

# 지능형 SDN 제어 구조를 채용한 다계층 미래전술네트워크를 위한 Riverbed Modeler M&S 프레임워크 설계 및 구현

이철웅\*, 이규민\*, 이호진\*\*, 노병희<sup>o</sup>

## Design and Implementation of Riverbed Modeler M&S Framework for Multi-Layered Future Tactical Networks with Intelligent SDN Control Architecture

Cheol-woong Lee\*, Gyu-min Lee\*, Hojin Lee\*\*, Byeong-hee Roh<sup>o</sup>

### 요 약

SDN (Software Defined Networking)은 네트워크를 제어와 데이터 평면으로 분리하여, 제어 평면의 컨트롤러가 데이터 평면의 스위치들의 동작을 동적으로 제어할 수 있도록 한다. 이질적이고 복잡한 장치와 체계들로 구성되는 전술네트워크에서도 SDN 구조를 채용하려는 노력이 이루어지고 있다. 전술네트워크는 다계층의 복잡한 구조를 갖으므로 SDN을 연계한 Modeling and Simulation (M&S) 환경을 구성하기가 용이하지 않다. 본 논문에서는 지능형 SDN 제어 구조를 채용한 다계층 미래전술네트워크를 위한 M&S 프레임워크를 제안한다. M&S 도구는 전술 네트워크 성능분석에 효과적인 것으로 검증된 Riverbed Modeler를 사용한다. 제안한 M&S 프레임워크를 구현하였고, 실험 결과는 이의 효과성을 보여준다.

**Key Words** : Future Tactical Network, Software Defined Networking, Multi-layered Architecture, Modeling and Simulation, Riverbed Modeler

### ABSTRACT

Software Defined Networking (SDN) divides the network functions into control and data planes so that a controller in the control plane can dynamically control the operations of switches in the data plane. Efforts to adopt the SDN structure for tactical networks composed of heterogeneous and complex devices and systems are being made. Since tactical networks have multi-layered complex structures, it is not easy to construct Modeling and Simulation (M&S) environments supporting SDN structure. This paper proposes an M&S framework for multi-layered future tactical networks employing intelligent SDN control architectures. As an M&S tool, we utilize Riverbed Modeler, which is widely used and proven for tactical network performance analysis. The proposed M&S framework is implemented, and the experimental results show its effectiveness.

※ 본 연구는 방위사업청과 국방과학연구소가 지원하는 미래전투체계 네트워크기술 특화연구센터 사업의 일환으로 수행되었습니다.(UD190033ED)

• First Author : Dept. of AI Convergence Network, Ajou University, cjdnd369@ajou.ac.kr, 학생회원

<sup>o</sup> Corresponding Author : Dept. of AI Convergence Network and Dept. of Computer Engineering, Ajou University, bhroh@ajou.ac.kr, 중신회원

\* Dept. of Computer Engineering, Ajou University, mybrand@ajou.ac.kr, 학생회원

\*\* Dept. of AI Convergence Network, Ajou University, hj2913@ajou.ac.kr

논문번호 : 202205-083-0-SE, Received April 29, 2022; Revised June 22, 2022; Accepted June 27, 2022

## I. 서 론

미래 전술네트워크는 네트워크 중심전 (NCW: Network Centric Warfare)을 지향한다. 전술네트워크는 각종 유·무인 단위전투 요소로서 이기종 디바이스들로 구성된다. 지상, 공중, 우주 공간을 활용하는 미래 전술네트워크는 모든 전투 요소들이 연합 하여 합동작전을 수행할 수 있도록 지속적이고 협력적으로 지휘통제 임무를 계획하고 작전을 관리할 수 있는 능력을 갖추어야 한다<sup>1)</sup>.

소프트웨어 정의 네트워킹 (SDN: Software Defined Networking)은 네트워크를 제어와 데이터 전달 평면으로 분리하여 운영하는 차세대 네트워킹 아키텍처다. 제어 평면의 컨트롤러는 OpenFlow 프로토콜을 기반으로 동적으로 데이터 평면의 스위치들의 동작을 제어하여, 유연하고 경제적으로 망을 관리할 수 있도록 한다<sup>2)</sup>.

미래 전술네트워크를 구성하는 이질적인 장치들과 체계들간 상호운용성 지원을 위해, SDN을 적용하려는 시도가 이루어지고 있다<sup>3-5)</sup>. 미래 전술네트워크를 단일 또는 소수 컨트롤러 기반 SDN 아키텍처를 적용하는 것은 성능 측면에서 부적합하다<sup>6,7)</sup>. 이를 해결하기 위하여, 대규모 네트워크 제어 및 멀티 도메인간 협업 구조를 지원하는 다계층 컨트롤러 기반 SDN 아키텍처가 제안되고 있다<sup>8-11)</sup>. 또한, 복잡한 네트워크 구조의 효과적 운용을 위한 의사결정 지원을 위하여 지능형 SDN 제어 구조를 채용하고 있다.

전술 SDN 기술의 성능 분석 및 평가를 위해서 모델링 및 시뮬레이션 (M&S: Modeling and Simulation) 기술이 필요하다. 특히, M&S 도구는 다계층 SDN 컨트롤러와의 연동을 지원해야 한다. SDN 기술의 M&S를 위한 대표적인 도구로는 Mininet이 있다<sup>12)</sup>. Mininet은 가상화 기반 네트워크 에뮬레이터로, 오픈소스로 개발되었다. 또한, 모든 버전의 OpenFlow 프로토콜을 지원하여 다양한 SDN 컨트롤러와의 연동이 가능하다. 하지만, 복잡하고 현실적인 SDN 환경에 대한 M&S를 수행하기 어려운 단점이 존재한다<sup>13-15)</sup>. Riverbed Modeler (구 OPNET Modeler)는 Discrete Event Simulation (DES) 기반의 시뮬레이터로<sup>16)</sup>, 보다 복잡하고 현실적인 환경에서 모의실험이 가능하다<sup>17,18)</sup>. Riverbed Modeler의 SDN 모델은 Mininet과 마찬가지로 도구 외부 실제 컨트롤러와의 연동이 필요하다. 따라서, 새로 개발된 SDN 기술을 적용하기 위해서는 모델화된 코드 뿐만 아니라 실제 응용에 대한 수정이 필요하다. 특히, OpenFlow 1.3 버전을 사용하는

OpenDayLight<sup>19)</sup> 컨트롤러와의 연동에 대해 최적화되어, 최신 SDN 기술에 대한 성능 분석 및 평가가 어렵다.

최근, 멀티 도메인 이기종 네트워크를 위한 SDN 시뮬레이터 개발 연구에서는<sup>20)</sup> SDN 컨트롤러의 주요 기능을 모델링하면, 실제 컨트롤러를 연동했을 때와 유사한 실험 결과를 얻을 수 있음을 보였다. 이에 본 논문에서는 Riverbed Modeler와 다계층 SDN 컨트롤러의 연동을 지원하고, 추후 학습 모델을 적용하여 그 효과를 분석 및 평가할 수 있도록 하는 다계층 컨트롤러 및 멀티 도메인 SDN 테스트베드의 구조를 제안하며, 간단한 실험 결과를 보인다. 본 논문에서 제안하는 프레임워크의 특성 및 장점은 다음과 같다.

Riverbed Modeler 기반으로 구축한다. Riverbed Modeler는 유무선 통신에 필요한 구성요소들을 모델링하여 다양한 형태로 제공하며, 보다 복잡하고 현실적인 시나리오를 적용할 수 있다.

모의 SDN 컨트롤러와 연동한다. 다양한 버전의 OpenFlow 프로토콜 및 SDN 기술에 대해 모델링을 수행하여, 구현 시간을 단축할 수 있다.

외부 SDN 컨트롤러와 연동한다. 네트워크 상황 분석 및 제어 수행 시간이 컴퓨팅 파워에 좌우되어, 고성능 컴퓨터에 컨트롤러를 설치하여 모의실험 시간을 단축시킬 수 있다. 또한, 관리자의 개입을 모의실험에 반영할 수 있다.

본 논문의 구성은 다음과 같다. 2장에서 본 논문의 제안 모델을 적용하기 위한 모의실험 도구인 Riverbed Modeler 및 Riverbed사에서 제공하는 SDN 모델에 대해 설명하고, 3장에서 Riverbed Modeler 기반 다계층 지능형 SDN 컨트롤러 연동 모델을 제안한다. 4장에서 제안 모델의 구현 및 실험 결과를 보이고, 5장에서 결론을 맺는다.

## II. 배경 및 관련연구

### 2.1 Riverbed Modeler

Riverbed Modeler는<sup>16)</sup> OPNET사가 Riverbed사에서 인수되기 이전부터 신뢰성을 인정받아, 미 국방정보체계국에서 사용하는 M&S 도구인 Joint Communication Simulation System (JCSS)의 기반이 되었다<sup>21)</sup>. 우리 군은 Riverbed Modeler 기반의 Network Simulstor and Planer for Interoperability (NetSPIN)을 사용하고<sup>22)</sup>, 국내외 다수의 기관에서도 Riverbed Modeler를 네트워크 기술 연구개발에 사용한다.

Riverbed Modeler는 네트워크의 성능 분석 및 평가를 위한 M&S 도구로, 노드, 링크 등 유무선 통신에

필요한 네트워크 구성요소들을 모델링하여 모의실험을 위한 기반으로 사용한다. Riverbed사는 주요 프로토콜들을 직접 모델링하여, 결과를 새로운 버전의 도구에 포함시켜 배포한다. 일부 상용장비의 동작은 공개하기 어렵기 때문에, 외부 라이브러리의 형태로 제공한다.

또한, Riverbed Modeler는 DES 기반의 시뮬레이터이다. 도구의 자체 커널이 이벤트를 이벤트 발생 시간(시뮬레이션 시간)의 순서대로 정렬하여 관리한다. 다음 이벤트가 발생할 차례가 된 경우, 시뮬레이션 시간이 사전 정의된 지점으로 변경되며, 해당 이벤트 발생 시에 동작하기로 예약된 작업을 수행한다.

## 2.2 Riverbed Modeler의 SDN 모델

Riverbed Modeler는 18.6 버전부터 OpenFlow<sup>[1]</sup> 1.3 버전 기반의 SDN 모델을 제공한다. SDN 모델을 사용하기 위해서는 외부 호스트와의 연동을 지원하는 System-In-The-Loop (SITL) 모델이 필요하다. SITL 모델을 기반으로 외부 호스트에 설치된 SDN 컨트롤러가 Riverbed Modeler와 연동되기 때문이다.

SDN 기술의 M&S를 위한 가상화 기반 오픈소스 애플레이터인 Mininet은 모든 버전의 OpenFlow 프로토콜을 다양한 OpenFlow 기반 오픈소스와의 연동을 지원하여, 많은 연구에서 사용되었다. 하지만, 단순한 SDN 환경에 대한 M&S를 수행하기에 적합한 것으로 평가 받는다<sup>[13-15]</sup>. 반면, Riverbed Modeler는 다양한 모델 및 기능 기반의 모의실험 제공하여, 보다 복잡하고 현실적인 환경을 위한 M&S가 가능하다<sup>[17,18]</sup>. 따라서, 전송 SDN 테스트베드를 구축하고 대규모 전투 요소들의 연합 및 합동작전을 고려하는 미래 전송네트워크 시나리오를 반영하기 위해서는 Riverbed Modeler를 이용하는 것이 더 적합하다.

## 2.3 Riverbed Modeler의 SDN 모델의 한계

Riverbed Modeler의 SDN 모델은 Mininet과 마찬가지로 도구 외부 실제 컨트롤러와의 연동이 필요하다. 하지만, SDN 모델이 OpenFlow 프로토콜 1.3 버전만을 지원하며, 특히 OpenDayLight<sup>[19]</sup> 컨트롤러 0.6.4-Carbon에 최적화되어 OpenDayLight 이외의 다른 컨트롤러를 사용할 경우, 모의실험의 안정성이 떨어진다. 따라서, 최신버전의 OpenFlow 기반 컨트롤러와의 연동을 위해서는 모델을 수정해야 한다. 게다가, SDN 기술을 개발하면서 OpenFlow 프로토콜을 개선하거나 다시 정의해야 할 경우, 외부 컨트롤러에 대한 수정 또한 불가피하다. 이는 실제 응용을 수정하는 것

이기 때문에, 연구자에게 프로토콜 모델링을 수행하는 것보다 더 큰 부담으로 다가올 수 있다. Riverbed Modeler의 SDN 모델은 상기 문제로 인해, 최신 SDN 기술에 대한 성능 분석 및 평가가 어렵다.

## 2.4 Riverbed Modeler 기반 SDN 시뮬레이터 및 테스트베드

최근부터 Riverbed Modeler는 SDN 모델을 지원하여 복잡한 시나리오를 대상으로 개발되는 SDN 기술의 모의실험을 가능하게 한다. 하지만, 2.3에서 언급한 문제들로 인해 SDN 기술에 대한 확장성이 떨어진다.

SDN 지원 Wireless Sensor and Actuator Networks (WSAN)을 위한 에너지 인지 라우팅 및 컨트롤러와 단말간 인터페이스 프로토콜<sup>[23,24]</sup> 연구, SDN 지원 Wireless Body Area Network (WBAN)을 위한 인터페이스 프로토콜 연구<sup>[25]</sup>, SDN 지원 Cognitive Radio (CR) 네트워크에 대한 아키텍처 연구<sup>[26]</sup>, 그리고 SDN 지원 5G 스몰셀에 대한 핸드오버 기술 연구<sup>[27]</sup> 제안 방법의 성능평가를 위해, 컨트롤러를 포함한 네트워크 구성을 위한 모든 노드들을 모델링하여 시뮬레이터 내부에 구현했다. Cloud Mobile Media에서의 비디오 스트리밍을 위한 QoE 메트릭 기반 적응형 SDN 아키텍처 연구에서는<sup>[28]</sup> 컨트롤러를 포함한 Service Provider (SP) 클라우드 관련 모델들을 시뮬레이터 내부에 구현했다. 이때, 컨트롤러 모델은 성능지표에 대한 예측을 수행하고, 그 결과에 따라 시뮬레이터 외부에 존재하는 미디어 클라우드를 제어한다. 다중 도메인 SDN 테스트베드를 구축한 연구도<sup>[15]</sup> 있는데, 두 개의 PC에 Riverbed Modeler를 설치하여 각각 도메인을 구성하고, 두 도메인 사이를 라즈베리파이 기반의 게이트웨이로 연결했다. 이 연구에서 두 도메인 내부 SDN 스위치의 동작은 각 도메인과 연결된 외부의 OpenFlow 1.3 지원 OpenDayLight 컨트롤러에 의해 제어된다.

상기 연구들의 Riverbed Modeler 기반 SDN 시뮬레이터 및 테스트베드들은 Riverbed사가 SDN 모델을 지원하기 시작한 이후에 발표되었다. 하지만, Riverbed Modeler가 지원하는 SDN 모델을 사용하는 경우, 신규 SDN 기술을 적용하지 않았다. 반면, 신규 SDN 기술의 적용을 필요로 하는 경우에는 컨트롤러의 기능을 시뮬레이터 내부에 모델링했다. 컨트롤러의 기능을 모델링하면 모의실험에 소요되는 시간을 단축할 수 있지만, 관리자의 개입이 필요한 관제시스템을 적용하기 어렵다.

멀티 도메인 이기종 네트워크를 위한 SDN 시뮬레이터 개발 연구에서는<sup>[29]</sup> SDN 컨트롤러의 주요 기능

을 모델링하여 모의실험을 수행했고, 실제 컨트롤러를 연동했을 때의 실험 결과와 유사한 결과를 얻을 수 있음을 보였다. 따라서, 본 논문에서는 Riverbed Modeler를 기반으로 도구 외부 다계층 SDN 컨트롤러와 연동하고 추후 학습 모델을 적용하기 위한 시뮬레이터 구조를 제안한다. 본 논문의 제안 구조를 적용할 경우, 보다 손쉽게 다양한 버전의 OpenFlow 및 새로운 SDN 기술을 적용할 수 있다. 또한, 테스트베드에 학습 기반 지능형 모의 관제시스템을 연동해서 네트워크 상황에 따른 관리자의 개입까지 모의실험에 반영할 수 있다.

### III. Riverbed Modeler 기반 다계층 지능형 컨트롤러 연동 모델

#### 3.1 Riverbed Modeler 기반 미래 전술네트워크 테스트베드 구조

Riverbed Modeler에서 모의실험을 실행하면, 구현 또는 설정된 노드, 프로세스, 링크 및 트래픽 모델 등이 시나리오 설정창에서 사용자에게 의해 정의된 내용과 같이 메모리에 복사된다. 메모리에 복사된 노드 및 프로세스 모델들은 Riverbed Modeler의 커널 인터페이스만으로는 서로의 메모리 영역에 접근하기 어렵다. 이를 극복하기 위해, 제안 테스트베드에 Riverbed Modeler를 위한 공유메모리 인터페이스를<sup>29)</sup> 구현 및 적용했다. 본 논문에서 제안하는 Riverbed Modeler의 공유메모리 기반 다계층 모의 컨트롤러 연동 구조는 그림 1과 같다.

SDN 테스트베드에서 데이터 평면 (DP: Data Plane)은 Riverbed Modeler에서 SDN 스위치 기능을 수행하는 노드 모델들로 구성된다. 노드 및 노드 내부의 프로세스 모델들은 공유메모리 인터페이스를 이용하여 공유메모리 영역에서 관리되는 노드별 공간에 대한 포인터를 갖는다. 이 포인터들은 외부 모의 컨트롤러에 전송할 모의실험 정보를 기록하기 위해 사용된다.

Riverbed Modeler는 DP 뿐만 아니라 제어 평면 (CP: Control Plane)을 포함한다. CP를 구성하는 요소 중 하나인 컨트롤러 브로커 모델은 시뮬레이터 외부의 모의 컨트롤러에 모의실험 정보를 전송하고, 컨트롤러로부터 명령을 수신하여 각 노드 및 프로세스 모델에 적용한다. 브로커 모델은 SDN 스위치 기능을 수행하는 노드들과는 달리, 공유메모리에서 관리되는 전역 메모리 공간에 대한 포인터만을 갖는다. 전역 메모리 공간에는 노드별로 할당된 공유메모리 공간에 대한 포인터 리스트가 있다. 브로커 모델은 이 리스트를 통해

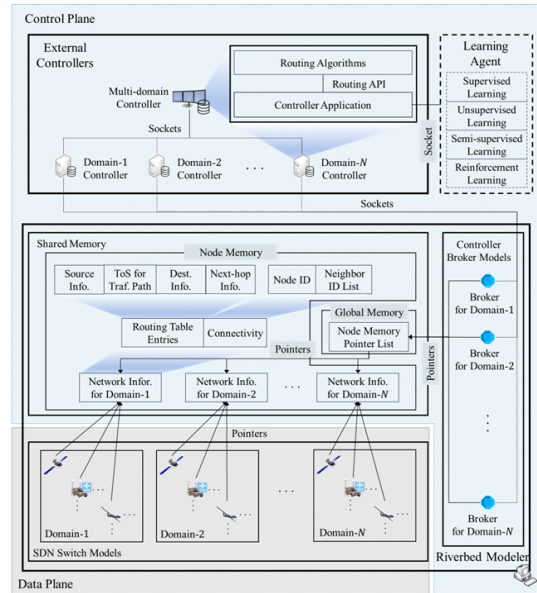


그림 1. 미래 전술네트워크 테스트베드의 공유메모리 기반 다계층 모의 컨트롤러 연동 구조  
Fig. 1. Shared Memory-based External Simulation Controller Interlocking Structure of Future Tactical Network Testbed

노드별로 기록한 정보를 수집 및 전송하며, 컨트롤러의 명령을 모의실험에 반영한다.

모의 컨트롤러들은 다계층 형태로 구성되어 멀티도메인 컨트롤러가 모든 도메인 컨트롤러들과 연결된다. 또한, 각 도메인 컨트롤러는 도메인별 정보 수집 및 전송을 담당하는 브로커 모델과 연결된다. 도메인 컨트롤러들은 브로커 모델에서 수신한 정보를 바탕으로, 자신과 연결된 DP 내부 도메인 노드들에 대한 제어를 수행한다. 또한, 멀티도메인 컨트롤러는 도메인 컨트롤러들로부터 모의실험 정보를 전달 받는다. 만약 하나의 도메인 컨트롤러만으로는 해결할 수 없는 문제가 발생하면, 멀티도메인 컨트롤러가 도메인 컨트롤러들로부터 수신한 정보를 바탕으로 제어를 수행한다.

본 연구에서는 그림 1의 도메인 컨트롤러 연결 구조까지 테스트베드에 적용하였다. 앞서 언급한 바와 같이, Riverbed Modeler는 DES 기반 시뮬레이터이다. 또한, 기존 SDN 모델에서 외부 컨트롤러와의 연동을 위해 SITL을 사용하는 것과는 달리, 제안 구조에서는 소켓을 사용하기 때문에, 모의 컨트롤러가 정보를 처리하고 명령을 생성하여 전송하기까지 모의실험을 일시정지시킬 수 있다. 따라서, 러닝 에이전트와 연계하여, 계산에 오랜 시간이 소요되는 알고리즘들을 수행할 수 있다.

### 3.2 다계층 지능형 컨트롤러와의 연동을 위한 정보 교환 과정

컨트롤러 브로커 모델들은 모의실험 정보를 컨트롤러에 전송하고, 시뮬레이터 외부의 컨트롤러들은 수신 정보를 처리하여 브로커 모델에 전송할 명령을 생성한다. 그림 2는 이러한 브로커 모델, 도메인 컨트롤러 및 멀티 도메인 컨트롤러 사이의 정보 교환 과정을 나타낸다.

이들은 Transmission Control Protocol (TCP)로 연결된다. 도메인 컨트롤러는 브로커 모델 및 멀티 도메인 컨트롤러와 연결하기 위해, 그리고 멀티 도메인 컨트롤러는 여러 도메인 컨트롤러들과 연결하기 위해 스레드마다 하나의 세션을 담당한다. 그림 2에서  $T_R$ ,  $T_{D1}$ ,  $T_{D2}$ ,  $T_{MD1}$ , 그리고  $T_{MD(i+1)}$ 과 같은 식별자가 사용된다. 이는 함수가 테스트베드 어느 요소의 몇 번째 스레드에서 호출되는지를 의미한다.  $T_R$ 은 함수가 Riverbed Modeler에서 호출됨을,  $T_{D1}$ 과  $T_{D2}$ 는 각각 도메인 컨트롤러의 첫 번째 및 두 번째 스레드에서 호출됨을 의미한다. 그리고  $T_{MD1}$ 과  $T_{MD(i+1)}$ 은 각각 멀티 도메인 컨트롤러의 첫 번째 및  $(i+1)$ 번째 스레드에서 호출됨을 의미한다.

도메인 컨트롤러와 연결된 후, Riverbed Modeler의 브로커 모델은 자신이 담당하는 도메인에 대해 초기 정보 (PKT\_INIT)를 구성하여 전송하며, 전송한 정보에 대해 수신을 (ACK\_TYPE\_OK) 확인한다. 도메인

컨트롤러는 브로커 모델로부터 수신한 정보를 멀티 도메인 컨트롤러에 전달하고 수신을 확인한다. 멀티 도메인 컨트롤러는 모든 도메인 컨트롤러들로부터 정보를 수신하기 때문에, 전체 네트워크에 대한 정보를 가질 수 있다.

초기 정보 전송 과정이 종료되면, 브로커 모델은 모의실험 과정에서 변하는 정보 (PKT\_SDN\_INFO)를 공유메모리에서 수집 및 전송한다. 이를 수신한 도메인 컨트롤러는 이전 수신 정보와 비교하여, 토폴로지의 변화를 확인한다 ( $compGraphs(G_i^{last}, G_i^{cur})$ ). 토폴로지의 변화가 없는 경우, 도메인 컨트롤러는 브로커 모델에 명령이 없음 (ACK\_TYPE\_OK)을 알리고, 멀티 도메인 컨트롤러에도 명령이 없다는 정보 (NO\_INSTR)와 함께 브로커 모델로부터 수신한 정보를 전달한다. 토폴로지의 변화가 있는 경우, 도메인 컨트롤러는 자신이 처리할 수 있는 문제인지 여부에 따라 다른 처리 과정을 수행한다. 자신이 처리할 수 있는 경우, 브로커 모델에 기다릴 것을 요청 (ACK\_TYPE\_WAIT)한 후, 명령을 생성 ( $calcNewPath(G_i^{last}, G_i^{cur})$ ) 및 전송한다. 이때, 멀티 도메인 컨트롤러에는 자신이 명령을 생성한다는 정보 (DOM\_INSTR)와 함께 브로커 모델로부터 수신한 정보를 전달한다. 반면, 자신이 처리할 수 없는 문제인 경우, 기다릴 것을 요청하면서 멀티 도메인 컨트롤러에는 제어가 필요함 (MDOM\_INSTR)과 함께 브로커 모델로부터 수신한 정보를 전달한다. 이때, 멀티 도메인 컨트롤러는 제어 메시지를 생성하여 ( $calcNewPath(G_i^{last}, G_i^{cur})$ ) 도메인 컨트롤러에 전송하고, 도메인 컨트롤러는 이를 브로커 모델에 전달한다. 모의실험이 종료되면 브로커 모델은 이를 (SIM\_FIN) 도메인 컨트롤러에 알린 후에 연결을 끊는다. 도메인 컨트롤러 역시 모의실험이 종료됨을 멀티 도메인 컨트롤러에 알린 후에 연결을 끊는다.

### 3.3 SDN 컨트롤러의 수신 정보 기반 명령 생성 알고리즘

3.2에서는 컨트롤러 브로커 모델, 도메인 컨트롤러, 그리고 멀티 도메인 컨트롤러 사이의 정보 교환 과정을 설명했다. 본 절에서는 정보 교환 과정 사이에 도메인 컨트롤러에서 수신 정보를 기반으로 명령을 생성하는 알고리즘에 대해 설명한다.

명령 생성 알고리즘의 설명을 위해, 교환 정보 및 정보 교환을 위한 주요 패킷 형식에 대해 설명한다. 그림 2에서의 PKT\_INIT, PKT\_SDN\_INFO 및 PKT\_INSTR에 대한 구조는 그림 3과 같다.

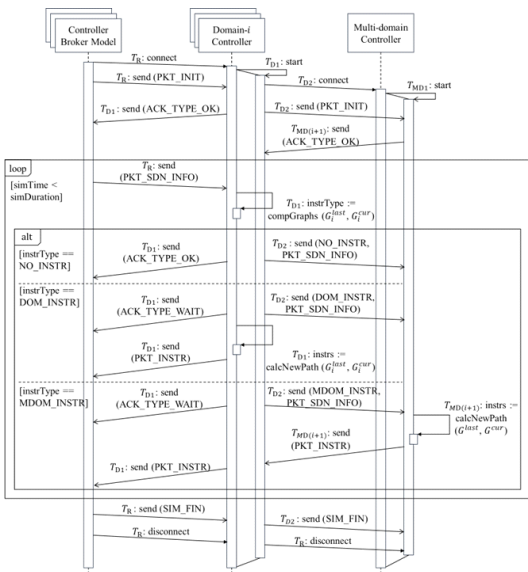


그림 2. 미래 전술네트워크 테스트베드 내부 구성 요소들의 연동을 위한 정보 교환 과정  
Fig. 2. Information Exchange Process for Interlocking Internal Components of Future Tactical Network Testbed

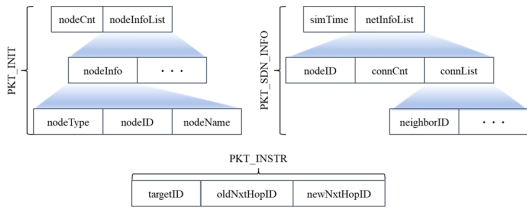


그림 3. 미래 전술네트워크 테스트베드 내부 구성 요소들의 연동을 위한 교환 정보 형식  
 Fig. 3. Format of Exchanged Information for Interlocking Internal Components of Future Tactical Network Testbed

PKT\_INIT은 도메인 구성 노드들의 정보를 알림과 동시에, 이후 전송할 PKT\_SDN\_INFO 및 PKT\_INSTR의 크기를 줄일 수 있는 패키지이다. 도메인 구성 노드의 수와 함께 노드 타입, 노드 ID 및 이름에 대한 리스트를 포함한다. 노드 타입은 해당 노드가 위성, 공중 및 지상 중 어느 차원에 존재하는지 나타낸다. 패키지 수신 후에 컨트롤러들은 노드 ID로 노드를 식별한다. PKT\_SDN\_INFO는 노드들의 연결 정보를 포함한다. 컨트롤러들은 매 주기마다 이를 수신하기 때문에, 이전에 수신한 내용과 비교하여 토폴로지에 변화로 인해 라우팅 경로 변경을 수행해야 하는지 여부를 판단할 수 있다. PKT\_INSTR은 기존 라우팅 테이블 엔트리를 변경하기 위해 컨트롤러에서 브로커 모델로 전송하는 제어 패키지의 형식이다. 두 노드 사이의 연결이 단절되었다고 하면, targetID는 소스 노드의 ID로, oldNxtHopID는 목적 노드의 ID로, 그리고 newNxtHop ID는 목적 노드로 트래픽을 전송하기 위해 경유하는 노드의 ID로 설정한다.

도메인 토폴로지는 그래프  $G=(V,E)$ 로 나타낼 수 있다.  $V$ 는 노드  $u$ 의 집합,  $E$ 는 연결  $e=(u,v)$ 의 집합을 의미한다. 그림 4는 그림 2에 나타난 것과 같이, 도메인 컨트롤러에서 토폴로지 기반 명령 타입 및 명령 리스트를 생성하는 알고리즘을 나타낸다.

도메인 컨트롤러가 가장 최근 수신한 토폴로지 정보 ( $G_i^{cur}$ )와  $G_i^{cur}$  직전 수신한 토폴로지 정보 ( $G_i^{last}$ )를 기반으로 명령을 생성할 수 있는지 여부를 판단하기 위해 compGraphs() 함수를 사용한다 (1번째 줄). 명령 생성이 필요 없는 경우에는 명령 타입을 NO\_INSTR로 지정한다 (2번째 줄). 노드 사이의 연결 관계가 사라진 경우 (4번째 줄), 연결을 가졌던 두 노드가 자신이 제어하는 도메인에 속했다면 명령 타입을 DOM\_INSTR로 지정한다 (5~6번째 줄). 하나라도 다른 도메인에 속했다면 명령 타입을 MDOM\_INSTR로 지정한다 (7~8번째 줄).

그림 4의 calcNewPath() 함수는 두 노드의 연결이

Algorithm 1 Get Instruction Type and New Routing Path

```

1: Procedure compGraphs( $G_i^{last}$ ,  $G_i^{cur}$ )
2:   instrType = NO_INSTR;
3:   while every  $e \in E_i^{last}$  do
4:     if  $e \in E_i^{cur}$  then
5:       if  $u \in V_i^{cur}$  and  $v \in V_i^{cur}$  then
6:         instrType = DOM_INSTR;
7:       else then
8:         instrType = MDOM_INSTR;
9:       end if
10:    end if
11:  end while
12:  return instrType;
13: Procedure calcNewPath( $G_i^{last}$ ,  $G_i^{cur}$ )
14:  while every  $e \in E_i^{last}$  do
15:    if  $e \in E_i^{cur}$  then
16:      targetID =  $u_{ID}$ ;
17:      oldNxtHopID =  $v_{ID}$ ;
18:      while every  $x \in N_1(u)$  do
19:        if  $x \in N_1(v)$  then
20:          newNxtHopID =  $x_{ID}$ ;
21:          break;
22:        end if
23:      end while
24:    end if
25:    createAndInsertInstr(instrs, targetID, oldNxtHopID, newNxtHopID);
26:  end while
27:  return instrs;
    
```

그림 4. 도메인 토폴로지 기반 명령 타입 및 명령 리스트 생성 알고리즘  
 Fig. 4. Domain Topology-based Command Type and Command List Generation Algorithm

단절되었을 때 공통 이웃을 찾아, 새로운 트래픽 경로를 지정하는 명령을 생성한다. 두 노드 사이의 연결 관계가 사라진 경우 (15번째 줄), 연결의 소스 노드  $u$ 를 targetID로, 연결의 목적 노드  $v$ 를 oldNxtHopID로 지정한다 (16~17번째 줄). 다음으로,  $u$ 와  $v$ 의 공통 이웃  $x$ 를 찾아 newNxtHopID로 지정한다 (18~21번째 줄).  $u$ 와  $v$ 의 공통 이웃은 여러 개가 있을 수 있으나, 리스트의 가장 앞쪽에 있는 노드를 선택한다. 생성된 명령을 명령 리스트에 반복하여 삽입 (25번째 줄)하며, 마지막으로 명령 리스트를 반환한다 (27번째 줄). 본 논문에서는 지상 노드가  $H \times W$  격자 형태를 이루고, 각 공중 노드가 4개의 지상 노드 사이에 배치되어 이들과 연결성을 갖는 이차원 미래전술네트워크 구조를 고려한다. 따라서, calcNewPath()와 같이 공통 이웃을 찾으면, 두 노드가 모두 지상노드일 경우 공중 노드, 하나라도 공중 노드일 경우 지상 노드가 공통 이웃이 된다.

도메인 컨트롤러로부터 PKT\_INSTR을 수신한 브로커 모델은 targetID를 갖는 노드의 라우팅 테이블 엔트리에 대해 다음 홉 노드를 newNxtHopID를 갖는 노드로 변경한다. 또한, newNxtHopID를 갖는 노드의 다

음 홉을 oldNxtHopID를 갖는 노드로 변경한다.

#### IV. 구현 및 결과

본 장에서는 III장에서 설계한 구조를 따르는 이차원 단일 도메인 전술네트워크 테스트베드에 대한 구현 결과를 확인한다. Riverbed Modeler 외부의 모의 컨트롤러가 연결 단절을 확인하고, 이를 극복하기 위한 새로운 라우팅 경로를 컨트롤러 브로커 모델에 잘 전달하는지 확인한다.

##### 4.1 구현 환경 및 모의실험 시나리오

제안 구조를 따르는 테스트베드에서 Riverbed Modeler는 18.6 버전이며, 모의 컨트롤러는 Visual Studio 2012로 구현했다. Riverbed Modeler와 모의 컨트롤러는 동일 PC에 구성한다.

모의실험 시나리오는 그림 5와 같이 계획한다. 전술 네트워크에서는 노드들이 지향성 안테나를 이용하여 통신하기 때문에, 지상 및 공중 노드들 사이의 링크를 우선으로 모델링했다. 지상 노드만을 통해 한 개의 트래픽, 지상 및 공중 노드를 통해 한 개의 트래픽이 전송된다. 두 개의 트래픽에 대해 전달 경로 중 붉은색 표기된 부분이 단절된다. 연결 단절 시간은 각각 2000 초 및 1500초 시점이며, 트래픽이 대체 경로를 통해 전달이 되는지 결과를 확인한다.

도메인 컨트롤러의 연결 단절 확인 및 컨트롤러 브로커 모델의 경로 변경 결과

그림 6, 그림 7 그리고 그림 8은 IV.1에서 설명한 시나리오에 대한 모의실험 결과를 나타낸다.

그림 6은 Riverbed Modeler 외부의 모의 컨트롤러에서 네트워크 연결 단절 정보를 확인한 결과를 나타

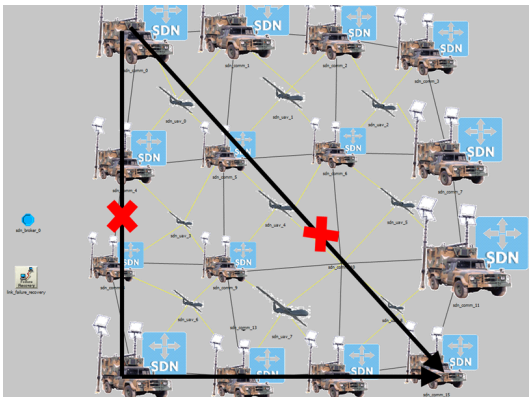


그림 5. 제안 테스트베드에서의 모의실험 시나리오  
Fig. 5. Simulation Scenario in the Proposed Testbed

```
C:\WINDOWS\system32\cmd.exe
Time: 1501.000000) Link sdn_comm_4 <-> sdn_comm_8 has been broken
Time: 1501.000000) Link sdn_comm_8 <-> sdn_comm_4 has been broken
Time: 2001.000000) Link sdn_comm_10 <-> sdn_uav_4 has been broken
Time: 2001.000000) Link sdn_uav_4 <-> sdn_comm_10 has been broken
```

그림 6. 모의 컨트롤러에서 연결 단절 정보 확인 결과  
Fig. 6. Result of Checking Disconnection Information in the Simulated Controller

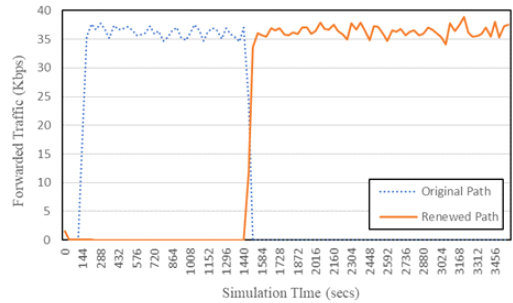


그림 7. 지상 노드를 통해 전달되던 트래픽의 경로 변경 결과  
Fig. 7. Result of Reroute of Traffic Carried through Ground Nodes

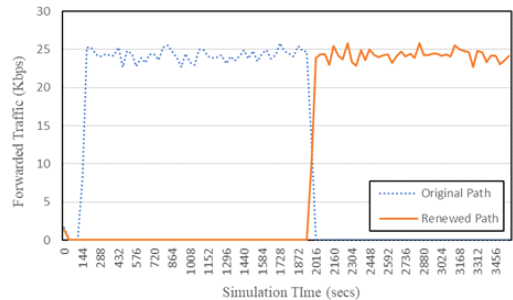


그림 8. 지상 및 공중 노드를 통해 전달되던 트래픽의 경로 변경 결과  
Fig. 8. Result of Reroute of Traffic Carried Through Ground and Aerial Nodes

낸다. 그림 5에서 지상 노드는 좌측 상단의 sdn\_comm\_0부터 우측 하단의 sdn\_comm\_15까지 배치되었다. 공중 노드는 좌측 상단의 sdn\_uav\_0부터 우측 하단의 sdn\_uav\_8까지 배치되었다. 따라서, 컨트롤러 화면을 통해 그림 5의 붉은색 표기된 부분이 단절된 것을 확인할 수 있다.

그림 7은 지상 노드만을 통해 전달되던 트래픽이 1500초에 연결 단절로 인해 중단된 모습을 보인다. 모

의실험에서 컨트롤러 브로커 모델은 모의실험 시간 1 초 간격으로 네트워크 토폴로지를 외부 컨트롤러로 전송하도록 설정했다. 또한, 외부 컨트롤러에서의 명령을 기다리는 동안 모의실험 시간은 흐르지 않기 때문에, 1500초에서 얼마 지나지 않아 새로운 경로로 트래픽이 전송되는 모습을 확인할 수 있다. 그림 8은 지상 및 공중 노드를 통해 전달되던 트래픽이 2000초에 연결 단절로 인해 중단된 모습을 보인다. 또한, 그림 6에서와 유사한 양상으로, 트래픽 전송이 중단되고 얼마 지나지 않아 새로운 경로로 트래픽이 전송되는 모습을 확인할 수 있다.

### V. 결론 및 향후연구

본 논문에서는 지능형 SDN 제어 구조를 채용한 다계층 미래전술네트워크를 위한 M&S 프레임워크를 제안하였다. M&S 도구로는 전술네트워크 성능분석에 효과적인 것으로 검증된 Riverbed Modeler를 사용하였다. 또한, 제안 구조를 적용하고 Riverbed Modeler의 System-In-The-Loop (SITL) 기능을 활용하여 외부에 SDN 컨트롤러와 단말 시스템을 연동한 미래 전술 네트워크 테스트베드를 구현하였다. 실험에서 연결이 단절시 컨트롤러가 이를 인지하고, 트래픽 경로를 효과적으로 변경함으로써, 제안 구조의 실효성을 검증하였다.

본 논문에서 제안하는 테스트베드 구조는 Riverbed Modeler의 공유 메모리 공간에 라우팅 경로 및 노드간 연결관계 정보만을 저장한다. 외부 컨트롤러들은 브로커 모델로부터 수신한 정보를 기반으로 네트워크 상황을 파악하여 명령을 생성 및 전달한다. 트래픽 병목현상에 의한 패킷 손실 등과 같은 추가 전술네트워크 시나리오를 반영하기 위해, 향후에는 Riverbed Modeler의 공유메모리 및 컨트롤러가 수신 및 처리하는 정보에 다양한 성능지표를 추가 반영할 계획이다. 또한, 본 연구에서 구현한 테스트베드를 다차원 멀티 도메인 환경으로 확장하여, 미래 전술네트워크 시나리오에서의 실험 결과를 보일 예정이며, 제안 구조에 학습 모델을 적용하여 복잡한 시나리오를 대상으로 정책 결정 과정의 모의실험을 수행할 수 있도록 할 예정이다.

### References

[1] J. Ghil, G. Lee, S. Lee, B. Roh, J. Kim, D. Kim, and J. Lee, "Operation scheme of aerial relay networks and the analysis of its

effectiveness against failures of terrestrial tactical networks," *J. KICS*, vol. 42, no. 1, pp. 172-180, Jan. 2017.

(<https://doi.org/10.7840/kics.2017.42.1.172>)

[2] N. McKeown, et al., "OpenFlow: Enabling innovation in campus networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 69-74, Seattle, WA, USA, Aug. 2008. (<https://doi.org/10.1145/1355734.1355746>)

[3] G. Lee, E. Choi, B. Cho, B. Roh, D. Ryu, and G. Park, "SDN-based development direction of c2 network system according to network technology development trend," *J. KICS*, vol. 45, no. 4, pp. 730-738, Apr. 2020. (<https://doi.org/10.7840/kics.2020.45.4.730>)

[4] D. Kidston, *Investigations in Software- Defined Networking (SDN)*(2016), Retrieved Apr. 04, 2022, from [https://cradpdf.drdc-rddc.gc.ca/PDF/S/unc220/p803448\\_A1b.pdf](https://cradpdf.drdc-rddc.gc.ca/PDF/S/unc220/p803448_A1b.pdf).

[5] M. Mitchiner and R. Prasad, *Software- defined networking and network programmability: Use cases for defense and intelligence communities* (2014), Retrieved Apr. 04, 2022, from [https://www.cisco.com/c/dam/en\\_us/solutions/industries/docs/gov/software\\_defined\\_networking.pdf](https://www.cisco.com/c/dam/en_us/solutions/industries/docs/gov/software_defined_networking.pdf).

[6] F. P. C. Lin and Z. Tsai, "Hierarchical edge-cloud SDN controller system with optimal adaptive resource allocation for load-balancing," *IEEE Systems J.*, vol. 14, no. 1, pp. 265-276, Mar. 2019. (<https://doi.org/10.1109/JSYST.2019.2894689>)

[7] W. Xia, J. Zhang, T. Q. Quek, S. Jin, and H. Zhu, "Mobile edge cloud-based industrial internet of things: Improving edge intelligence with hierarchical SDN Controllers," *IEEE Veh. Technol. Mag.*, vol. 15, no. 1, pp. 36-45, Jan. 2020. (<https://doi.org/10.1109/MVT.2019.2952674>)

[8] J. Hua, L. Zhao, S. Zhang, Y. Liu, X. Ge, and S. Zhong, "Topology-preserving traffic engineering for hierarchical multi-domain SDN," *Comput. Netw.*, vol. 140, pp. 62-77, Jul. 2018. (<https://doi.org/10.1016/j.comnet.2018.04.011>)

[9] D. A. Chekired and L. Khoukhi, "Distributed



- SDN-based C4ISR communications: A delay-tolerant network for trusted tactical cloudlets,” *ICMCIS 2019*, pp. 1-7, Budva, Montenegro, May 2019.  
(<https://doi.org/10.1109/ICMCIS.2019.8842820>)
- [10] T. Taleb, I. Afolabi, K. Samdanis, and F. Z. Yousaf, “On multi-domain network slicing orchestration architecture and federated resource control,” *IEEE Network*, vol. 33, no. 5, pp. 242-252, Sep.-Oct. 2019.  
(<https://doi.org/10.1109/MNET.2018.1800267>)
- [11] A. M. Hegland, M. Hauge, and A. Holtzer, “Federating tactical edge networks: Ways to improve connectivity, security, and network efficiency in tactical heterogeneous networks,” *IEEE Commun. Mag.*, vol. 58, no. 2, pp. 72-78, Feb. 2020.  
(<https://doi.org/10.1109/MCOM.001.1900508>)
- [12] *Mininet: An Instant Virtual Network on your Laptop (or other PC)*, Retrieved Apr. 04, 2022, from <http://mininet.org/>
- [13] M. Azizi, B. Redouane Benaini, and B. Mouad, “Delay measurement in OpenFlow-enabled MPLS-TP network,” *Modern Appl. Sci.*, vol. 9, no. 3, pp. 90-101, Jan. 2015.  
(<https://doi.org/10.5539/mas.v9n3p90>)
- [14] O. Flauzac, E. M. G. Robledo, and F. Nolot, “Is Mininet the right solution for an SDN testbed?,” *GLOBECOM 2019*, pp. 1-6, Puako, HI, USA, Dec. 2019.  
(<https://doi.org/10.1109/GLOBECOM38437.2019.9013145>)
- [15] S. Lee, Y. Jeong, and B. Roh, “Design and implementation of simulation system for multi-domain tactical SDN Using OPNET modeler,” *J. KICS*, vol. 45, no. 4, pp. 739-747, Jan. 2020.  
(<https://doi.org/10.7840/kics.2020.45.4.739>)
- [16] *Riverbed Modeler: The fastest discrete event-simulator for network modeling*, Retrieved Apr. 06, 2022, from <https://www.riverbed.com/products/npm/triverbed-modeler.html>
- [17] S. Lee, J. Ali, and B. Roh, “Performance comparison of software defined networking simulators for tactical network : Mininet vs. OPNET,” *ICNC 2019*, pp. 197-202, Honolulu, HI, USA, Feb. 2019.  
(<https://doi.org/10.1109/ICCNC.2019.8685572>)
- [18] T. Hu, Z. Guo, P. Yi, T. Baker, and J. Lan, “Multi-controller based software-defined networking: A survey,” *IEEE Access*, vol. 6, pp. 15980-15996, Mar. 2018.  
(<https://doi.org/10.1109/ACCESS.2018.2814738>)
- [19] The Linux Foundation, *OpenDayLight*, Retrieved Apr. 07, 2022, from <https://www.opendaylight.org/>
- [20] G. Lee, Y. Jeong, C. Lee, B. H. Roh, D. Ryu, and G. Park, “SDN simulator for multi-domain heterogeneous network,” *WTS 2020*, Washington, DC, USA, Apr. 2020.
- [21] OPNET Technologies, Inc., *JCSS Technical Reference Manual 10.0 Final (OPNET 2.5.2): Contract HC1047-09-C-4020*, Retrieved Apr. 06, 2022, from [https://www.disa.mil/\\_large\\_files/JCSS\\_10.0\\_Final\\_Tech\\_Reference\\_Manual.pdf](https://www.disa.mil/_large_files/JCSS_10.0_Final_Tech_Reference_Manual.pdf)
- [22] M. Ahn and Y. Kim, “Research on system architecture and simulation environment for cyber warrior training,” *J. KIIISC*, vol. 26, no. 2, pp. 533-540, Apr. 2016.  
(<https://doi.org/10.13089/JKIISC.2016.26.2.533>)
- [23] M. Al-Hubaishi, C. Çeken, and A. Al-Shaikhli “A novel energy aware routing mechanism for SDN enabled WSAN,” *Int. J. Commun. Syst.*, vol. 32, no. 17, Jun. 2018.  
(<https://doi.org/10.1002/dac.3724>)
- [24] A. B. Al-Shaikhli, C. Ceken, and M. Al-Hubaishi, “WSANFlow: An interface protocol between SDN controller and end devices for SDN-oriented WSAN,” *Wirel. Pers. Commun.*, vol. 101, no. 2, pp. 755-773, Apr. 2018.  
(<https://doi.org/10.1007/s11277-018-5714-5>)
- [25] M. Cicioğlu and A. Çalhan, “HUBsFLOW: A novel interface protocol for SDN-enabled WBANs,” *Comput. Netw.*, vol. 160, pp. 105-117, Sep. 2019.  
(<https://doi.org/10.1016/j.comnet.2019.06.007>)
- [26] M. Cicioğlu, S. Cicioğlu, and A. Çalhan,

“SDN-enabled cognitive radio network architecture,” *IET Commun.*, vol. 14, no. 18, pp. 3153-3160, Jun. 2020.

(<https://doi.org/10.1049/iet-com.2019.1346>)

[27] M. Cicioğlu, “Multi-criteria handover management using entropy based SAW method for SDN-based 5G small cells,” *Wirel. Netw.*, vol. 27, no. 4, pp. 2947-2959, Apr. 2021.

(<https://doi.org/10.1007/s11276-021-02625-y>)

[28] M. García-Pineda, S. Felici-Castell, and J. Segura-García, “Adaptive SDN-based architecture using QoE metrics in live video streaming on cloud mobile media,” *SDS 2017*, pp. 100-105. Valencia, Spain, May 2017.

(<https://doi.org/10.1109/SDS.2017.7939148>)

[29] C. Lee, K. Kim, B-h. Roh, B-s. Roh, and J. Choi, “SMAT: Simulator monitoring and analysis tool,” *ICUFN 2015*, pp. 482-485, Sapporo, Japan, Jul. 2015.

(<https://doi.org/10.1109/ICUFN.2015.7182590>)

**이 철 응 (Cheol-woong Lee)**



2015년 2월 : 아주대학교 컴퓨터공학과 졸업

2017년 2월 : 아주대학교 컴퓨터공학과 석사

2017년 3월~현재 : 아주대학교 AI융합네트워크학과 박사과정

<관심분야> 지능형 네트워킹, 미래인터넷, 국방전술 통신, 네트워크 모델링 & 시뮬레이션

[ORCID:0000-0001-9287-7034]

**이 규 민 (Gyu-min Lee)**



2014년 2월 : 아주대학교 컴퓨터공학과 졸업

2016년 2월 : 아주대학교 컴퓨터공학과 석사

2016년 3월~현재 : 아주대학교 컴퓨터공학과 박사과정

<관심분야> 국방전술통신, 네트워크 모델링 & 시뮬레이션, SDN, CR, IoT, AI최적화

[ORCID:0000-0002-6384-795X]

**이 호 진 (Hojin Lee)**



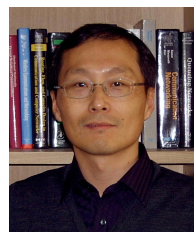
2021년 2월 : 아주대학교 사이버보안학과 졸업

2021년 3월~현재 : 아주대학교 AI융합네트워크학과 석사과정

<관심분야> AI융합 네트워킹, 국방전술 네트워킹

[ORCID:0000-0001-9337-1940]

**노 병 희 (Byeong-hee Roh)**



1998년 2월 : KAIST 전자공학과 박사

1989년 3월~1994년 2월 : KT 연구원

1998년 2월~2000년 2월 : 삼성 전자 연구원

2014년 3월~2015년 2월 : ADD 객원연구원

2000년 3월~현재 : 아주대학교 교수

<관심분야> IoT 플랫폼, AI융합 미래 인터넷, 국방통신네트워크, 가상현실/증강현실

[ORCID:0000-0003-2509-4210]