전술메쉬 트래픽 QoS 조율을 위한 네트워킹 노드의 개념 설계 및 실시간 모니터링

(Conceptual Design of Networking Node with Real-time Monitoring for QoS Coordination of Tactical-Mesh Traffic)

신준식*, 강문중*, 박주만**, 권대훈**, 김종원*

(Jun-Sik Shin, Moonjoong Kang, Juman Park, Daehoon Kwon, JongWon Kim)

요 약

정보통신 기술의 발전으로 응용 서비스들을 IP(Internet protocol) 기반으로 통합하는 All-IP 기반 미래형 전술망으로의 전환이 지속적으로 진행되고 있다. 이러한 변화에 대응하기 위해 전술 WAN 노드들이 메쉬구조로 연결된 미래형 전술메쉬망에서 응용 서비스의 QoS(quality of service) 보장을 위해, 인프라 계층과 응용 서비스 계층 사이에 존재하는 전술 서비스 메쉬(tactical service mesh) 계층을 도입하는 것을 시도하는 제안이 있다. 하지만 기존 전술망을 구성하는 폐쇄형 네트워킹 박스들과 정적인 QoS 관제도구들은 전술 서비스 메쉬 계층에서 요구하는 지능형 QoS 조율을 위한 동적 QoS 관제를 지원하는 것이 어려운 상황이다. 따라서 본 논문에서는 SDN/NFV 기반 DANOS 화이트박스 서버스위치를 기반으로 다중-액세스 네트워킹 인터페이스와 가상화된 네트워킹 스위치를 포함하도록 하드웨어/소프트웨어 통합 설계한 전술메쉬 WAN 노드를 구성하는 방안을 제안하고 이를 개념화 수준에서 설계하여 제시한다. 또한 설계한 전술메쉬 WAN 노드에 eBPF 기반의 트래픽 모니터링 방식을 연계하는 방안을 제안하고, 이를 통해 전술 트래픽의 QoS 조율을 지원하는 트래픽 모니터링이 가능함을 검증한다.

■ 중심어 : 미래형 전술메쉬망, 지능형 QoS 조율, eBPF 기반 패킷 모니터링, 전술메쉬 WAN 노드

Abstract

With the advancement of information and communication technology, tactical networks are continuously being converted to All-IP future tactical networks that integrate all application services based on Internet protocol. Futuristic tactical mesh network is built with tactical WAN (wide area network) nodes that are inter-connected by a mesh structure. In order to guarantee QoS (quality of service) of application services, tactical service mesh (TSM) is suggested as an intermediate layer between infrastructure and application layers for futuristic tactical mesh network. The tactical service mesh requires dynamic QoS monitoring and control for intelligent QoS coordination. However, legacy networking nodes used for existing tactical networks are difficult to support these functionality due to inflexible monitoring support. Thus, in this paper, we propose a tactical mesh WAN node as a hardware/software co-designed networking node. The tactical mesh WAN node is conceptually designed to have multi-access networking interfaces and virtualized networking switches by leveraging the DANOS whitebox server/switch. In addition, we explain how to apply eBPF-based traffic monitoring to the tactical mesh WAN node and verify the traffic monitoring feasibility for supporting QoS coordination of tactical-mesh traffic.

■ keywords: futuristic tactical mesh network, intelligent QoS coordination, eBPF-based packet monitoring, tactical service mesh WAN node

Ⅰ. 서론

정보통신 기술의 발전에 따라 전술망의 네트워킹을 가속화하여 빠른 지휘 하달 및 정보 공유를 통한 정보 우위를 바탕으로 전투력을 향상시키는 네트워크 중심전의 중요성이 점차 증가하고 있다[1]. 미군 육군의 사례[2]에서 볼 수 있듯이 이러한 네트워크 중심전에 대응하기 위하여, 복잡한 기능들을 하드웨어 노

* 광주과학기술원 전기전자컴퓨터공학부

** 국방과학연구소 2기술연구본부 1부

본 연구는 국방과학연구소가 지원하는 운용체계의 전술망내 정보유통 품질보증 방안 연구과제(UD170050ED)의 일환으로 수행되었습니다..

접수일자 : 2019년 04월 24일

수정일자 : 2019년 05월 02일

게재확정일 : 2019년 05월 02일

교신저자 : 김종원 e-mail: jongwon@nm.gist.ac.kr

드에 모두 담아 전술망에 배치하는 하드웨어 중심의 설계가 아닌, 화이트박스 형태로 하드웨어 노드들을 단순화하고 지능화된 소프트웨어가 SDN(software-defined networking) / NFV(network function virtualization)를 활용하여 노드들의 복잡한 조율 및 관리를 수행하는 소프트웨어 중심의 전술망 설계로 전환되고 있다.

화이트박스 서버스위치는 오픈소스 기반 범용 하드웨어와 리눅스를 기반한 오픈소스 네트워킹 운영체제를 자유롭게 선택하여 구성하는 네트워킹 박스를 의미한다*[3]. 기존의 폐쇄형 네트워킹 박스들을 운영함에 있어 벤더 종속성, 불필요한 네트워킹 기능 탑재로 인한 비용 증가, 자동화 관리의 어려움, 제한적인 관리 도구 등의 문제들을 해소하기 위하여 등장하였다. 범용 서버에서 운용되는 응용 서비스와 유사하게, 화이트박스가 제공할 네트워킹 기능은 운영체제 내의 프로세스 또는 가상화된 형태로 배포된다. 따라서 기존 응용 서비스의 운영을 위해 개발된 다양한 관제 및 자동화 도구를 큰 변경 없이 적용하는 것이 가능하다는 장점이 있다. 게다가 화이트박스는 SDN/NFV 제어기를 활용하는 소프트웨어-정의 기반의 자동화된 네트워킹 관리, 관제를 통한 비용 및 운영효율성 측면의 장점을 갖고 있다.

그림 1. DANOS 기반 화이트박스 서버스위치

이러한 특장점으로 인해 페이스북, 구글, 마이크로소프트 등 거대 규모의 데이터센터를 자체 운영하는 기업들을 중심으로 이미 데이터센터 패브릭(fabric) 스위치를 화이트박스 기반으로 전환하여 수년간에 걸쳐 운영하고 있다. 통신사 또한 5G 네트워크를 구성하기 위한 화이트박스 기반 네트워킹 박스의 설계, 구현, 검증을 활발히 진행하였으며 근래에는 실제 서비스 환경에 투입되기 시작하였다. 일례로 미국의 통신사 AT&T는 5G 네트워크를 구성하기 위한 화이트박스 서버스위치의 리눅스 기반 오픈소스 네트워킹 운영체제인 DANOS(Disaggregated Network Operating System)를 개발하고 있다[4]. 그리고 DANOS를 활용하여 수십 Gbps 수준의 5G 대응 액세스(access) 네트워킹을 종단 사용자들에게 제공함에 있어서, 그림 1과 같은 화이트박스 서버스위치를 개발하여 셀 타워(cell tower)에 배치하는 단계이다[5].

소프트웨어 중심의 미래 전술망으로의 전환에 대비하여 국내에서도 설계 및 시험적인 검증이 필요한 상황이나, 안전성과 견실함이 중요한 전장환경의 특성상 화이트박스나 가상화 기술을 활용하는 네트워킹 노드를 개발하거나 실제 환경에서 검증하는 것이 쉽지 않은 상황이다. 따라서 본 논문에서는 미래형 전술망 구도의 타당성을 검증하기 위하여, 오픈소스 소프트웨어들을 활용하여 구성할 수 있는 시험적 구성의 네트워킹 노드를 설계한다. 이 때 전술메쉬망을 구성하는 네트워킹 노드로 데이터센터를 고려하여 개발된 화이트박스 서버스위치를 기반하는 것은 적합하지 않으므로, 미국 AT&T에서 주도하는 5G 대응 무선 환경과 연동하여 동작하도록 준비된 DANOS를 기반한 화이트박스 서버스위치를 기반으로 설계한다. 그리고 전술망의 동적인 QoS 관제(모니터링 및 제어)를 지원할 수 있도록, 리눅스 eBPF(extended Berkeley packet filter) 기반의 실시간 모니터링을 수행하는 네트워킹 노드를 설계하고, 이를 실제 실험을 통해 가능성을 검증한다.

Ⅱ. 미래형 전술메쉬망을 위한 네트워킹 노드

1. 트래픽 QoS 조율을 지원하는 미래형 전술메쉬망

본 논문에서는 그림 2와 표 1과 같은 오픈소스 소프트웨어들을 활용하여 구성할 수 있는 소프트웨어 중심의 미래형 전술메쉬망(Tactical Mesh WAN)을 고려한다. 수십 킬로미터에 걸쳐 산재한 화이트박스 기반의 전술 메쉬 WAN(wide area network) 노드들이 지향성 또는 섹터 안테나를 통해 서로 메쉬 구조로 연결되어 WAN을 구성한다. 이때 WAN 노드들은 장거리 무선 연결의 특성상 제한된 대역폭을 가지고, 전장 상황 변화에 따라 연결성이 불안정해지며, 때로는 릴레이 노드를 경유해서 연결되어야 한다. 따라서 전술망은 구조와 상태 전반에 걸쳐 동적인 변동성이 심한 제약을 가진다. 또한 응용 서비스 트래픽의 관제(즉, 모니터링 및 제어)를 기반한 지능형 QoS 조율을 수행하는 소프트웨어 프레임워크(framework) 수준의 접근 방식인 전술 서비스 메쉬(tactical service mesh, TSM)를 도입하고 있다[6].

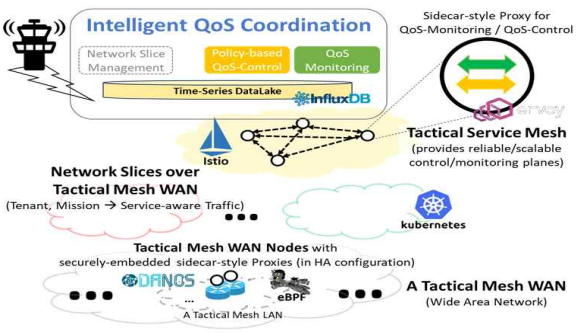

그림 2. 오픈소스 소프트웨어를 활용한 소프트웨어 중심 미래형 전술메쉬망의 시험적인 설계방안

* 오픈소스 커뮤니티를 통해 하드웨어 설계가 공개되어 있으며, 이를 위한 네트워킹 기능들은 상용/오픈소스 네트워크 운영체제로 개발됨. 공개된 하드웨어 설계를 기반하면 운영체제 호환성, 네트워킹 성능, 안정성 등을 보장하는 화이트박스 서버스위치를 누구나 제작 가능. 현재 다양한 벤더에서 하드웨어, 소프트웨어를 별도/통합 제공하고 있음.

표 1. 시험적인 미래형 전술메쉬망에서 활용하는 오픈소스 소프트웨어

계층	SW명	역할	웹사이트
관제타워	InfluxDB	시계열 모니터링 데이터 저장소	https://docs.influxdata.com/influxdb/v1.7/
TSM	Istio	관제타워에서 분산 프록시들을 하나의 TSM 계층으로 관리	https://istio.io/
	Envoy	네트워킹 노드 내 프록시로써 트래픽 모니터링 및 제어 수행	https://www.envoyproxy.io/
네트워크 슬라이스	Kubernetes	응용 서비스의 컨테이너 배포 지원	https://kubernetes.io/
네트워킹 노드	DANOS	화이트박스 서버스위치의 운영체제	https://www.danosproject.org/
	eBPF	리눅스 커널 내 경량 모니터링 프로그램 동적 실행	https://www.iovisor.org/technology/ebpf

전술 서비스 메쉬는 클라우드 기반 마이크로서비스 구조의 확산으로 주목받는 서비스 메쉬[7]를 전술망 환경에 맞추어 확장한 것으로, 네트워킹 인프라 계층과 응용 서비스 계층 사이에 추가적인 계층인 TSM 계층을 구성한다. TSM 계층은 지능형 QoS 조율을 담당하는 관제타워의 지시에 따라 응용 서비스를 구성하는 요소 기능(function)들의 트래픽이 인프라로 전달되기 전에 가로채 모니터링 및 제어하는 방식으로 응용 서비스별 QoS 보장을 지원한다.

2. 전술메쉬망을 구성하는 네트워킹 노드의 설계 요구사항

전술메쉬망을 구성하는 전술메쉬 WAN 노드는 응용 서비스 트래픽들을 위한 게이트웨이 역할로써, 트래픽을 목적지에 전송하도록 포워딩(forwarding)/라우팅(routing)하는 네트워킹 기능을 제공하는 것이 필요하다. 이 때 우선순위에 따른 차별화된 대우가 필요한 전술 응용 서비스 트래픽의 QoS를 보장하기 위해서는 전술메쉬망의 실시간 네트워킹 상태를 기반하는 동적인 QoS 관제(모니터링 및 제어) 방식이 효과적이다. 이와 같은 역할을 위해 네트워킹 박스는 아래 기술한 4가지 요구사항(R1-R4)을 고려하여 설계하는 것이 효과적이다.

- **R1. 고성능의 견실한 가상화 기반 네트워킹:** 전술메쉬 WAN 노드는 전술망에서 요구하는 네트워킹 노드 간 Gbps 급의 네트워킹 성능을 지원하면서 안정적인 트래픽 전송을 보장할 수 있는 무선 중심의 네트워킹 연결성을 확보하는 것이 필요하다. 이와 같은 연결성을 기반으로 포워딩/라우팅을 포함하는 네트워킹 기능들을 빠르고 유연하게 도입, 관리, 운용할 수 있도록 가상화된 형태로 구성한다. 이를 통해 SDN과 NFV 등 가상화 기반의 네트워킹 기술과 연계를 고려하는 것이 필요하다. 나아가 응용 서비스 종단간 네트워킹 경로에 위치하는 네트워킹 노드들의 컴퓨트/네트워킹 자원을 가상화 기술을 통해 분리하여 제공하는 종단간 네트워킹 슬라이싱(End-to-End Networking Slicing) 기술이 성숙되었을 때, 이를 적용한 보다 효과적인 전술메쉬망의 동적 QoS 관제를 지원할 수 있도록, 세분화된 데이터 평면(즉, 네트워크 슬라이스)마다 격리된 가상 스위치를 구성하는 것이 요구된다.

- **R2. 높은 수준의 보안성 강화:** 일반적인 네트워킹 인프라와 비교하여 높은 수준의 보안과 안정성이 요구되는 전술망에서 네트워킹 노드는 예상되는 다양한 공격에도 안전하고 견실한 운용을 보장하는 것이 매우 중요하다. 따라서 전술메쉬 WAN 노드의 정상동작을 위해 실행이 보장되어야 할 핵심 구성요소들에 대해 높은 수준의 격리 기술을 적용하여 보안 공격들로부터 보호하는 것이 필요하다. 또한 전술망 전체의 동적인 QoS 관제를 지원하는 논리적인 관제타워와의 네트워킹은 네트워킹 노드의 동작방식에 영향을 미치는 매우 중요한 메시지를 송수신한다. 따라서 네트워킹 노드와 관제타워 간 안전한 네트워킹을 제공하는 것이 필요하다.

- **R3. 실시간 트래픽 모니터링 및 컨디셔닝 지원:** 전술메쉬 WAN 노드는 논리적인 관제타워의 모니터링과 제어를 연계하는 동적인 QoS 관제를 지원하기 위해 트래픽의 패킷/플로우를 실시간으로 수집하고 논리적인 관제타워에 제공하는 실시간 모니터링을 수행해야 한다. 동시에 관제타워의 QoS 제어 메시지를 수신하면, 이에 따라 패킷들을 태깅(tagging)하여 식별한 트래픽에 대한 컨디셔닝(conditioning)을 수행할 수 있어야 한다.

- **R4. 전술 서비스 메쉬 지원:** 네트워킹 노드는 그림 2과 같이 목표한 전술메쉬망에 추가적인 계층을 형성하는 전술 서비스 메쉬를 지원하는 것이 필요하다. 이를 위해 각 노드마다 TSM 대응 프록시(proxy)를 포함하고, 이를 인접한 노드들과 연결하여 메쉬 구조 형태의 TSM 계층을 구성하는 것이 필요하다. 또한 논리적인 관제타워의 지능형 QoS 조율에 따라 QoS 관제를 수행하는 실무 계층으로 동작할 수 있어야 한다.

Ⅲ. 동적 QoS 관제를 지원하는 전술메쉬 WAN 노드의 설계

본 절에서는 앞서 기술한 요구사항들을 기반으로 오픈소스 소프트웨어들을 기반한 전술메쉬 WAN 노드의 개념수준 설계를 제시한다. 제안하는 전술메쉬 WAN 노드의 구성은 기존의 폐쇄적인 네트워킹 노드(스위치, 라우터 등)와는 달리 리눅스 기반의 개방형 화이트박스 서버스위치를 중심으로, 전 세계적인

영향력을 바탕으로 혁신적인 속도로 발전하고 있는 SDN, NFV, TEE(trusted execution environment), eBPF와 같은 최신 기술들을 리눅스를 중심으로 통합 활용하도록 설계한다.

동일한 기술의 구현을 대상으로 개발된 복수의 오픈소스 소프트웨어들은 접근방식의 차이로 인해 세부적인 설계/구현이 다를 수 있으나, 소프트웨어 자체가 제공하는 기본 기능은 동일하다. 본 논문에서는 높은 수준의 오픈소스 개방성, 빠른 기술성숙, 영향력을 고려하여 Linux Foundation 하에서 개발하는 DANOS 화이트박스 서버스위치와 ACRN Hypervisor, eBPF를 참고하여 전술메쉬 WAN 노드의 각 구성요소를 설계하나, 이에 대응하는 다른 오픈소스 소프트웨어로 대체하는 것도 가능하다.

1. 전술메쉬 WAN 노드의 개념 설계

본 논문에서 가정하는 시험적인 전술메쉬망에서 전술메쉬 WAN 노드가 제공해야할 기능들과 내부 구성, 그리고 이들 간의 관계를 개념수준으로 정리한 설계 방안을 그림 3과 같이 나타내었다.

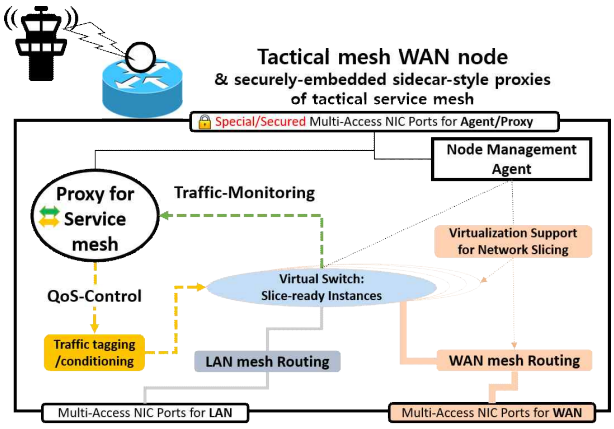

그림 3. 전술메쉬 WAN 노드의 개념수준 설계

우선 전술메쉬 WAN 노드의 주요 구성요소는 노드관리 에이전트(node management agent)와 TSM 대응 프록시(proxy for service mesh)이다. 에이전트는 전술메쉬 WAN 노드가 화이트박스 서버스위치로써 안정적으로 동작할 수 있도록, 노드의 내부를 구성하는 가상화 개체의 생성, 상태확인, 업데이트, 제거, 복구 등의 단계들을 포함하는 생애주기(lifecycle)을 관리한다. 노드 내 가상화 개체는 가상 스위치와 네트워킹 슬라이싱을 위한 가상머신, 응용 서비스의 요소기능을 포함하는 컨테이너 또는 가상머신, 그리고 TSM 대응 프록시들이 해당된다.

TSM 대응 프록시는 미래형 전술메쉬망의 서비스메쉬를 적용하고자 네트워킹 박스 내에 사이드카 형태로 부착되는 소프트웨어 모듈이다. 분산 네트워킹 노드의 프록시들은 메쉬 구조로 네트워킹 연결되며, 논리적인 관제타워를 통해 관리됨으로써 하나의 추가적인 계층인 전술 서비스 메쉬 계층을 구성하므로 요구사항 R4를 만족할 수 있다. 프록시는 노드를 통해 송수신되는 트래픽을 모니터링 도구들을 활용하여 수집 및 관제타워로 전송하는 모니터링 실무 작업을 수행한다. 또한 관제타워의 QoS 제어메시지에 따라 가상스위치를 대상으로 트래픽 태깅(tagging) 및 컨디셔닝(또는 조절)한다.

동적인 QoS 관제를 수행하는 논리적인 관제타워는 분산된 전술메쉬 WAN 노드의 프록시와 에이전트와의 상호작용을 통해 전술메쉬망 전체의 응용 서비스 QoS와 네트워킹 노드들의 상태를 관리한다. 전술메쉬 WAN 노드의 다른 구성요소들의 역할 및 설계는 다음 절에서 설명한다.

2. DANOS를 활용한 가상 기반 네트워킹

Linux Foundation의 DANOS는 화이트박스 서버스위치는 그림 4와 같이 응용 관리 및 API를 제공하는 제어/관리 평면, 박스 자원(CPU, RAM 등)을 관리하는 기본 운영체제, 스위치 하드웨어의 추상화 계층을 제공하고 패킷 처리 로직을 소프트웨어로 구현한 데이터 평면으로 구성된다. DANOS는 패킷 라우팅, 방화벽, 트래픽 모니터링 등의 네트워킹 기능들을 응용 서비스로 실행함으로써 기능의 신속한 개선 또는 도입이 가능하다. 또한 범용 리눅스 서버 박스를 대상으로 개발된 클라우드, SDN, NFV, 네트워크 슬라이싱, 인프라 자동화 기술의 적용이 용이하다.

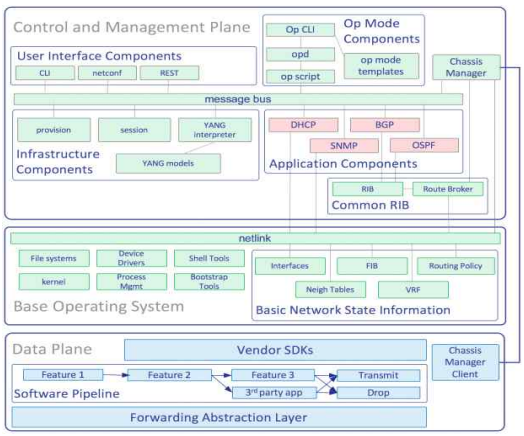

그림 4. Linux Foundation의 DANOS 소프트웨어 구조

전술메쉬 WAN 노드는 화이트박스 서버스위치의 유연성, 개방성과 함께 관련사례를 통해 검증된 고성능, 안정성, 운영효율성의 특장점을 내재화하기 위해, 다수의 유/무선 멀티액세스 네트워킹 인터페이스를 장착하고 Tbps 급으로 설계된 리눅스 기반 네트워킹 운영체제를 탑재한 DANOS 화이트박스 서버스위치를 참조하여 설계한다. 이를 통해 전술망 환경에서 요구하는

성능(Gbps급)을 무난하게 제공할 수 있다고 예상되며, 무기체계/전술제대 별로 상이할 수 있는 이종의 네트워킹을 호환성 있게 지원하기 위한 하드웨어 확장이나 QoS 조율을 위한 다양한 리눅스 기반의 소프트웨어 기능들을 쉽게 수용하는 유연성을 제공할 수 있다.

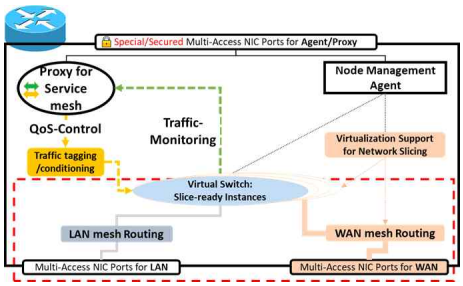

그림 5. 전술메쉬 WAN 노드 개념설계의 가상화 기반 네트워킹 (요구사항 R1)

전술메쉬 WAN 노드를 DANOS 기반의 화이트박스 서버스위치를 기반으로 설계하면 그림 5에 빨간색 박스로 표기된 가상화 기반 네트워킹을 위한 기능들을 확보할 수 있다. 참조한 DANOS의 데이터 평면 계층은 이종의 네트워킹 인터페이스에 대한 추상화와 패킷 처리로직에 대한 소프트웨어 구현이 가능하도록 지원하여, 화이트박스 서버스위치에 장착된 멀티-액세스 네트워킹 인터페이스들이 종류와 관계없이 동작할 수 있다. 또한 DANOS의 리눅스 기반 운영체제 계층은 SDN/NFV와 같은 가상화 기술을 지원하도록 설계되어 내부적으로 복수의 가상스위치들을 생성할 수 있으며, 이 때 노드에 장착된 네트워킹 인터페이스 별로 네트워크 슬라이스들을 위한 독점적인 가상스위치를 연결하여 활용함으로써, 해당 슬라이스를 통해 노드 내로 유입되는 트래픽들에 대한 기본적인 포워딩 기능을 수행한다. 그리고 DANOS에서 제어/관리 평면 내에 구현된 DHCP, SNMP, BGP, OSPF 등의 응용 요소들은 가상스위치와의 연계를 통해 네트워킹 노드를 위한 WAN/LAN mesh routing 기능을 위해 활용한다.

네트워킹 슬라이스들의 실 구현 측면을 고려하였을 때, 한 노드 안에 생성되는 복수의 가상 스위치는 논리적으로는 구분되는 개체이나 운영체제는 이를 단일 프로세스로 관리하므로 과도한 트래픽, 의도적인 공격 등으로 인해 특정 가상 스위치에서 장애가 발생하는 경우 모든 가상 스위치에 치명적인 영향을 미치게 된다. 따라서 단일 프로세스 기반의 가상 스위치들로 응용서비스 별로 독점적으로 할당해야할 네트워킹 슬라이스를 제공하는 것은 구현 측면에서 어려운 상황이다. 그러므로 전술메쉬 WAN 노드는 식별된 트래픽 별 슬라이스마다 가상 스위치를 포함한 가상머신을 생성함으로써 가상 스위치들을 독립적으로 분리하는 방식으로 요구사항 R1에 명시된 네트워킹 슬라이싱을 기술을 지원하는 것이 가능하다.

3. TEE 기반 격리구역을 활용하는 보안강화

전술메쉬 WAN 노드의 정상 동작을 보장하기 위해서는 요구사항 R2에서 명시한 것과 같이, 노드 내의 트래픽 모니터링 및 컨디셔닝을 수행하는 프록시와 가상화 요소들의 관리를 수행하는 에이전트의 안전한 운용이 전제되어야 한다.

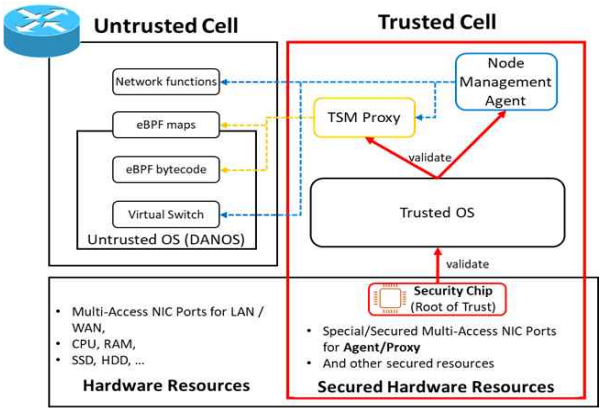

그림 6. 전술메쉬 WAN 노드의 TEE 기반 보안강화 설계

그림 6과 같이 전술메쉬 WAN 노드는 하드웨어 보안칩으로부터 단계별로 검증받은 소프트웨어만 실행을 허용하는 신뢰적인 실행환경(trusted execution environment, TEE)[8] 기술을 통해 프록시와 에이전트를 안전하게 격리된 환경(trusted cell) 내에 배치한다. TEE 기술은 인가되지 않은 사용자의 소프트웨어 접근을 원천적으로 차단하며, 하드웨어 자원 수준의 격리를 통해 일반 환경(즉, 비격리 구역) 내 가상 스위치 및 LAN/WAN 네트워킹 인터페이스에서 발생한 장애의 영향을 받지 않는다. 따라서 일반 환경에 장애가 발생하더라도 프록시와 에이전트의 안정성을 보장 가능하며, 나아가 에이전트가 장애를 탐지하고 복구하는 기능을 제공하면 높은 안정성을 보장할 수 있다.

동시에 관제타워와의 안전하고 안정적인(secured and reliable) 네트워킹을 위해, 관제타워와 프록시/에이전트 간 네트워킹 트래픽이 전송되는 모니터링/제어 평면을 서비스 트래픽이 흐르는 데이터 평면와 분리하기 위한 별도의 멀티액세스 네트워킹 인터페이스를 확보하고, 관제타워와의 암호화된 네트워킹을 수행한다. 이 때 해당 인터페이스들은 TEE 기반의 격리된 실행환경에 독점적으로 할당하여, 격리된 환경에 위치한 프록시/에이전트만 접근 및 활용하도록 제한한다. 이를 통해 프록시/에이전트는 외부로부터 격리되어 위/변조 공격으로부터 안전하며, 데이터 평면 상의 트래픽 혼잡, 과도한 부하, 물리적인 장애 등이 발생하더라도 관제타워와 송수신하는 모니터링/제어 트래픽의 성능 및 안정성을 보장할 수 있다. 따라서 TEE

기반의 보안성 강화를 통해 요구사항 R2를 충족 가능하다.

이와 같은 TEE를 적용한 전술메쉬 WAN 노드는 오픈소스 TEE 소프트웨어인 Linux Foundation의 ACRN hypervisor[9] 또는 Jailhouse[10] 등을 활용하여 구성 가능하다. 일례로 ACRN hypervisor는 그림 7에 나타난 것과 같이 인텔 CPU 기반의 범용 서버 노드 내에 안전한 격리 구역인 safety/security critical domain을 확보하고 하드웨어 칩으로부터 검증된 trusted OS에 대응하는 service OS를 구성한다. 그리고 service OS를 통해 비격리 구역 상에 일반 운영체제 환경인 user OS를 구성, 모니터링, 제거하는 관리를 지원한다. 따라서 ACRN hypervisor를 화이트박스 서버스위치에 적용하여 service OS상에 TSM 프록시, 에이전트를 실행하고 이를 위한 모니터링/제어 평면 용 네트워킹 인터페이스를 할당한다. 그리고 service OS를 통해 DANOS를 비격리 구역에 구성 및 관리하는 방식으로 전술메쉬 WAN 노드의 TEE 환경을 구성할 수 있다.

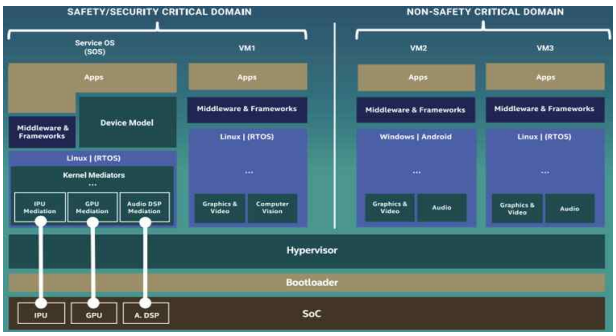

그림 7. Linux Foundation의 ACRN Hypervisor

IV. 전술메쉬 WAN 노드의 리눅스 eBPF 기반 실시간 네트워킹 모니터링

본 절에서는 지능형 QoS 조율을 지원하기 위한 전술메쉬 WAN 노드의 네트워킹 인프라 모니터링 및 제어 방법을 기술한다. 구체적으로 전술메쉬 WAN 노드는 리눅스 eBPF 기반의 단일화된 방식을 통해 기존 네트워킹 박스의 대표적인 관제도구들의 기능을 포괄한다. 그리고 관제타워의 요청에 따라 패킷/플로우 모니터링 및 제어의 범위 및 방법을 동적으로 변경 가능한 유연한 관제를 지원한다.

1. 리눅스 eBPF를 활용한 실시간 모니터링

리눅스 eBPF(extended Berkeley packet filter)는 1990년대 유닉스 서버 박스를 대상으로 실시간 패킷 수집을 위해 개발된 BPF(Berkeley packet filter)[11]를 현대 컴퓨터 구조에 맞추어 확장 개선한 기술이다. 기존 패킷 필터링으로 제한되어 있던 기능을 크게 확장하여 네트워킹 뿐만 아니라 컴퓨트(CPU, 프로세스 등), 스토리지(디스크, RAM 등)을 대상으로 한 모니터링, 트레이싱, 분석, 보안 등을 손쉽게 구현하여 적용할 수 있는 기반을 제공하고 있다.

리눅스 eBPF는 그림 8에 도식화된 것과 같이 유저공간(user space)에서 C 언어로 작성한 eBPF 프로그램 코드를 컴파일하여 eBPF의 저수준 명령어 집합인 바이트코드(bytecode)를 생성한다. 생성한 바이트코드는 커널에 주입되어 eBPF verifier를 통해 바이트코드의 정상 실행 여부, 안전성 등을 검증한 후 가상 머신에 동적으로 주입된다. 주입된 eBPF 코드는 리눅스 내 다양한 이벤트에 대응하는 kprobes, tracepoints 등의 훅(Hook)에 부착되어, 대상 이벤트가 발생할 때마다 시 자동으로 실행된다[13]. 커널 내부의 eBPF 코드와 이를 주입한 유저공간의 프로세스는 eBPF map이라고 하는 테이블 구조의 저장소를 공유하며, 이를 통해 유저공간에서 커널 내로 설정 정보를 전달하거나 반대로 커널에서 모니터링 등을 통해 생성한 결과 데이터를 유저공간으로 전달하는 방식으로 동작한다. 리눅스 eBPF는 기존 BPF의 지원 기능을 크게 확장하여 네트워킹 뿐만 아니라 CPU, 메모리, 디스크 등의 컴퓨트/스토리지 자원에 대한 커널 이벤트로 확장된 훅들을 확보하였다.

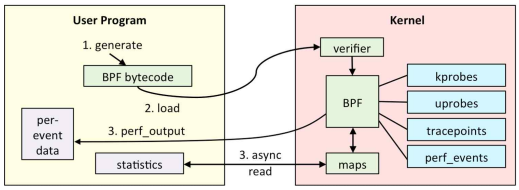

그림 8. Linux eBPF의 동작방식[12]

리눅스 eBPF를 기반으로 구현한 모니터링 프로그램은 지정한 물리/가상 네트워킹 인터페이스에 부착되어, 해당 인터페이스가 패킷을 송수신하는 저수준의 커널 이벤트 발생 시 자동으로 커널 내에서 실행된다. 한편 리눅스 eBPF의 특장점에 주목하여 트래픽 태깅/컨디셔닝 프로그램도 eBPF 기반으로 전환 또는 개발하고 있는 추세이다. 이러한 eBPF 프로그램은 4096개 이하의 저수준 명령어들을 포함하는 제한된 크기의 경량 프로그램으로 적은 CPU와 RAM 자원을 활용하며, 또한 저수준의 커널에서 바로 실행되므로 신속한 모니터링 및 컨디셔닝이 가능하다. 따라서 많은 양의 네트워킹 트래픽이 발생하는 상황에서도 네트워킹 또는 응용 서비스에 영향을 미치지 않는 실시간 모니터링이 가능하다.

2. 리눅스 eBPF 기반 실시간 모니터링 설계

전술메쉬 WAN 노드의 eBPF 기반 실시간 모니터링을 위한 내부 구성을 그림 9와 같이 설계한다. TEE를 통해 격리된 구역에서 동작하는 TSM 대응 프록시는 eBPF 프로그램들을 템플

릿 형태로 보유하고 있으며, 커널 내부에 이를 주입하고 물리 네트워킹 인터페이스, 가상스위치 등의 자원들과 관련된 socket, tracepoints 등의 훅들에 부착시킨다. 패킷의 송수신 이벤트들이 발생할 때마다 부착된 경량의 eBPF 바이트코드는 커널의 저수준에서 실행되어 매개변수로 전달된 패킷의 raw 데이터를 바이트 연산을 통해 필요한 헤더 정보들을 실시간으로 수집하여 map을 통해 TSM 대응 프록시로 전달한다. 반대로 프록시는 수집 메트릭 및 샘플링 주기를 map을 통해 전달함으로써, eBPF 바이트코드의 모니터링 방법을 실행 중에 실시간으로 조정하는 것이 가능하다.

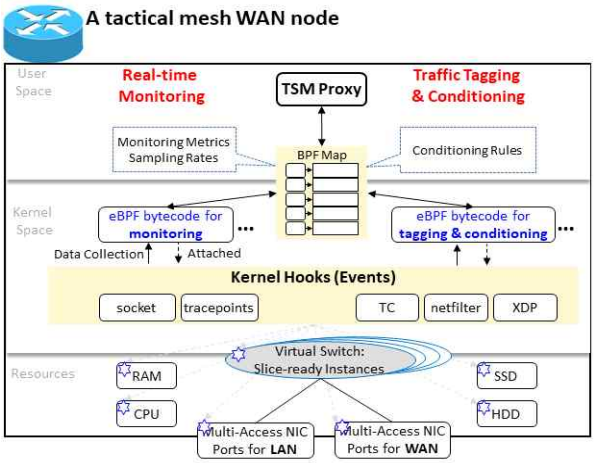

그림 9. 전술메쉬 WAN 노드를 위한 eBPF 기반 실시간 모니터링 설계

한편 동적인 QoS 제어를 지원하기 위하여 전술메쉬 WAN 노드는 eBPF 기반의 트래픽 태깅 및 컨디셔닝을 그림 9의 우측과 같이 모니터링과 유사한 방법으로 수행한다. 이 때 eBPF 바이트코드는 리눅스의 트래픽 태깅/컨디셔닝이 가능한 훅(hook)들에 부착된다. 이 훅들은 패킷의 고성능 데이터평면을 eBPF로 구현하는 XDP[14], 패킷의 필터링/포워딩 등을 수행하는 bpfilter[15], 플로우 수준의 트래픽 조절(conditioning)/우선순위 조정/필터링 등을 수행하는 TC(traffic control)[16]이 해당된다. TSM 프록시는 관제타워로부터 수신한 QoS 제어메시지를 기반으로 트래픽 태깅/컨디셔닝을 위한 세부적인 규칙을 map을 통해 전달하여 바이트코드의 동작방식을 조정한다.

앞 절을 통해 설명하였듯 eBPF 기반으로 구현된 프로그램의 경량성으로 인해, 그림 9의 설계와 같이 전술메쉬 WAN 노드를 구성하면 그림 10에 표기한 실시간 모니터링 및 트래픽 컨디셔닝을 수행하는 것이 가능하다. 게다가 관제타워와의 상호작용을 통해 전술메쉬 WAN 노드의 상황에 따라 모니터링 및 트래픽 컨디셔닝 방식을 조정하는 실시간성도 제공한다.

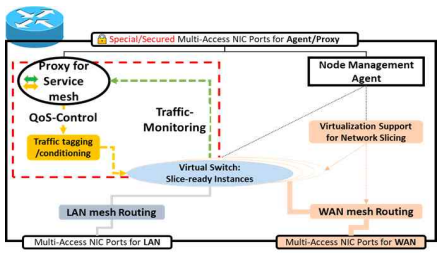

그림 10. 전술메쉬 WAN 노드 개념설계의 실시간 모니터링 (요구사항 R3)

3. 실시간 모니터링 가능성 검증

본 절에서는 제안한 eBPF 기반 전술메쉬 WAN 노드의 실시간 모니터링의 가능성 검증을 제시한다.

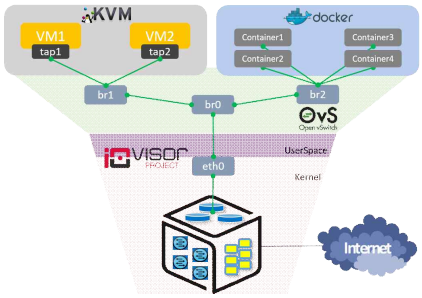

그림 11. eBPF 기반 실시간 모니터링 검증을 위한 노드 구성

우선 eBPF를 기반한 가상머신/컨테이너 형태로 배포되는 네트워킹 기능과 가상스위치에 대한 실시간 모니터링 검증을 위해 6 코어 Intel Xeon-D CPU와 32GB RAM을 포함한 범용 서버노드 내에 그림 11과 같이 가상머신과 컨테이너가 가상스위치를 통해 연결되어 있는 실험환경을 구성한다.

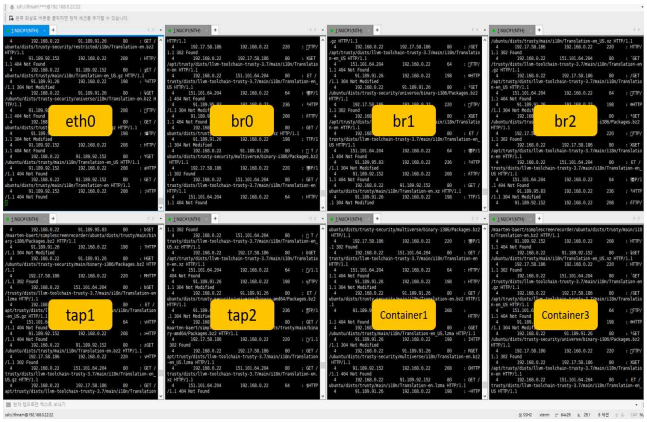

그림 12. 물리/가상/컨테이너 네트워킹 인터페이스의 실시간 모니터링

상기 실험환경에서 동일한 eBPF 기반의 패킷 모니터링 프로그램을 가상스위치, 물리/가상/컨테이너 네트워킹 인터페이스

들에 모두 부착하고, 패킷을 발생시켰을 때 실시간 모니터링의 가능 여부를 검증한다. 검증을 위해 일반적인 네트워킹 환경에서 보편적으로 발생하여 전술망 환경에서도 부분적으로 활용될 것으로 예상되는 HTTP 트래픽을 지속적으로 발생시켰다. 그리고 eBPF 기반 모니터링 프로그램은 패킷에 포함된 정보 중 IP 버전, SRC/DEST IP 주소(L3 헤더), 포트번호(L4 헤더), HTTP 요청 URL/응답 코드(L7 페이로드)를 수집하도록 구현하였다. 각 그림 12에서 볼 수 있듯 eBPF 기반 모니터링은 물리/가상/컨테이너와 관계없이 부착가능하며, 실시간으로 전송되는 트래픽의 모니터링이 가능함을 확인할 수 있다[17].

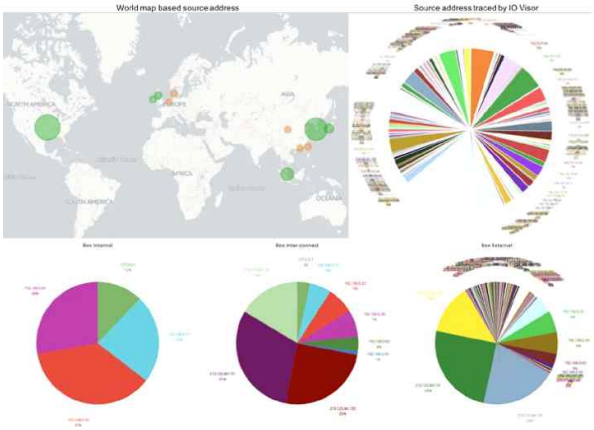

그림 13 eBPF 네트워킹 모니터링을 기반한 관제타워의 실시간 가시화 가능성 검증결과

또한 앞서 수행한 단일 노드를 대상으로 검증한 실시간 모니터링의 현실적인 실용가능성을 검토하기 위하여, 그림 11과 같은 구성의 노드를 4개 준비하고, 논리적인 관제타워를 시계열 데이터베이스와 가시화 도구를 포함하도록 구성하여 노드들의 eBPF 모니터링을 통해 수집한 데이터를 그래프로 가시화하여 제공한다. 이 때 모니터링/제어 트래픽을 데이터 트래픽과 분리하기 위해 각 노드들과 관제타워 간에는 모니터링/제어평면을 위한 별도의 네트워킹 인터페이스를 연결하였다. 그림 13의 하단에 나타난 그래프는, 4대의 노드들로부터 eBPF를 통해 수집한 데이터를 단일 노드 내 가상머신/컨테이너 간 네트워킹, 다른 노드에 위치한 가상머신/컨테이너 간 네트워킹, 그리고 외부 네트워크에 위치하는 IP 주소로 통신한 경우의 3가지로 구분하여 송수신 IP의 등장빈도를 기준으로 파이차트 형태로 실시간 가시화 한다. 그리고 3가지 경우를 모두 포함하는 그래프는 우측 상단에 나타내었다. 마지막으로 실시간 모니터링 데이터의 IP 주소를 기반으로 네트워킹을 수행한 두 종단의 물리적인 위치와 빈도를 지도상에 표기하는 검증을 수행하였다. eBPF 기반 실시간 모니터링 기능을 적용한 전술메쉬 WAN 노드와 논리적인 관제타워를 연계하면 본 검증과 같이 네트워킹 전체의 상황을 실시간으로 확인할 수 있는 효과적인 모니터링이 가능함을 확인하였다[18].

상기 두 검증을 통해 물리/가상/컨테이너의 네트워킹 인터페이스를 대상으로 eBPF 기반의 실시간 모니터링이 가능하며, 논리적인 관제타워로 이를 수집하는 경우에도 모니터링/제어평면이 분리되어 있으면 실시간 모니터링을 가시화와 연계하여 전술망 전체의 상태를 실시간을 확인하는 효과적인 모니터링이 가능함을 확인하였다.

V. 결론

본 논문은 소프트웨어 중심의 미래형 전술망으로의 전환을 대비하는 오픈소스 소프트웨어를 활용하는 소규모의 검증 가능한 연구를 위하여, 미래형 전술망의 단순화된 네트워킹 노드인 전술메쉬 WAN 노드를 개념화 수준에서 설계하였다. 전술메쉬 WAN 노드는 가상화된 네트워킹 기능을 제공하기 위해 DANOS를 기반한 화이트박스 서버스위치로 설계하고, 보안성 강화를 위해 ACRN hypervisor를 기반한 Trusted Execution Environment를 제공하도록 설계하였다. 또한 미래 전술망의 논리적인 관제타워를 중심으로 한 동적인 QoS 관제를 지원하기 위해, 리눅스 eBPF를 활용한 실시간 모니터링 및 트래픽 태깅/컨디셔닝을 수행하는 전술메쉬 WAN 노드를 설계하고, 부분적인 검증을 통해 eBPF 기반 실시간 네트워킹 모니터링의 구현에 대한 접근방안에 대해 확인하였다.

본 논문의 전술메쉬 WAN 노드와 이를 위한 eBPF 모니터링은 개념수준의 설계만을 제한적으로 제시하고 있어, 이를 실체화하기 위한 보안, 안정성, 성능 측면의 지속적인 연구개발이 필요한 상황이다. 우선 오픈소스 소프트웨어들을 통합함으로 인해 발생할 수 있는 보안 위협 및 성능 감소의 고려가 필요하며, 관제타워의 데이터분석을 기반으로 QoS 모니터링과 제어를 연계하는 기술, 그리고 이를 활용하였을 때의 실시간 제어를 위한 알고리즘 등의 연구가 수행되어야 한다. 나아가 전술 WAN 환경의 안전성을 제고하기 위해 전술메쉬 WAN 노드의 트래픽 모니터링 및 컨디셔닝을 보안 관제 기술[19, 20, 21]과 연계하는 연구도 필요하다. 이와 같이 산재한 향후 과제들을 단계별로 해결하고자 단기목표로 본 논문에서 제시한 설계안에 따라 전술메쉬 WAN 노드의 세부적인 설계 및 시제품 수준으로 구현하는 연구를 수행할 계획이다.

REFERENCES

[1] A. K. Cebrowski and J. J. Garstka, “Network-centric warfare: Its origin and future,” *US Naval Institute Proceedings Magazine*, vol. 124, no. 1, pp. 28-35, Jan. 1998.
[2] Warfighter Information Network-Tactical (WIN-T), https://gdmissionsystems.com/en/communications/warfighter-information-network-tactical (accessed May 03, 2019)

[3] PICA8, “Bare Metal Networking - Leveraging White Box Thinking,” *PICA8 White paper*.
[4] AT&T, “Towards an Open, Disaggregated Network Operating System,” *DANOS White Paper*, 2017.
[5] AT&T Releases OCP specifications for its White Box cellular gateway routers (2018), https://www.telecomtv.com/content/white-boxes-merchant-silicon/at-t-submits-specifications-for-white-box-cell-site-gateway-routers-to-power-5g-era-32567/ (accessed May 03, 2019)
[6] 강문중, 신준식, 박주만, 박찬이, 김종원, “미래 전술망의 지능적 트래픽 QoS 조율을 위한 전술 서비스 메쉬,” *한국군사과학학회지* (심사중)
[7] Lyft’s Envoy: Experiences Operating a Large Service Mesh (SREcon2017, 2017), https://www.usenix.org/sites/default/files/conference/protected-files/srecon17americas_slides_klein.pdf (accessed May 03, 2019)
[8] Mohamed Sabt, Mohammed Achemlal, and Abdelmadjid Bouabdallah, “Trusted Execution Environment: What It is, and What It is Not,” *Proc. the 14th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (IEEE TrustCom-15)*, Aug. 2015.
[9] ACRN: A Big Little Hypervisor for IoT Development (2018), https://projectacrn.org/wp-content/uploads/sites/59/2018/05/ACRN-Overview_v9.pdf (accessed May 03, 2019)
[10] Valentine Sinistyn, “Jailhouse,” *Linux Journal*, vol. 2015, no. 252, June 2015.
[11] S. Mccanne and V. Jacobson, “The BSD Packet Filter: A New Architecture for User-level Packet Capture,” *Proc. the USENIX Winter 1993 Conference*, San Diego, CA, Jan. 1993.
[12] Linux Extended BPF (eBPF) Tracing Tools (2016), http://www.brendangregg.com/ebpf.html (accessed May 03, 2019)
[13] Linux Socket Filtering aka Berkeley Packet Filter (BPF) (2014), https://www.kernel.org/doc/Documentation/networking/filter.txt (accessed May 03, 2019)
[14] T. Høiland-Jørgensen, et al. "The eXpress data path: fast programmable packet processing in the operating system kernel." *Proc. the 14th International Conference on emerging Networking EXperiments and Technologies*. ACM, Dec. 2018.
[15] Linux tc and eBPF (FOSDEM2016, 2016), https://archive.fosdem.org/2016/schedule/event/ebpf/attachments/slides/1159/export/events/attachments/ebpf/slides/1159/ebpf.pdf (accessed May 03, 2019)
[16] BPF comes to firewalls (2018), https://lwn.net/Articles/747551/ (May 03, 2019)
[17] 남택호, 김종원, “네트워크 인터페이스에 선별적으로 적용하는 IO Visor 기반의 SmartX Box 패킷 트레이싱,” *제27회 통신정보 합동학술대회 (JCCI 2017)*, 2017. 04
[18] T. Nam and J, Kim, “Prototype Implementation of Site Visibility Framework employing IO Visor-based Packet Tracing,” *Proc. 12th KIPS International Conference on Ubiquitous Information Technologies and Applications (CUTE 2017)*, Dec. 2017.
[19] 남승수, 서창호, 이주영, 김종현, 김익균, “통합 사이버 보안 상황분석을 통한 관제 상황인지 기술,” *스마트미디어저널*, 제4권, 제4호, 80-86쪽, 2015년 12월
[20] 차병래, 박선, 김종원, “사이버 탄력성 기반 가상 허니팟 서비스 프레임워크 구상 및 가능성 검증,” *스마트미디어저널*, 제5권, 제2호, 65-76쪽, 2016년 6월
[21] 차병래, 최명수, 강은주, 박선, 김종원, “Cybersecurity를 위한 SOC & SIEM 기술의 동향,” *스마트미디어저널*, 제6권, 제4호, 41-49쪽, 2017년 12월

저 자 소 개

신준식 (학생회원)
2014년 아주대학교 정보및컴퓨터공학과 학사
2016년 광주과학기술원 정보통신공학부 석사
2016년-현재 광주과학기술원 전기전자컴퓨터공학부 박사과정
<주관심분야: 클라우드 기반 Secured & Composable 인프라 기술>

강문중
2016년 창원대학교 정보통신공학과 학사
2019년 광주과학기술원 정보통신공학부 석사
<주관심분야 : 클라우드 컴퓨팅>

박주만
2009 경남대학교 전자공학과 학사
2011 경북대학교 전자공학과 석사
2012~ 현재 국방과학연구소 연구원
<주관심분야 : 전술통신, 신뢰성 및 M&S>

권대훈
1999년 경북대학교 컴퓨터공학과 학사
2002년 경북대학교 컴퓨터공학과 석사
2002년-현재 국방과학연구소 선임연구원
<주관심분야: 군 전술통신, MANET, 무선통신>

김종원

1987: 서울대학교 제어계측공학과 학사
1989: 서울대학교 제어계측공학과 석사
1994: 서울대학교 제어계측공학과 박사
1994-1999: 공주대학교 전자공학과 조교수
1998-2001: 미국 Univ. of Southern California, EE-System Department 연구조교수
2001-현재 광주과학기술원 전기전자컴퓨터공학부 교수
2008-현재 광주과학기술원 슈퍼컴퓨팅센터 센터장

<주관심분야: Networked Computing System focusing on "Dynamic & Resource-aware Composition of Media-centric Service employing Programmable / Virtualized Computing / Networking Resources">