

대규모 함정 네트워크 검증을 위한 성능시험 도구 설계 및 구현

김재우*, 이민선*, 이재민°, 김동성*, 이재호**, 유재학**, 고영근***, 최낙중***

Design and Implementation of a Performance Test Tool for Large Scaled Naval Ship Networks

Jae-Woo Kim*, Min-Seon Lee*, Jae-Min Lee°, Dong-Seong Kim*, Jae-Ho Lee**, Jae-Hak Yu**,
Young-Keun Go***, Nak-Jung Choi***

요약

본 연구는 함정 시스템에서 네트워크 성능을 평가하기 위한 네트워크 성능시험 도구의 설계 및 구현기법을 제안하였다. 제안하는 구현기법은 함정 내의 각 체계를 구성하는 대규모 네트워크 장비를 비롯한 노드들 사이의 네트워크 성능을 측정한다. 함정 네트워크는 다수의 네트워크 장치와 임무를 수행하는 각종 노드로 구성되며 각 노드 사이에 전술 및 전신 정보를 주고받는다. 네트워크 통신 문제는 전신 상황에서 중요한 역할을 하므로 사전에 다양한 시험을 통해 네트워크의 안전성을 확보하여야 한다. 본 연구는 함정 네트워크 성능 검증을 위해 함정 네트워크 특징으로부터 요구사항을 정의하였고, 요구사항을 만족하는 시험 도구의 설계 및 구현 기술을 제안하였다. 본 연구에서는 제안하는 네트워크 시험 도구를 실제 함정 네트워크 프로토타입 환경에서 시험을 진행하였고, 시험 도구의 기능을 검증하고 시험 결과를 확인하였다. 제안하는 시험 도구를 이용하여 함정 네트워크의 상태를 사전에 시험 할 수 있고 네트워크 구조의 안정성을 확보하고 가용성을 예측할 수 있다.

Key Words : Naval Ship Network, Network Performance Test, Traffic Test

ABSTRACT

This study proposes a design and implementation technique for a network performance testing tool to evaluate network performance in naval ship systems. The proposed implementation technique measures the network performance among nodes, including network equipment that constitutes each system within the naval ship. The naval ship network is composed of numerous network devices and various nodes that perform missions, exchanging tactical and operational information between nodes. Network communication issues play a crucial role in operational situations, so it is necessary to ensure the network's reliability through various tests in advance. This study defined requirements from the characteristics of the naval ship network for the

※ 본 연구는 국방과학연구소의 지원으로 수행되었습니다.

♦ First Author : ICT Convergence Research Center, Kumoh National Institute of Technology, jaewookim@kumoh.ac.kr, 정회원

° Corresponding Author : Department of IT Convergence Engineering, Kumoh National Institute of Technology, ljmpaul@kumoh.ac.kr, 종신회원

* Department of IT Convergence Engineering, Kumoh National Institute of Technology, 20236117@kumoh.ac.kr; dskim@kumoh.ac.kr, 종신회원

** Naval System/Naval R&D Center, Hanwha Systems, {jh8028.lee,jahak90}@hanwha.com

*** Maritime Technology Research Institute, Agency for Defense Development, {ykgo, njchoi}@add.re.kr

논문번호 : 202406-116-C-RN, Received June 14, 2024; Revised August 13, 2024; Accepted August 27, 2024

verification of network performance and proposed a design and implementation technology for a testing tool that satisfies these requirements. In this study, the proposed network testing tool was tested by simulating an actual naval ship network environment, and the results were confirmed. Using the proposed testing tool, it is possible to conduct a preliminary test of the naval ship network status, secure the stability of the network structure, and predict its availability.

I. 서 론

해군 함정 내에는 전투체계, 통신체계, 복합 센서 체계 및 기관 체계와 같은 다양하고 복잡한 이종의 체계로 구성되어 있고, 통제하는 측면에서 전투 관리 체계, C4I(Command, Control, Communications, and Intelligence) 체계, 통합기관체계, 행정체제로 분류할 수 있다¹⁾. 다수의 복합 체계로 구성된 기존의 함정 체계는 대규모의 복잡한 네트워크로 구성되며 각 체계간에는 보안레벨에 따라 서로 분리된 네트워크로 구성된다. 그러나 각 체계를 유기적으로 통합하여 운용할 때 전투성능을 극대화할 수 있는 이유로 점차 체계 간 데이터들이 통합되고 있다²⁾. 2000년대 이후 우리 해군은 함정 통합네트워크 구성 방안 등에 대해서 지속적인 검토를 추진하였고 KDX-III Batch-II, FFX 사업 추진사업에 네트워크 측면에서 통합이 검토되었다. 한편 국방부는 2019년 국방안보정책 세미나의 SMART Navy 추진계획에 따라 함정 체계통합을 추진하였고 네트워크 또한 통합되어 가는 형태로 재구성되고 있으며 함재기 탑재함과 같은 차세대 함정에 적용되어 운용될 예정이

다.^{2,3)} 그림 1은 각종 체계를 포함하는 함정 통합 네트워크 구조이다. 함정 통합 네트워크의 구조는 백본 네트워크를 비롯한 로컬 네트워크단, 운영자 콘솔단, 각 체계에 연결된 장치를 연동하는 연동단, 센서와 무기들로 구성된 단말단으로 구성된다. 제시된 구조는 통제체계를 완전 통합하기 위해 백본 네트워크를 기반으로 모든 통제체계를 통합한다. 따라서 운용자는 운영자 콘솔(HCI control)을 통해 백본 네트워크에 연결된 모든 통제체계에 접근할 수 있고 일부 콘솔이 손상되더라도 백본 네트워크의 다른 콘솔을 통해 모든 통제체계에 접근할 수 있다.

한편 차세대 함정 네트워크를 효율적으로 설계하기 위한 연구와 가상화 및 미들웨어를 통해 컴퓨팅을 최적화하고 성능을 검증하는 연구가 진행되고 있다⁴⁻⁶⁾. 함정 네트워크의 설계를 검증하기 위해서는 함정 네트워크체계 전체가 아닌 각 체계의 일부를 프로토타입으로 구성하여 시험을 진행하여 향후 전체 체계 네트워크 설계의 타당성을 검증한다. 이때 프로토타입은 각 체계를 대상으로 하고 실제 함정에 탑재될 스위치, 라우터, 방화벽과 같은 네트워크 장비를 포함한다. 함정 전투체계

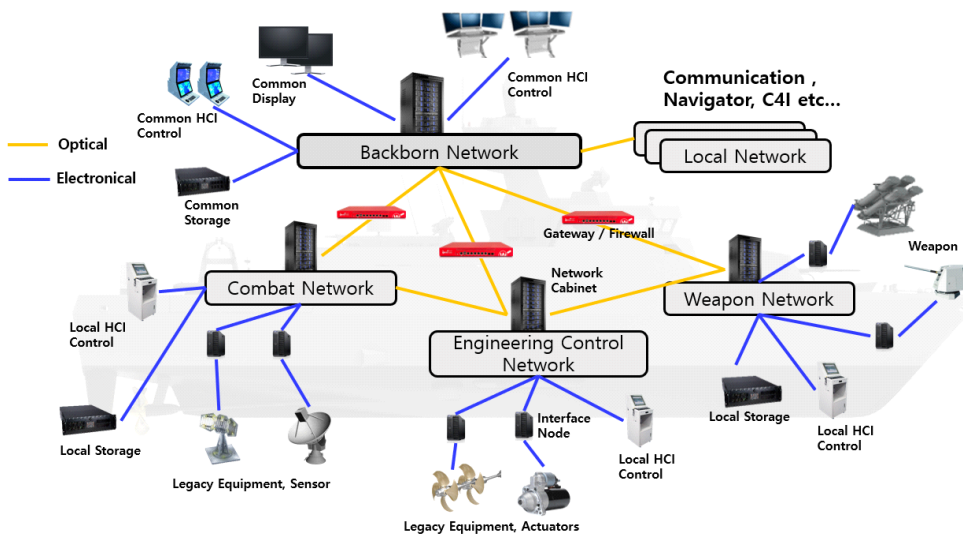


그림 1. 함정 네트워크 구조 [1]
Fig. 1. Naval Ship Network Architecture [1]

에는 지휘 무장 통제 캐비닛, 운용자의 명령을 입력받고 전술 상황을 전지하는 다기능 콘솔, 전술 상황을 실시간으로 출력하는 대형 전시기, 각종 전술 서버, 센서 및 무장을 연동하는 연동 서버들로 구성된 운영 노드들로 구성되며 실제 합정에서 운용할 각각의 소프트웨어(SW)를 탑재하여 시험을 진행한다. 향후 실제 합정에 확대 적용되기전 소프트웨어 기능 검증을 위해 합정이 아닌 육상에 구성하여 시험을 진행한다. 이때 네트워크 시험도구 또한 각 운영 노드에 탑재하여 시험을 진행한다. 각 운영 노드는 크게 윈도우 계열의 운영체제(OS)와 리눅스 계열의 OS로 구분되며 리눅스의 경우 서버 컴퓨터와 임베디드 노드처럼 특정 임무를 빠르게 수행하는 노드에서 사용된다. 따라서 네트워크 시험 도구는 다양한 운영체제에서 동작하여야 한다. 네트워크 시험 도구는 네트워크 환경에서 발생하는 대역폭 부족, 패킷 손실, 지연시간 등 네트워크의 성능을 미리 식별할 수 있다. 특히 안정성이 강조되는 국방 네트워크에서는 사전에 장애를 식별하고 해결하여 네트워크의 신뢰성을 확보하는 것이 매우 중요하다. 네트워크 시험 도구는 국방연구뿐만 아니라 IT 산업계에서도 폭넓게 사용되었다. 네트워크 구조를 설계하고 실제로 서비스 운용을 하기 전 네트워크 환경에서 발생하는 다양한 문제들을 식별하거나 네트워크의 최대 성능을 확인하기 위해서 네트워크 시험 도구를 이용하여 사전 검증을 진행한다. 현재 국방연구기관에서는 차세대 대규모 합정 네트워크 설계 및 검증을 위해 사용중인 도구는 미비한 상태이며 네트워크의 패킷을 캡처하고 분석하는 기능을 제공하는 Wireshark 같은 경우 국방 네트워크 분석을 위해 일부에서 제한적으로 사용되고 있다. 따라서 합정 네트워크 설계 및 검증을 위해 특화 및 적합한 네트워크 시험 도구가 필요한 상태이다.

본 논문에서는 국방 네트워크 중에서 합정 네트워크의 사전 성능시험에 적합한 네트워크 시험 도구를 설계하고 구현하며 이를 검증하는 것을 목표로 하고 있다. 합정 네트워크의 성능시험은 동시에 여러 종류의 트래픽을 발생하여 시험할 수 있어야 하며, 복잡한 네트워크 노드들을 직접 제어하는 것이 아닌 원격으로 중앙에서 트래픽을 시험하는 기능이 필요하다. 또한 합정 내의 네트워크 노드들의 운영체제는 윈도우와 리눅스를 각 운영체제에서 동작이 가능해야 하는 등 몇 가지 요구사항이 있다. 본 논문에서는 합정 네트워크의 성능시험을 위한 네트워크 트래픽 시험 도구를 설계하고 구현 방법을 제안한다. 이때 합정 네트워크 시험에 요구되는 요구사항을 만족하는 시험 도구를 설계하였다. 또한 제안하는 설계 및 구현 방법에 따라 시험 도구를 구현하고

기능을 검증한다. 검증은 시험 도구가 설계한 대로 올바르게 구현되었는지 확인하고, 합정 네트워크 프로토타입 환경에서 합정 네트워크 트래픽 특성을 반영한 시나리오를 시험하여 요구사항을 만족하는지 확인한다.

본 논문 II장에서는 기존의 네트워크 성능시험 도구들의 사례와 특징을 분석하였고 추가로 합정 네트워크의 특성을 고려하여 시험 도구 요구사항을 정의하였다. III장에서는 도출된 요구사항을 기반으로 네트워크 성능시험 도구 설계기법 및 구현 기술을 제안하였다. IV장에서는 제안하는 설계 기법에 따라 구현된 결과를 기술하였고 실제 테스트 베드에서 네트워크 성능 도구를 이용하여 시험을 수행하여 시험 도구를 검증하였다. V장에서는 결론 및 향후 연구에 대해 논의하였다.

II. 배경연구

본 장에서는 기존 네트워크 성능시험 도구를 분석하였고, 합정 네트워크 성능측정을 위한 요구사항을 도출하였다.

2.1 네트워크 성능시험 도구

네트워크 시험 도구는 네트워크 관리자, 시스템 엔지니어 및 보안 전문가 등 네트워크를 시험하고 관리를 위해 매우 중요한 도구이다. 네트워크 시험도구의 필요성은 다음과 같이 정리할 수 있다.

- 네트워크 성능 최적화 : 네트워크 시험 도구는 네트워크의 성능을 평가하고 최적화하는 데 필요하다.
- 보안 강화 : 시험 도구를 사용하여 네트워크의 취약점을 식별하고 보안 문제를 해결할 수 있다.
- 장애 복구 : 네트워크 시험 도구는 네트워크에서 발생하는 문제를 빠르게 진단하여 장애 복구 시간을 단축한다.
- 트래픽 모니터링 : 네트워크 시험 도구는 실시간으로 네트워크 트래픽을 모니터링하고 이상 행위나 과부하를 식별한다.
- 효율적인 자원 할당 : 네트워크 시험 도구를 사용하여 자원 사용률을 모니터링하고 최적화한다.

일반적으로 네트워크의 상태를 확인하기 위한 도구로는 ICMP를 사용하여 호스트 간에 패킷을 보내고 응답시간을 측정하기 위해 프로세스 형태의 Ping과 상태 진단과 경로추적 그리고 네트워크상의 문제점 진단을 위한 Traceroute, Nslookup 등 별도의 설치 없이 OS에서 제공하는 도구들이 있다. 추가로 다양한 측정 인자를 확인하기 위해 iPerf, Wireshark, Netflow 등 다양한 도

구들이 개발되었다. 본 절에서는 기존 대표적인 네트워크 성능시험 도구들의 장단점을 기술한다.

표 1은 대표적인 4가지 시험도구에 대해 주요특징과 장단점을 요약한 것이다. iPerf는 네트워크 성능 측정을 하기 위한 강력한 도구로, 특히 TCP와 UDP 스트림의 대역폭을 시험하는 데 널리 사용된다⁷⁾. iPerf는 콘솔 기반으로 동작하기 때문에 사용자가 명령어를 직접 입력해야 한다. 따라서 명령어를 조합하여 스크립트 형태로 작성하여 시험을 진행할 수도 있다. 또한 송신 측과 수신 측에 각각 프로그램을 실행시켜 양방향 시험을 동시에 수행할 수 있다. Wireshark는 네트워크 트래픽 분석을 위한 가장 널리 사용되는 오픈 소스 도구 중 하나이다. Wireshark는 네트워크 트래픽을 캡처하고 분석하여 문제를 진단하고 네트워크 성능을 최적화하는데 주로 사용된다⁸⁾. 주로 시험보다는 분석하는 측면에 특화된 기능을 GUI형태로 제공한다. NetFlow는 Cisco Systems에서 개발한 네트워크 프로토콜과 트래픽을 모니터링하고 분석하는데 사용된다⁹⁾. NetFlow는 네트워크 단말이 아니라 네트워크 장비에서 발생하는 트래픽 흐름을 기록하고 이 정보를 바탕으로 어떤 애플리케이션이나 장치가 가장 많은 대역폭을 사용하는지 파악할 수 있다. Nagios는 네트워크 모니터링 도구로, 시스템 및 네트워크 인프라의 상태를 모니터링하고, 문제를 감지하여 대응할 수 있게 한다¹⁰⁾. Nagios는 서버, 네트워크 장비, 애플리케이션, 서비스 등 IT 인프라 전반을 모니터링

크 장비, 애플리케이션, 서비스 등을 포함한 IT 인프라 전반을 모니터링하며, 다양한 플러그인과 확장 기능을 통해 높은 유연성과 확장성을 제공한다. 그 외에도 상용 네트워크 성능 분석 도구로는 SolarWinds가 제공하는 SolarWinds Network Performance Monitor (NPM)은 ManageEngine에서 개발한 네트워크 모니터링 및 관리 도구인 OpManager가 있다¹¹⁾. 그러나 이러한 도구들을 이용하여 합정 네트워크를 시험하기에는 몇 가지 제한 사항이 존재한다. 국방 네트워크는 높은 보안 수준을 유지해야 하기에 민간 상용도구 사용이 제한될 수 있다. 예를 들어 제한된 권한 내에서만 도구를 설치하고 사용할 수 있다. 또한 합정 내 하드웨어 자원이 제한적일 수 있다. 특히 네트워크 시험 도구가 많은 하드웨어 자원을 소비한다면 합정에 운용되는 다른 시스템에 영향을 줄 수 있다. 또한 합정에서는 특수한 구성과 프로토콜을 사용할 수 있어 상용도구의 경우 호환이 되지 않을 수 있다.

2.2 합정 네트워크 특성 및 시험 요구사항

본 절에서는 합정 네트워크 시험을 위해 합정 네트워크의 특성과 성능시험을 위한 요구사항을 정리하였다. 그림1의 합정 네트워크의 구조처럼 합정 네트워크는 무기체계가 더욱 대형화되고 복잡해지며 각 체계의 상호연동 및 통합으로 다양한 시험과 요구 조건들이 정의되고 있다. 합정 네트워크 시험 도구의 주된 기능은 합정

표 1. 기존 네트워크 시험 도구 특징 및 장단점 요약
Table 1. Existing Network Test Tool Features and Pros and Cons Summary

이름	개발주체	특징	장점	단점
iPerf/iPerf3	Energy Sciences Network, ESnet	- 네트워크 성능 측정, TCP, UDP, 패킷 손실률, 지터, 대역폭 측정 - 멀티캐스트 트래픽 지원	- 유연한 테스트 옵션 - 사용 편의성 - 양방향 동시 테스트 지원 - 다양한 프로토콜지원 - 오픈 소스	- 콘솔 기반 - 트래픽 패턴 사용자 직접구현필요
Wireshark	Wireshark Foundation	- 네트워크 트래픽 분석 - 네트워크 패킷 분석	- 다양한 프로토콜 지원 - 편의를 위한 GUI제공 - 멀티미디어 플랫폼 지원 - 오픈 소스	- 패킷 분석을 위한 깊은 이해 필요 - 복잡한 기능 - 높은 시스템 자원 필요
NetFlow	Cisco Systems	- 네트워크 프로토콜 및 트래픽 흐름을 모니터링 - 네트워크 장비에서 트래픽 흐름을 파악	- 대규모 네트워크 트래픽 모니터링 - 표준화된 데이터 포맷을 사용하여 장비와 호환성을 제공 - 오픈소스	- 실시간 모니터링 한계 - 고성능 하드웨어 및 대용량 저장공간 요구
Nagios	Nagios Enterprises	- 시스템 및 네트워크 인프라의 상태를 모니터링 - 서버, 네트워크 장비, 애플리케이션, 서비스 등 IT 인프라 전반을 모니터링	- 다양한 플러그인을 통해 기능을 쉽게 확장 - 다양한 성능지표를 모니터링 가능 - 강력한 알림 및 보고 기능 - 오픈 소스	- 초기설정 어려움 - 유료 버전에서만 다양한 기능 및 GUI사용 - 고성능 하드웨어 요구

네트워크의 각 노드가 정상적으로 동작하는지 확인하기 위해 종단 간(End-to-End)의 처리량(Throughput)을 측정한다. 본 연구에서는 그중 네트워크 설계 결과를 시험하기 위한 시험용 SW로 네트워크 성능시험 도구의 요구사항은 다음과 같이 정의하였다.

- 소프트웨어 구성 : 함정 네트워크 성능시험용 SW는 트래픽을 시험하는 송수신 SW와 송수신 SW를 제어하고 트래픽 송수신 결과를 확인하는 시험분석 SW로 분리 구성되어야 한다.
- 이식성 : 함정 네트워크 성능시험용 SW는 PC 기반 함정 네트워크 시험 노드 장치 하드웨어(HW)에 이식 및 통합이 가능해야 한다. 이때 시험 노드 장치의 OS가 Window 10/11/Server 계열과 Ubuntu/CentOS 리눅스 및 임베디드 리눅스에서 모두 이식 가능해야 한다.
- 트래픽 제어 기능 : 함정 네트워크 성능시험용 SW는 각 노드마다 트래픽을 생성하고 송수신하는 기능을 제공해야 한다. 트래픽의 종류로는 UDP, TCP, Multicast, Unicast와 같은 종류의 트래픽을 발생시킬 수 있어야 하며 함정 네트워크 특성을 반영한 다양한 크기의 트래픽을 모의해야 한다.
- 시험 실시간성 : 함정 네트워크 성능시험용 SW는 함정 전체 노드의 트래픽을 동시에 제어하고 모니터링하는 기능을 제공해야 한다.
- 시험 편의성 : 함정 네트워크는 구성하는 노드의 갯수가 많고 가상화 노드, 임베디드 노드, PC기반 노드 등 다양하게 구성되어 있어 노드를 직접 제어하는 것이 아닌 원격으로 중앙에서 시험하는 기능이 요구된다.

이외에도 성능시험 SW는 함정 네트워크에서 일어날 수 있는 모든 트래픽을 발생시켜 각 체계를 통합하는 망연계장치나 게이트웨이, 방화벽에 대한 기능을 검증할 수 있어야 한다.

III. 함정 네트워크 시험 도구 설계 기법

3.1 함정 네트워크 성능시험 도구 시스템 모델

본 절에서는 함정 네트워크 요구사항을 기반으로 함정 네트워크의 성능시험 도구 설계기법을 제안하였다. 함정 네트워크 성능시험 도구는 트래픽 송수신 SW(TRx SW)와 시험분석 SW(Test & Analysis SW)로 구성된다. 함정 네트워크 성능시험 도구를 이용한 함정 네트워크 시험 절차는 다음과 같이 네트워크 시험을 위한 준비부터 결과 분석까지 4단계로 구분할 수

있다.

- Step 1: 시험분석 SW 및 송수신 SW 실행 및 연결
- Step 2: 네트워크 시험 시나리오 작성
- Step 3: 시험 명령어 전달 및 시험 진행
- Step 4: 시험 결과 수집 및 분석

그림 2는 제안하는 함정 네트워크 성능시험 도구의 운용도이다. 송수신 SW는 함정 네트워크를 구성하는 각 장치에 이식되어 동작하며 터미널 기반 애플리케이션이며 대문형태로 동작도 가능하게 설계하였다. 송수신 SW는 네트워크 트래픽을 송신하는 기능과 트래픽을 수신하여 성능을 측정한다. 따라서 하나의 트래픽 시험은 송신 노드와 대응되는 수신 노드가 필요하다. 이때 송신 노드는 Client로 동작하고 수신 노드는 Server로 동작한다. 시험분석 SW는 함정 네트워크 장치에 이식된 송수신 SW와 통신하며 전체 성능시험 시나리오를 전달하며 성능시험에 필요한 제어메시지를 전달한다. 이처럼 시험분석 SW는 중앙제어 역할을 통해 동시에 여러 노드에 시험 시나리오에 따라 시험 명령어를 전달하고 성능측정 결과를 수집하며 측정 결과를 분석함으로써 시험 편의성 및 원격제어와 실시간성 요구사항을 만족하였다. 이때 송수신 SW와 시험분석 SW는 TCP/IP 소켓연결을 통해 메시지를 주고받는다.

그림 3은 제안하는 함정 네트워크 시험 도구 시스템 모델이다. 제안하는 시험 도구는 네트워크 트래픽 시험을 위해 iperf를 사용한다. iperf 도구는 다른 소프트웨어나 플랫폼 가볍게 플러그인 할 수 있고 트래픽 제어 기능을 제공한다. 따라서 TCP, UDP 프로토콜 시험과 Unicast, Multicast 시험을 지원하며 트래픽 패킷 크기를 자유롭게 가변할 수 있다. iperf를 이용하여 함정 네트워크 시험 도구를 설계함으로써 함정 네트워크 시험 요구사항을 만족하고 개발시간도 단축할 수 있다. 송수

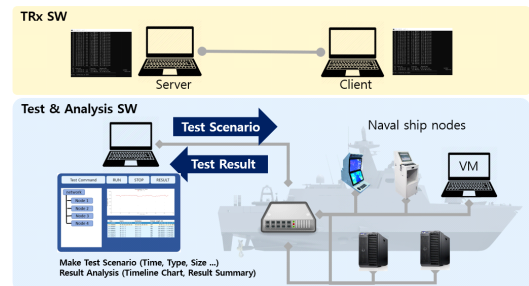


그림 2. 함정 네트워크 성능시험 도구 운용도
Fig. 2. Operation Diagram of Testing Tool for Naval Ship Network

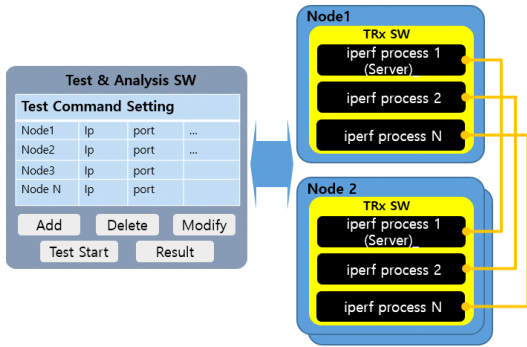


그림 3. 함정 네트워크 성능시험 도구 시스템 모델
Fig. 3. System Model of Testing Tool for Naval Ship Network

신 SW는 또 다른 노드에 이식된 송수신 SW와 대응되어 트래픽 시험을 진행한다. Unicast 시험의 경우 1:1로 Multicast의 경우 1:N으로 대응되어 네트워크 트래픽 시험을 진행한다. 하나의 노드에 하나의 송수신 SW가 동작하며, 하나의 송수신 SW는 여러 개의 iperf 프로세스를 생성하여 트래픽 시험을 진행한다.

시험분석 SW는 네트워크 시험을 위한 전체 시험 시나리오를 작성하고 관리하며 시험 결과를 수집하고 분석한다. 시험분석 SW는 네트워크 시험 시나리오 작성 시 송수신 SW에 전달하여 iperf 프로세스를 동작하기 위해 명령어들을 생성한다. 시험분석 SW는 사용자 편의를 위해 GUI형태로 설계하였다.

본 논문에서 제안하는 송수신 SW와 시험분석 SW는 Qt6 도구를 이용하여 구현한다. Qt6 도구는 크로스 컴파일 프레임워크로써 윈도우와 리눅스, 임베디드 시스템에서 동일한 소스코드로 개발 및 배포가 가능한 장점이 있으며 이식성 요구사항을 만족할 수 있다¹²⁾.

3.2 송수신 SW 설계

본 절에서는 송수신 SW의 설계 방법을 제안하였다. 송수신 SW는 전술하였듯이 함정 네트워크 단말 노드에 이식되어 시험분석 SW에서 트래픽 시험 시나리오를 전달받아 해당 프로세스를 동작한다. 트래픽 시험 시나리오는 iperf 프로세스에서 수행되는 시험 명령어들의 집합으로 구성된다. 그림 4는 송수신 SW의 상태 및 흐름도이다. 기본적으로 송수신 SW의 모든 상태는 송수신 SW의 제어 메시지에 따라 결정된다. 따라서 송수신 SW는 시험분석 SW와 연결이 이루어져야 시험을 수행할 수 있다. 먼저 Idle 상태에서 주기적으로 시험분석 SW에 연결을 시도한다. 시험분석 SW의 노드 주소와 포트 번호 그리고 접속 시도 횟수는 별도의 파일로 저장되어 있고 송수신 SW에서 불러온다. 시험분석 SW

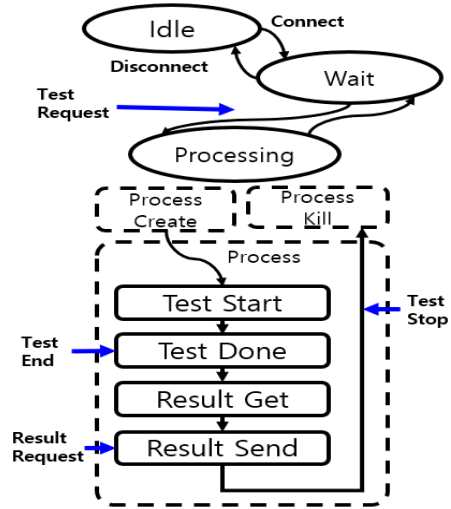


그림 4. 송수신 SW 상태도
Fig. 4. TRx SW State Diagram

와 연결이 이루어지면 시험분석 SW로부터 네트워크 트래픽 시험을 수행할 수 있는 대기 상태인 Wait 상태가 된다. 만일 애기치 않은 네트워크의 문제나 노드의 재시작으로 연결이 끊어지면 송수신 SW는 다시 연결을 시도한다. 시험분석 SW로부터 단일 또는 다수의 시험 명령어가 도착할 수 있다. 시험 명령어가 도착하면 각 시험 명령어마다 각각의 iperf 프로세스를 생성하고 Processing 상태로 천이 되고 시험을 시작한다. 트래픽 시험이 완료되면 송수신 SW에서 각 프로세스의 결과를 수집하고 시험분석 SW의 요청에 따라 결과를 전송한다. 시험 종료 메시지를 받으면 해당 프로세스를 종료한다. 시험 종료 메시지는 정상적으로 시험이 종료되었을 때 또는 사용자가 강제적으로 종료할 때 발생한다. 만일 잘못된 시험 시나리오를 시험 중이거나 진행되고 있는 시험이 연결 및 노드의 문제로 정상적으로 진행되지 않을 때 사용자는 시험을 강제로 종료하고 재시작할 수 있다.

3.3 시험분석 SW 설계

시험분석 SW는 함정 네트워크 시험 전체를 관리하는 핵심 시스템이다. 그림 5는 시험분석 SW의 흐름도이다. 시험분석 SW는 Wait, Scenario Setting, Result Analysis 그리고 Testing 상태를 가지며 데이터를 저장하는 Client List, Result List, Scenario List를 가진다. 시험분석 SW가 시작되면 Wait 상태가 되며 TCP/IP 서버 소켓을 열어 송수신 SW로부터 연결을 기다린다. Wait 상태에서 사용자로부터 입력을 받아 해당 입력을 수행한다. 송수신 SW로부터 연결요청이 오면 Client

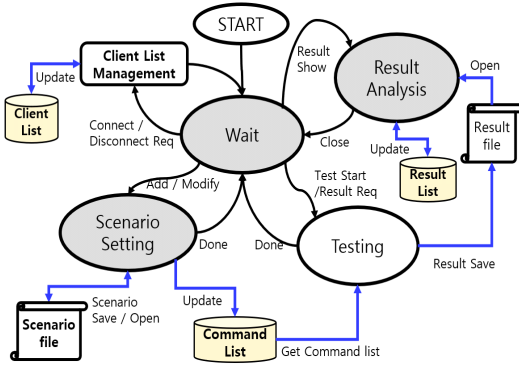


그림 5. 시험분석 SW 상태도
Fig. 5. Test & Analysis SW State Diagram

List Management에서 연결을 허락하고 Client List 저장소에 송수신 SW가 구동되는 Client 목록들을 저장하고 관리한다. 송수신 SW와 연결이 종료되거나 장애로 인해 끊어지면 Client List를 갱신한다. 사용자가 네트워크 시험을 위해 시험 시나리오를 작성하는 요청이 들어오면 Scenario Setting 상태로 천이되고 명령어 Add/Modify/Delete를 통해 전체 시험 시나리오를 구성한다. 이때 Command List 저장소에 현재 생성된 시나리오에 따라 명령어를 갱신한다. 추가로 전체 시나리오 구성을 시나리오 파일에 저장하여 향후 네트워크 시험

에서 재사용할 수 있도록 한다. Test Start 메시지가 들오면 Testing 상태로 천이되며 송수신 SW에 시험 시나리오를 전송하여 시험을 시작한다. 이때 Command List를 참조하여 송수신 SW에 시험 명령어들을 전송한다. 시험이 완료되고 Result Req 요청이 들어오면 Testing 상태에서 각 송수신 SW에 결과를 요청하여 시험 결과를 수집하여 Result File에 저장한다. Result Analysis 상태에서는 네트워크 시험 결과를 종합적으로 분석하여 결과를 보여준다. 시험 결과 항목은 크게 시나리오의 각 명령어에 대한 시험 결과와 시나리오에 대한 종합 그리고 시험 결과에 대한 종합분석 정보로 구성된다. 시험 결과는 시나리오에 따라 하나의 파일로 저장된다. Result Analysis 상태에서 분석하고자 하는 결과 파일을 열어 결과를 분석하여 출력한다.

그림 6은 시험분석 SW와 송수신 SW를 이용한 네트워크를 시험 순서도이다. 먼저 시험분석 SW와 송수신 SW가 연결되면 Wait 상태가 되며 시험 준비가 완료된다. 시험분석 SW에서 사용자가 시험 시나리오를 완료하고 시험을 시작하면 시험 시나리오의 명령어를 순차적으로 송수신 SW로 전송한다. 송수신 SW는 순차적으로 도착한 시험 명령어를 순차적으로 수행한다. 이후 시험분석 SW는 모든 시험 명령어의 갯수와 각 명령어의 트래픽 시험시간을 고려하여 시험완료 대기시간

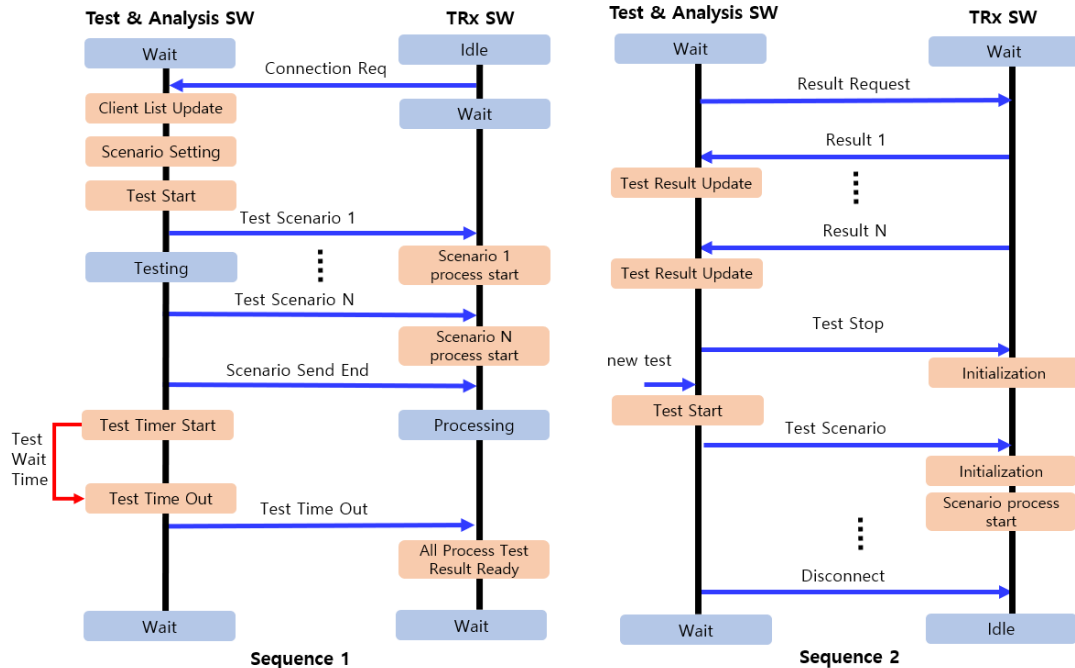


그림 6. 합정 네트워크 시험 도구 시험 순서도
Fig. 6. Naval Ship Network Test Tool Testing Sequence Chart

(Test Wait Time)을 계산하고 시험완료 타이머를 시작한다. 시험완료 대기시간이 만료되면 연결된 모든 송수신 SW에 시험시간 완료(Test Time Out) 메시지를 보낸다. 송수신 SW는 시험시간 완료 메시지를 받으면 시험이 완료된 프로세스의 결과를 저장하여 송신할 준비를 한다. 시험분석 SW가 송수신 SW에 결과를 요청하면 각각 시험 명령어에 따라 결과를 회신한다. 시험분석 SW는 결과를 수집하여 파일로 저장하고 결과를 분석하여 출력한다. 만약 네트워크 시험을 수행하는 도중 강제로 종료될 원하거나 새로운 시험을 진행하고자 하면 강제 종료(Test Stop) 메시지와 새로운 시험 명령어를 전달한다. 새로운 시험 명령어가 전달되면 기존의 진행 중인 모든 프로세스는 종료되고 다시 시험을 시작하게 된다.

3.4 합정 네트워크 트래픽 모델 설계

합정 네트워크를 대상으로 네트워크 트래픽 시험을 진행하기 위해서는 합정 네트워크에서 주로 사용되는 트래픽 종류와 크기를 고려해야 한다. 본 절에서는 합정에서 주로 사용되는 트래픽 크기와 종류를 정의한다. 제안하는 시험 도구는 합정 특화 네트워크 트래픽을 손쉽게 네트워크 시험 시나리오로 작성할 수 있도록 한다.

합정 네트워크에서 주로 사용되는 트래픽의 종류와 그에 따른 정보를 표2에 나타내었다. 합정 데이터는 여러 가지 종류가 있으나 네트워크 트래픽 시험에서 고려해야 할 데이터로는 전시정보, 전술정보, 전시기정보, 상태정보, 연동정보, 비디오정보, 기록정보, 명령어 정보, 원격전시정보 등이 있다. 데이터의 종류에 따라 네트워크 트래픽이 사용하는 프로토콜과 전송방식, 패킷

표 2. 합정 네트워크 트래픽 정보
Table 2. Naval Ship Network Traffic Information

	Data Type	Traffic Type	Protocol	Traffic Size (Mbps)	Paket Size (byte)
1	Display	Unicast/Multicast	UDP	35	1204
2	Tactical	Multicast	UDP	24	1204
3	Display Equipment	Unicast	UDP	143	1204
4	Status	Unicast	UDP	10	1204
5	Integration	Multicast	UDP	40	1204
6	Video	Multicast	UDP	247	1204
7	Log	Unicast	UDP	10	1204
8	Command	Unicast	TCP	9	1204
9	Remote	Multicast	UDP	18	1204

크기가 결정된다. 합정 네트워크에서 사용되는 전시정보나 전술정보 데이터의 세부 항목에 따라 차이가 있으나 표 2에서는 각 항목의 최대 대역폭을 기준으로 정의하였다. 이는 최대 대역폭을 수용한다면 네트워크의 안정성을 검증할 수 있기 때문이다. 제안하는 시험 도구는 이런 정보들을 반영하여 트래픽 시나리오를 손쉽게 작성한다.

IV. 합정 네트워크 시험 도구 구현 및 검증

본 장에서는 제안하는 합정 네트워크 시험 도구의 구현한 결과를 기술한다. 또한 제안하는 합정 네트워크 시험 도구의 검증을 위해 합정 네트워크 프로토타입 환경을 정의하고 프로토타입 환경에서 네트워크 트래픽 시험 진행 결과를 기술한다.

4.1 송수신 SW 구현 결과

그림 7은 송수신 SW의 구현 결과이다. 송수신 SW는 콘솔 형태로 실행되며 시험 과정을 화면에 출력한다. 그림 7에서는 특정 IP 주소로 연결을 시도하고 있으며 4번째 시도에서 연결되어 대기 상태로 천이한 결과이다. 출력 로그는 시험분석 SW로부터 수신된 제어 메시지와 그에 따른 처리상태를 출력하고 네트워크 트래픽 시험이 시작되면 실시간으로 시험 결과를 출력한다.

그림 8은 송수신 SW의 콘솔 명령어 사용법이다. 송수신 SW는 간단한 사용자 명령을 받아서 동작할 수도 있다. 편의를 위해 현재 상태, 재연결 시도, 연결취소, 강제종료 등 몇 가지 간단한 명령어를 구현하였다. 콘솔 명령어는 연결 상태, 시험 상태가 확인되지 않을 경우 진단을 위해 사용할 수 있다.그림 9는 송수신 SW가 시험분석 SW로부터 시험 명령어를 받아 트래픽 시험을 수행중인 화면이다. 두 개의 프로세스가 만들어져 시험이 진행 중인 것을 확인 할 수 있다. Process(3)는

```

Network Deamon Console version v0.3 (Release 240719)
"windows:Windows 10 Version 21H2"
Unique IP count: 3
IP 0 : "202.31.134.231"
IP 1 : "192.168.142.1"
IP 2 : "192.168.80.1"
Please select an IP address by entering the corresponding index:
0
Client IP : "202.31.134.231"
Current State : "Wait"
Server IP "202.31.134.231"
Port : 8888
TryTime: 1000
Attempting to connect to "202.31.134.231" , port number "8888"
Invalid command.
Attempting to connect to "202.31.134.231" , port number "8888" try: 0
Attempting to connect to "202.31.134.231" , port number "8888" try: 1
Attempting to connect to "202.31.134.231" , port number "8888" try: 2
Attempting to connect to "202.31.134.231" , port number "8888" try: 3
Attempting to connect to "202.31.134.231" , port number "8888" try: 4
Attempting to connect to "202.31.134.231" , port number "8888" try: 5
Connected
Current State : "Wait"
    
```

그림 7. 송수신 SW 실행 화면
Fig. 7. TRx SW Start Screen


```

help
-h : help
Ctrl + C : exit program
<IP> <Port> <Option> : connect to <IP> Server using <Port>
                   <Option> : 0 (one time, default), 1 (Continuously trying)
-d : disconnect from Server
-k : kill all iperf process objects in progress
-s : print program state
-i : print client IP address
-a : Abort Connecting
    
```

그림 8. 송수신 SW 콘솔 명령어
Fig. 8. TRx SW Console Commands

```

Process(0): 1 | 1.00-2.00 sec | 1.24 Mbytes | 10.4 Mb/s | 0.166 ms | 0/1274 (0%)
Process(3): 1 | 1.00-2.00 sec | 1.26 Mbytes | 10.6 Mb/s | 0.122 ms | 0/1288 (0%)
Process(0): 1 | 2.00-3.00 sec | 1.25 Mbytes | 10.6 Mb/s | 0.122 ms | 0/1288 (0%)
Process(3): 1 | 2.00-3.00 sec | 1.24 Mbytes | 10.4 Mb/s | 0.170 ms | 0/1275 (0%)
Process(0): 1 | 3.00-4.00 sec | 1.25 Mbytes | 10.5 Mb/s | 0.096 ms | 0/1286 (0%)
Process(3): 1 | 3.00-4.00 sec | 1.26 Mbytes | 10.6 Mb/s | 0.122 ms | 0/1272 (0%)
Process(0): 1 | 4.00-5.00 sec | 1.26 Mbytes | 10.6 Mb/s | 0.122 ms | 0/1290 (0%)
Process(3): 1 | 4.00-5.00 sec | 1.24 Mbytes | 10.4 Mb/s | 0.144 ms | 0/1280 (0%)
Process(0): 1 | 5.00-6.00 sec | 1.24 Mbytes | 10.4 Mb/s | 0.115 ms | 0/1289 (0%)
Process(3): 1 | 5.00-6.00 sec | 1.24 Mbytes | 10.4 Mb/s | 0.124 ms | 0/1289 (0%)
Process(0): 1 | 6.00-7.00 sec | 1.24 Mbytes | 10.4 Mb/s | 0.192 ms | 0/1297 (0%)
Process(3): 1 | 6.00-7.00 sec | 1.25 Mbytes | 10.5 Mb/s | 0.197 ms | 0/1278 (0%)
Process(0): 1 | 7.00-8.00 sec | 1.25 Mbytes | 10.5 Mb/s | 0.115 ms | 0/1289 (0%)
Process(3): 1 | 7.00-8.00 sec | 1.24 Mbytes | 10.4 Mb/s | 0.124 ms | 0/1289 (0%)
Process(0): 1 | 8.00-9.00 sec | 1.24 Mbytes | 10.4 Mb/s | 0.124 ms | 0/1289 (0%)
Process(3): 1 | 8.00-9.00 sec | 1.26 Mbytes | 10.6 Mb/s | 0.124 ms | 0/1289 (0%)
Process(0): 1 | 9.00-10.00 sec | 1.26 Mbytes | 10.6 Mb/s | 0.124 ms | 0/1289 (0%)
Process(3): 1 | 9.00-10.00 sec | 1.25 Mbytes | 10.5 Mb/s | 0.192 ms | 0/1297 (0%)
iperf 3.0.0-10.00 sec | 1.25 Mbytes | 10.4 Mb/s | 0.197 ms | 0/1278 (0%)
Server Report: 1 | Server Report | Interval | Transfer | Bandwidth | Jitter
-----|-----|-----|-----|-----|-----
Total | Data | Conn | M | 0.00-10.00 sec | 12.5 Mbytes | 10.5 Mb/s | 0.197 ms | 0/1278 (0%)
    
```

그림 9. 송수신 SW 시험 진행 화면
Fig. 9. TRx SW Test Processing Screen

iperf Client측 결과이며 Process(0)은 Server측 시험 결과이다. 송수신 SW는 리눅스와 윈도우의 터미널 환경에서 실행되어 동작하였다.

4.2 시험분석 SW 구현 결과

시험분석 SW는 C++ 기반의 Qt6 프레임워크를 이용하여 GUI 형태로 구현하였다. 따라서 윈도우와 GUI 환경을 제공하는 리눅스에서 동작이 가능하다. 그림 10은 시험분석 SW의 메인화면이다. 화면 상단에 시험분석 SW와 송수신 SW와의 연결을 위한 버튼이 있고, 중앙 좌측의 Test Command List에는 시험 명령어 목록들이 표시된다. 시험 명령어 작성을 위해 ADD/DELETE/MODIFY 버튼과 시나리오를 불러오고 저장할 수 있도록 Open List/Save List 버튼을 구성하였다. 화면 우측의 Client List에는 현재 연결된 송수신 SW 노드들의 주소가 표시된다. 하단에는 시험 수행을 위한 TEST START 버튼과 시험 결과 수집을 위한 RESULT

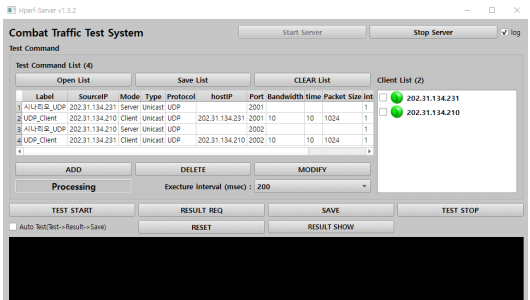


그림 10. 시험분석 SW 메인창
Fig. 10. Test & Analysis SW Main Window

REQ, 결과를 분석하고 확인하기 위한 RESULT SHOW 버튼이 있다. 그리고 최하단에는 시험 진행 및 상태를 보여주는 로그 창을 구현하였다. 중앙의 Execution interval 타임은 각 시험 명령어를 노드에 전송할 때 일정한 시간 간격을 두고 전달하기 위함이다. 이는 소켓을 통해 전달되는 시험 명령어들이 오류 없이 전달하기 위함이다.

그림 11은 시험 명령어 작성 창 구현 화면이다. 시험 시나리오 작성 창에서 각 시험 명령어를 설정하고 iperf 명령어를 생성하게 된다. 시험 명령어 작성 창에는 송신지 주소, 수신지 주소, iperf 프로세스 동작 모드, 프로토콜 종류, 전송 타입, 시험 포트, 시험 시간, 목표 데이터 대역폭, 패킷 크기를 설정할 수 있다. Templates에는 합정 네트워크 특화된 시나리오를 작성하기 위해 포 2의 항목들을 만들었다. Templates 항목에서 하나를 선택하면 해당 네트워크 트래픽 종류를 반영한 시험 명령어가 자동으로 설정된다. 추가로 중단 간 지연시간을 측정할 수 있도록 Delay 기능을 추가하였다. Delay는 Ping 프로세스를 이용하여 측정한다. 마지막 하단에는 iperf 명령어들을 수동으로 추가할 수 있도록 하였다.

그림 12는 시험분석 결과창을 나타낸다. 합정 네트워크 시험 도구는 하나의 시나리오에 대한 전체 결과를 하나의 파일로 저장한다. 시험분석 결과 창에서 Open

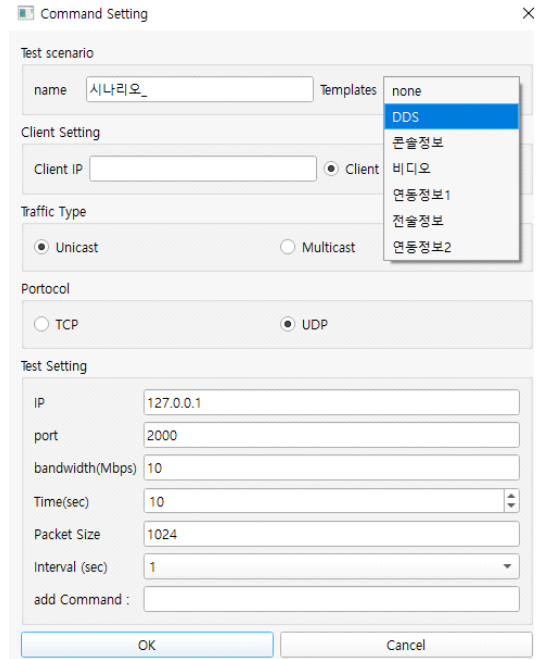


그림 11. 시험 명령어 작성 화면
Fig. 11. Test Command Setting Screen

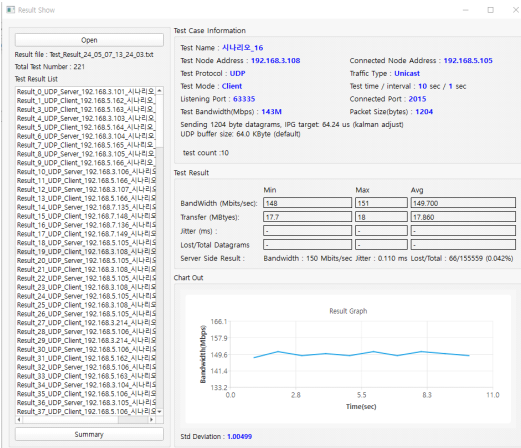


그림 12. 함정 네트워크 시험 결과 창
Fig. 12. Naval Ship Network Test Result Window

버튼을 통해 특정 시험 결과 파일을 불러오면 파일 내 모든 시험 결과를 분석하고 결과를 출력한다. 시험분석 결과창 좌측에는 시험 시나리오에 포함된 모든 시험 명령어에 대한 리스트를 출력한다. 각 항목은 순번, 동작 모드, 송신지IP주소, 시나리오라벨 형식으로 표시된다. 임의의 항목을 선택하면 우측에 각 시험 명령어 결과를 출력한다. Test Case Information 탭에서는 시험 명령어에 대한 내용을 출력한다. Test Result 탭에서는 시험 결과를 출력한다. 이때 처리량, 전송량, 지터, 패킷 손실률에 대한 최대, 최소, 평균 수치를 출력한다. Chart Out 탭에서는 시험 시간에 따른 처리량 그래프를 표시한다.

그림 13은 종합분석 결과를 보여준다. 종합분석 결과는 시나리오에 대한 분석과 시험 결과에 대한 분석으로 나누어진다. 시나리오에 대한 분석은 각 시험 명령어가 시험 종류에 따라 명령어의 갯수, 평균 전송 트래픽, 시나리오에서의 최대 수신 노드를 분석한다. 이를 통해 해당 시나리오에서 가장 트래픽 부하가 가장 큰 부분이

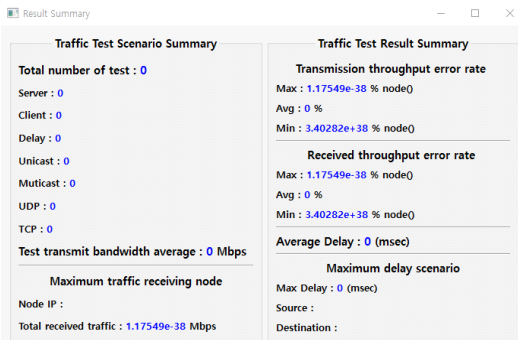


그림 13. 시험분석 SW 종합분석 창
Fig. 13. Test & Analysis SW Result Summary Window

어디인지 확인할 수 있다. 시험 결과에 대한 분석 내용은 최대/최소/평균 송수신 오차율과 해당 노드를 출력한다. 지연시간을 시험할 경우 최대 지연시간을 가지는 중단 노드 정보를 출력한다.

4.3 시험 환경 구성

본 논문에서는 제안하는 함정 네트워크 시험 도구의 검증에 전술한 4단계 시험 순서에 따라 진행한다. 시험 도구 검증을 위해 실제 함정 네트워크를 모사한 함정 네트워크 프로토타입 테스트베드를 구성하였고 함정 네트워크에서 실제로 오고가는 트래픽 시나리오를 생성하였다. 테스트베드는 그림 14처럼 차세대 함정 네트워크 설계를 위해 고려되고 있는 계층형 네트워크 구조로 구성하였다. 네트워크 장치 구성은 최상위 백본 네트워크로 백엔드 통합 네트워크 장치 2대와 각 체계를 모사하는 프런트 엔드 스위치 5대, 방화벽 1대로 구성된다. 트론트 엔드 스위치는 콘솔 네트워크, 전술 네트워크, 연동 네트워크, 전시 네트워크를 모사한다. 테스트베드의 단말 노드들은 전시기 200대, 다기능 콘솔 PC 8대, 전술서버 10대, 연동서버 4대, 운영자 단말 20대로 총 242대로 구성되며 서버 PC는 리눅스와 윈도우서버 운영체제를 사용하였고 나머지 단말 노드는 윈도우 11로 구성하였다. 백본 네트워크 장치 사이의 링크 대역폭은 40Gbits로 구성하고 프론트엔드 스위치와 단말사이 는 1Gbits 또는 10Gbits로 구성하였다.

함정 네트워크 검증 시나리오 구성은 3장에서 전술한 함정 네트워크 트래픽 특성을 기반으로 총 240개의 iperf 트래픽 시험 명령어로 구성하였다. 수신 시험 명령어 141개, 송신 시험 명령어 97개로 구성된다. 그 중 Unicast는 158개, Multicast는 80개로 구성하였고 UDP 통신은 224개, TCP 통신 14개로 구성되었다. 추가로 2개의 End-to-End 지연시간 시나리오를 추가하였다. 함정 네트워크 대부분의 트래픽은 DDS(Data Distribution Service) 통신 미들웨어를 사용하기 때문에 데이터 중심의 UDP 통신을 기반으로 한다. 각 명령어의 대역폭 설정은 표2을 참조하여 설정하였다. 트래

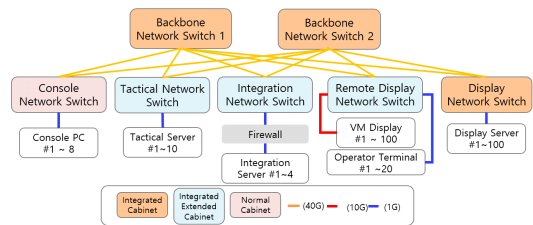


그림 14. 함정 네트워크 테스트베드 구성도
Fig. 14. Test-bed Architecture for Naval Ship Network

픽 시험은 모든 시나리오에 대해서 300초 동안 시험하였다. 시험 시간이 시험 시나리오의 갯수보다 더 클 때 모든 트래픽에 대해서 동시에 네트워크를 시험할 수 있다.

4.4 시험 결과 분석 및 평가

그림 15는 제안하는 합정 네트워크 시험 도구의 시험분석 SW 설정 화면이다. Test Command List 240개의 시험 명령어로 구성된 시나리오를 설정하였고, Client List에는 총 242개의 송수신 SW가 실행 중인 클라이언트 노드가 연결된 것을 확인할 수 있다. TEST START로 시험을 시작하였고 트래픽 시험, 결과 수집 및 저장이 순차적으로 진행되었다.

그림 16은 모든 시험 과정을 진행한 후 결과 파일로 저장된 시험 결과를 불러온 화면이다. 시험 시나리오에 대응되는 총 240개의 결과가 모두 성공적으로 수집되었으며 각각의 시험 결과의 처리량 최대/최소/평균값과,

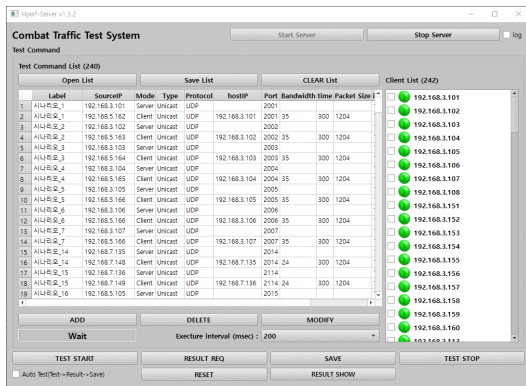


그림 15. 합정 네트워크 시험 설정 화면
Fig. 15. Naval Ship Network Test Setting Screen

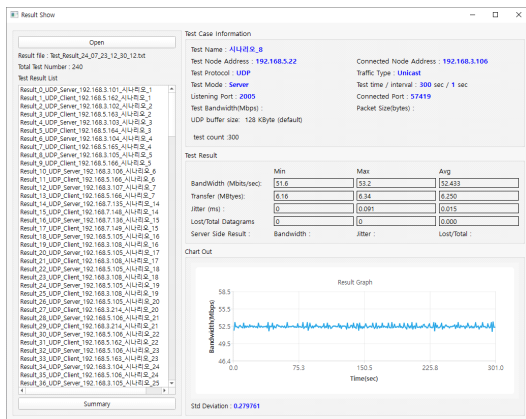


그림 16. 합정 네트워크 시험 결과 화면
Fig. 16. Naval Network Test Result Screen

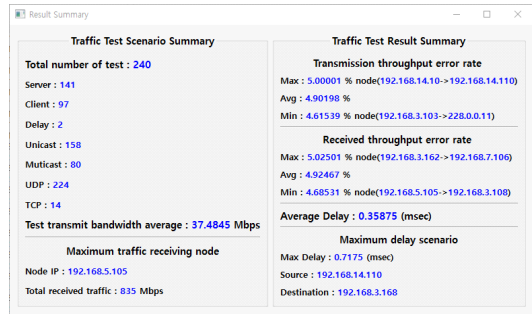


그림 17. 합정 네트워크 시험 결과분석 화면
Fig. 17. Naval Network Test Result Analysis Screen

수신 측의 경우 지터(Jitter)값, 그리고 처리량의 그래프를 확인할 수 있고 처리량의 표준편차를 확인할 수 있다. 그림 17의 결과 분석 창을 통해 시험 시나리오에 대한 전체 결과를 확인할 수 있다. Traffic Test Scenario Summary 항목에서는 앞서 구성한 시나리오대로 결과또한 동일하게 수집된것을 확인할 수 있다.

총 전송 시험 평균 트래픽은 37.4845Mbps이고, 가장 많은 트래픽을 수신한 노드는 192.168.5.105의 주소를 가지는 전시 정보노드로 835Mbps의 데이터를 수신하였다. 트래픽 시험 결과 송신 측 오차율은 평균 4.90%이며 최대 오차율은 +5%로 192.168.14.10 노드에서 192.168.14.110 노드로 전송하는 시나리오이다. 수신 측 오차율은 평균 +4.92%이며 최대 5.025%로 192.168.3.162 노드에서 192.168.7.106 노드로 8.4Mbps 전송한 경우이다. 시험 결과 모두 목표 처리량에 비해 높게 측정되면서 오차가 발생하였다. 이는 iperf로 트래픽 시험을 진행할 때 트래픽 오버헤드가 포함되면서 시험 목표로 하는 대역폭보다 처리량이 약간 높게 측정되는 것으로 분석된다. 최대 End-to-End 지연시간 측정결과는 0.7175msec 로 192.168.14.110 노드와 192.168.3.168 노드 사이에서 측정된 값이다. 본 시험을 통해 합정 네트워크 프로토타입의 구성은 시험 대역폭에 해당하는 트래픽을 정상적으로 수용할 수 있음을 검증하였다.

V. 결론

본 논문에서는 합정 네트워크의 성능 검증과 사전 시험을 위한 시험 도구를 설계하였다. 합정 네트워크를 시험하기 위해 합정 네트워크의 특성을 분석하고 시험을 위한 요구사항을 도출하였다. 그리고 요구사항에 만족하는 시험 도구를 설계 방법을 제안하였다. 제안하는 방법을 기반으로 실제로 합정 네트워크 시험 도구를 구

현하였다. 표3은 합정 네트워크 성능시험을 위한 요구 사항 대비 제안하는 시험 도구의 결과를 정리하였다. 제안하는 시험 도구는 합정 네트워크 시험 도구의 요구 사항을 모두 만족한다. 또한 본 논문에서는 제안하는 기법으로 구현한 합정 네트워크 시험 도구의 성능을 검증하기 위해 합정 네트워크 프로토타입 테스트베드환경을 구성하여 합정 네트워크 트래픽 특성을 기반으로 하는 시나리오를 작성하여 실제 시험을 진행하였고 결과를 통해 시험 도구의 유용성을 확인하였다. 기존의 네트워크 시험도구는 범용 네트워크의 성능 분석을 위해 개발되었으나 합정 네트워크 시험에 대해서는 불필요한 기능과 복잡한 사용법등 적합하지 않는 단점이 있다. 제안하는 시험 도구는 합정 네트워크 시험을 위한 요구사항을 만족하며 특수한 상황에서 네트워크를 사전 시험할 때 유용하게 사용될 수 있을 것으로 기대된다.

향후연구로는 추가로 합정 네트워크에서 요구되는 시험 요구사항을 조사하여 다양한 기능과 성능측정의 정확도를 고도화할 계획이다. 또한 다양한 체계와 다수의 합정 통합 네트워크와 같은 보다 큰 규모를 대상으로 제안하는 시험 도구를 활용하여 성능 검증 진행하고 기능을 보완할 계획이다.

표 3. 요구사항 달성결과표
Table 3. Requirements Achievement Result Table

	요구사항	제안하는 시험 도구
1	소프트웨어 구성	송수신 SW와 시험분석 SW 구분하여 구현
2	이식성	리눅스, 윈도우에서 동작 확인
3	트래픽 제어 기능	트래픽 사이즈 가변 가능, Unicast, Multicast 시험 가능 확인
4	시험 실시간성	송수신 SW 단말에서 실시간으로 시험 진행
5	시험 편의성	편의를 위한 GUI 와 파일형태로 저장 및 결과 분석 화면 제공 합정 특화 트래픽 패턴 설정 자동화

References

[1] S. Oh, "An integrated architecture for control and monitoring systems on naval surface combatants," *J. KIMST*, vol. 21, no. 1, pp. 103-114, 2018. (<https://doi.org/10.9766/KIMST.2018.21.1.103>)
 [2] G.-S. Park, B.-C. Yoo, K. Kim, and B.-W.

Choi, "A methodology for the ship system integration with open architecture: Focusing on the total ship computing environment based architecture building and validation," *J. Soc. Korea Ind. and Syst. Eng.*, vol. 43, no. 3, pp. 68-76, 2020. (<https://doi.org/10.11627/jkise.2020.43.3.068>)
 [3] S. Jeong, H. Ji, S. Choi, N. Jeong, and J. Im, "Development directions for naval system integration," *J. Soc. Naval Architects of Korea*, vol. 57, no. 1, pp. 15-20, 2000. (<https://koreascience.kr/article/JAKO202010163510007.page>)
 [4] G. Kim, M. Jang, J. Ryu, B. Jung, J. Lee, and D. Kim "Edge computing based framework for next generation naval ship combat system," *J Korean Inst. Commun. and Inf. Sci.*, vol. 47, no. 12, pp. 2078-2085, 2022. (<https://doi.org/10.7840/kics.2022.47.12.2078>)
 [5] K. Song, Y. Choi, and D. Kim, "A design of message oriented management and analysis tool for naval combat systems," *J. Korean Inst. Commun. and Inf. Sci.*, vol. 51, no. 2, pp. 197-204, 2014. (<https://doi.org/10.5573/ieie.2019.56.2.41>)
 [6] J. Y. Im and D. Kim "Performance evaluation of virtualization solution for next generation naval combat systems," *J. Korean Inst. Commun. and Inf. Sci.*, vol. 56, no. 2, pp. 2078-2085, 2019. (<https://doi.org/10.5573/ieie.2019.56.2.41>)
 [7] A. Tirumala, M. Gates, F. Qin, J. Dugan, and J. Ferguson, "Iperf- The TCP/UDP bandwidth measurement tool," from <http://dast.nlanr.net/Projects/Iperf>
 [8] V. Ndatinya, Z. Xiao, V. R. Manepalli, K. Meng, and Y. Xiao, "Network forensics analysis using wireshark," *Int. J. Sec. and Netw.*, vol. 10, no. 2, pp. 91-106, 2015. (<https://doi.org/10.1504/IJSN.2015.070421>)
 [9] Z. Weiwei, G. Jian, G. Wenjie, and C. Shaomin, "NetFlow-based network traffic monitoring," *IEEE 13th Asia-Pacific Netw. Operat. and Manag. Symp.*, pp. 1-4, Taipei, Taiwan, Sep. 2011.

(<https://doi.org/10.1109/APNOMS.2011.6076975>)

- [10] W. Kocjan and P. Beltowski, “*Learning nagios*,” Packt Publishing Ltd., pp. 10-16, 2016.
- [11] Americas, Emea, Apac, “*Solarwinds network performance monitor*(2015),” Jun. 4. 2024, from http://www.kottwitz.info/Management/SolarWinds_Network_Performance_Monitor.pdf
- [12] The Qt Framework, 2023, <https://www.qt.io/product/framework>

김재우 (Jae-Woo Kim)



2004년 2월 : 국립금오공과대학교 컴퓨터공학과 졸업
 2006년 2월 : 국립금오공과대학교 컴퓨터공학과 석사졸업
 2013년 8월 : 국립금오공과대학교 컴퓨터공학과 박사졸업
 2014년~2018년 : (주)에이알텍 광통신 연구소 차장

2018년~현재 : ICT융합특성화연구센터 연구교수.
 <관심분야> 사물 인터넷, 이동통신, 임베디드 시스템, 엣지 컴퓨팅
 [ORCID:0000-0002-2622-4219]

이민선 (Min-Seon Lee)



2023년 2월 : 국립금오공과대학교 전자IT융합전공 졸업
 2023년 3월~현재 : 국립금오공과대학교 IT융복합공학과 석사과정
 <관심분야> 실시간 시스템, 사물 인터넷, 블록체인

[ORCID:0009-0005-1277-469X]

이재민 (Jae-Min Lee)



1997년 2월 : 경북대학교 전자공학과 학사졸업
 1999년 2월 : 경북대학교 전자공학과 석사졸업
 2005년 3월 : 서울대학교 전기 및 컴퓨터공학부 박사졸업
 2004년~2016년 : 삼성전자 수석연구원

2016년~2017년 : 국립금오공과대학교 ICT융합특성화연구센터 산학협력중점교수
 2017년~현재 : 국립금오공과대학교 전자공학부 부교수
 2024년~현재 : 국립금오공과대학교 스마트국수혁신융합연구센터 센터장(과기정통부)
 <관심분야> 산업용 통신망, 네트워크 기반 임베디드 시스템 설계 및 성능분석
 [ORCID:0000-0001-6885-5185]

김동성 (Dong-Seong Kim)



1992년 : 한양대학교 전자공학과 학사 졸업
 2003년 : 서울대학교 전기 및 컴퓨터공학부 박사졸업
 2004년 : Cornell 대학교 ECE 박사 후 연구원
 2024년~현재 : 국립금오공과대학교 전자공학부 교수

2014년~현재 : ICT융합특성화연구센터 센터장(과기정통부 ITRC 및 연구재단 중점연구소)
 2014년~현재 : IEEE/ACM Senior 회원
 2019년~2022년 : 국립금오공과대학교 산학협력단장
 2022년~현재 : (주)엔에스랩 대표
 <관심분야> 실시간 통신망 및 IoT 시스템, 네트워크 기반 분산제어시스템, 실시간 S/W, 블록체인
 [ORCID:0000-0002-2977-5964]

이 재 호 (Jae-Ho Lee)



2007년 2월 : 국립금오공과대학교 전자공학부 졸업
2009년 2월 : 경북대학교 센서 및 디스플레이공학 석사졸업
2009년 3월~2011년 1월 : LG 디스플레이 OLED 사업부 연구원

2011년~현재 : 한화시스템 해양시스템1팀 해양연구소 수석연구원

<관심분야> 함정전투체계, 체계공학, 전자공학, 영상처리

[ORCID:0000-0003-0861-128X]

고 영 근 (Young-Keun Go)



2014년 2월 : 고려대학교 컴퓨터공학부 졸업
2016년 8월 : 고려대학교 컴퓨터공학과 석사졸업
2016년 10월~현재 : 국방과학연구소 선임연구원
<관심분야> 함정전투체계, 유무인복합체계, 체계공학

[ORCID:0009-0002-2433-7372]

유 재 학 (Jae-Hak Yu)



2023년 2월 : 부산대학교 전자공학부 졸업
2022년 12월~현재 : 한화시스템 해양시스템1팀 해양연구소 연구원
<관심분야> 함정전투체계, 체계공학, 전자공학, 통신공학

[ORCID:0009-0004-5835-4300]

최 낙 중 (Nak-Jung Choi)



2011년 2월 : 경북대학교 IT대학 컴퓨터학부 졸업
2013년 2월 : 경북대학교 전자전기컴퓨터학부 석사졸업
2020년 2월 : 경북대학교 컴퓨터학부 박사졸업

2018년 5월~현재 : 국방과학연구소 해양기술연구원 함정전투체계단 선임연구원

<관심분야> 함정전투체계, 유무인복합체계, 체계공학

[ORCID:0009-0005-9685-3538]