 소켓 수를 사용했을 때 % 패킷 손실, 바이트 단위의 패킷 크기 및 제어 규칙 수 간의 관계>

소켓이 1개이고, 패킷 크기가 132 바이트 일 경우 - 패킷 손실률은 70%

패킷 크기가 1024,1280,1500 바이트일 경우 - 0%에 수렴

소켓이 2개이고, 패킷 크기가 132 바이트 일 경우 - 패킷 손실률은 40%

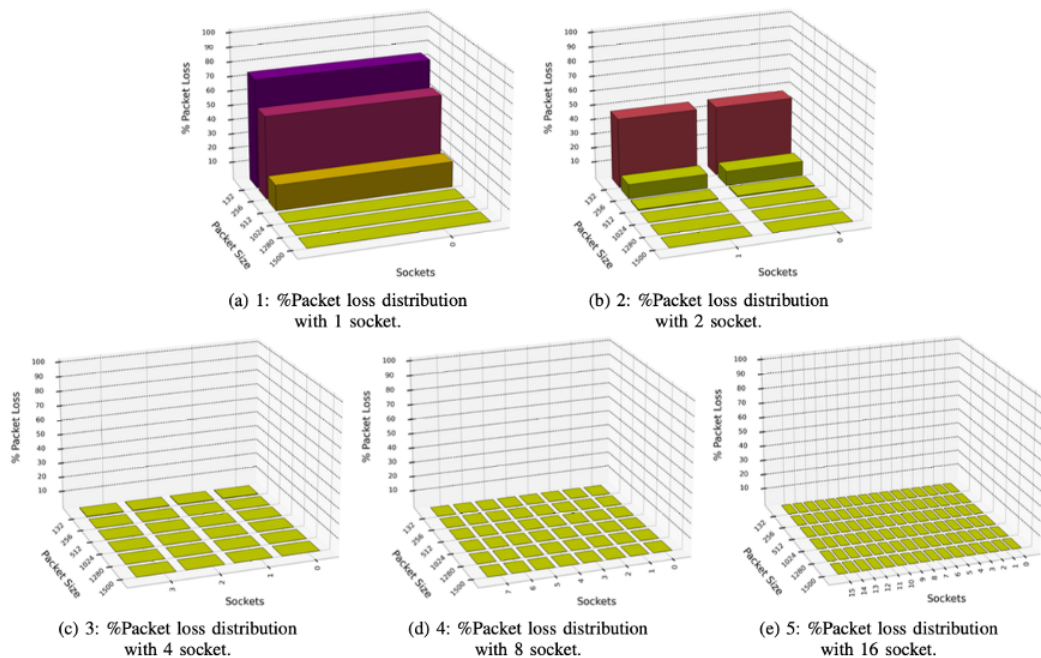
패킷 크기가 512 바이트 이상일 경우 - 0%에 수렴

소켓이 4개 이상일 경우, 최악의 조건(패킷 크기 132바이트, 제어 규칙수 131072)에서도 패킷 손실률은 0%에 수렴

SmartNIC이 처리하는 네트워크 트래픽은 다양한 소켓 간에 고르게 분산된다.

소켓이 서로 격리되어 있고 각각의 소켓에 균등하게 트래픽이 분배됨으로 각 소켓의 성능이 최적화된다. (최대 131072 제어규칙을 활용하여 이루어짐)

소켓 수가 증가할수록 패킷 손실률이 감소하며, 시스템의 신뢰성과 성능이 향상됨을 입증한다.



<% 패킷 손실, 바이트 단위 패킷 크기, 소켓 수 간의 관계>

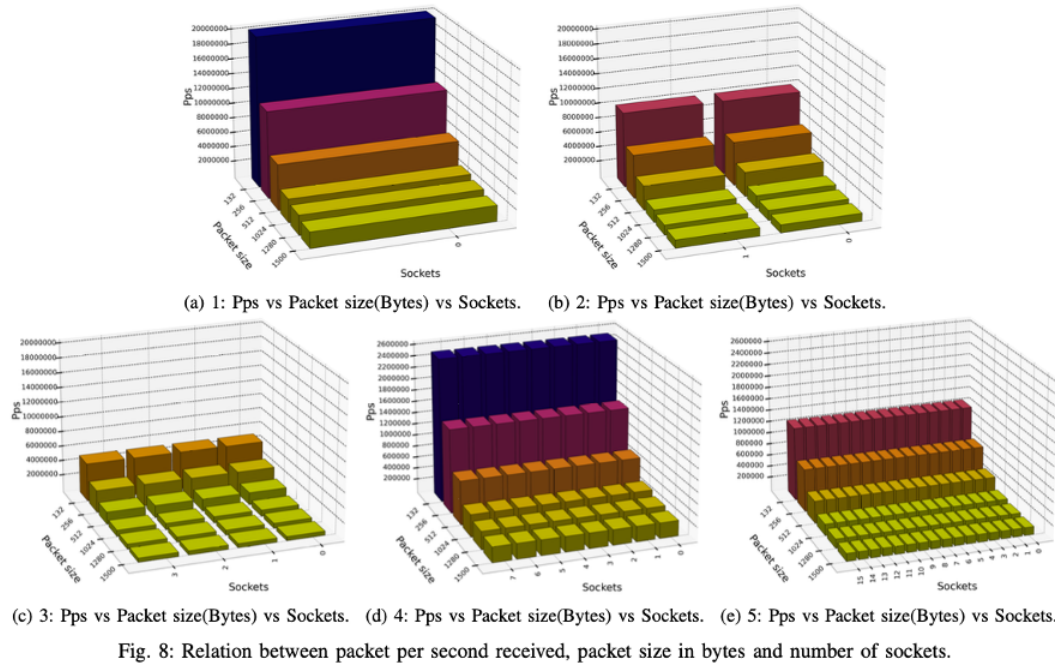
이 그림은 모든 제어 규칙의 평균을 계산한 것이다.

패킷 크기가 132바이트일 때 거의 70%의 패킷이 손실되지만 소켓 수가 4개 이상인 경우 손실률은 0이 된다. 즉, 패킷 크기와 소켓 수에 따른 변화를 보여준다.

▼ 대역폭

초당 패킷수, 대역폭, 처리량을 중점을 두었다.

패킷 크기에 따라 초당 패킷 수가 다르며, 네트워크의 성능을 나타내는 주요 지표이다.



<초당 수신된 패킷 수, 바이트 단위 패킷 크기, 소켓 수 간의 관계>

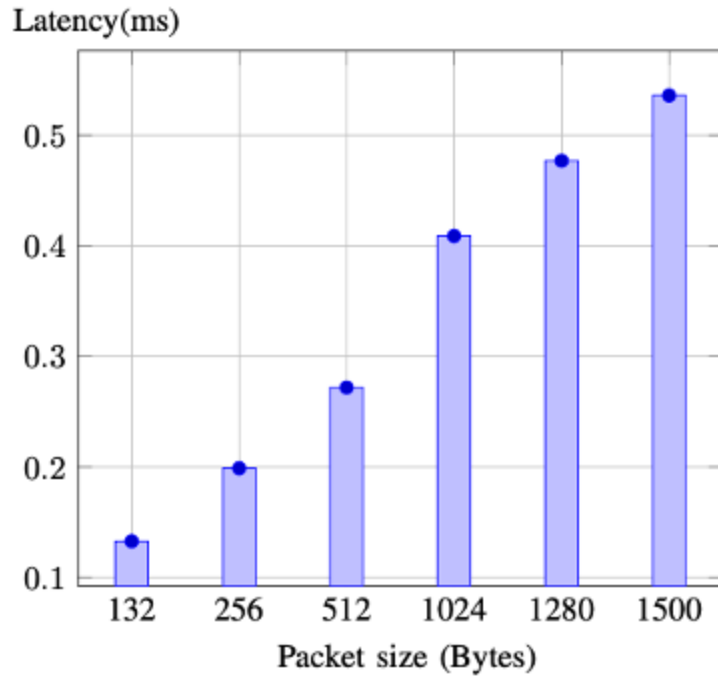
최악의 시나리오 (그래프 a) : 패킷 크기가 132 바이트일 때 → 초당 처리되는 패킷 수는 2000만개 (70% 손실)

최적의 시나리오 (그래프 c) : 각 소켓당 500만개씩 초당 처리 할 경우 → 0% 손실이 일어남

패킷 크기와 소켓 수를 조정하면서 초당 전송되는 패킷 수를 계산하여 수행한다.

패킷 손실은 패킷 크기가 증가함에 따라 감소하고, 소켓 수가 증가함에 따라 처리되는 트래픽 양은 동일하게 유지된다.

▼ 지연률



네트워크 전송 지연 시간이 감소하면 최종 사용자의 경험 품질이 직접적으로 향상 되는데, 네트워크의 효율성, 연결 속도 및 대역폭과 관련이 있다.

RTT는 패킷이 출발하여 네트워크를 통과하고, 하드웨어 데이터 경로를 거쳐 맞춤형 애플리케이션에 도달하는데 소요되는 시간을 말한다.

그래프를 보면 패킷의 크기가 커짐에 따라 RTT가 증가하는 경향이 있다.

여기서 가장 까다로운 시나리오는 132바이트 크기인 패킷이 전송될 때 발생되는데 총 2000만개의 패킷이 초당 처리되어야함을 의미한다. 하지만 이 솔루션은 까다로운 시나리오를 만족한다. 하지만 위에서 말했던 5번째 사례같은 큰 패킷 크기와 매우 낮은 지연 시간이 필요한 특정 사례에서는 높은 성능을 갖춘 SmartNIC, 추가 하드웨어 큐, 그리고 서버 CPU 처리 능력이 필요하다.

또는 로드 밸런서를 사용하여 트래픽을 분산시키는 등 여러가지 접근 방식이 고려 될 수 있다.

▼ 결론

eBPF/XDP 기반 네트워크 필터링은 멀티 테넌트 흐름을 동적으로 식별하고 분류할 수 있으며, 빠르게 트래픽 처리할 수 있다. 또한 패킷 필터링과 처리를 더욱 빠르고 효율적으로 만들어 높은 네트워크 성능을 보장한다.

eBPF 하드웨어 맵은 실시간 네트워크 메트릭을 기록하고 보고할 수 있으며 하드웨어 맵은 메모리 접근 시간을 최적화하여 데이터 처리 속도를 향상시킨다.

Agilio CX Smart NIC를 사용하여 네트워크 데이터 플레인의 하드웨어 기반 오프로드를 실현하며 Linux 커널을 우회하여 직접 하드웨어와 통신하게 하여 네트워크 성능을 개선한다.

AF XDP 지원의 NFP 드라이버는 SmartNIC을 통한 고속 데이터 전송을 가능하게 하여 AF XDP 성능을 최적화 한다.

즉, 6G 사용 사례의 까다로운 KPI 충족할 수 있는 확장성과 적합성을 명확하게 보여준다.