IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

IEEE SYSTEMS, MAN AND CYBERNETICS SOCIETY SECTION

Received 26 March 2025, accepted 22 April 2025, date of publication 29 April 2025, date of current version 30 May 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3565301

RESEARCH ARTICLE

Design and Implementation of Network Simulator for High-Level Naval Ship System

MIN-HUI JANG[1], HYEONG-JIN KIM[2], YOUNG-KEUN GO[3], NAK-JUNG CHOI[3], JAE-HO LEE[4], JAE-HAK YU[4], JAE-MIN LEE[1], (Member, IEEE), TAE-SOO JUN[5], AND DONG-SEONG KIM[1,2], (Senior Member, IEEE)

[1]Department of IT Convergence Engineering, Kumoh National Institute of Technology, Gumi 39177, Republic of Korea
[2]Technology Research Institute, NSLab Inc., Gumi 39177, Republic of Korea
[3]Maritime Technology Research Institute, Agency for Defense Development, Jinhae, Republic of Korea
[4]Naval Research and Development Center, Hanwha Systems, Gumi, Republic of Korea
[5]Department of Computer SW Engineering, Kumoh National Institute of Technology, Gumi 39177, Republic of Korea

Corresponding author: Dong-Seong Kim (dskim@kumoh.ac.kr)

This work was supported by the Agency for Defense Development grant funded by Korean Government.

ABSTRACT Operational naval ship systems rely on advanced sensors and equipment to detect targets and manage weapon control and engagement. However, their growing complexity, driven by technological advancements and the increasing integration of clients and servers, necessitates changes in system architecture and network structure. To address these challenges, this paper presents the design and implementation of NSNS (Network Performance Evaluation Simulator for Naval Ship Combat System), a specialized simulator for advancing research on naval ship system networks. The NSNS provides a robust environment for performance analysis by leveraging the unique characteristics of naval ship combat system nodes and network properties. Its reliability was validated through comparisons with real-world system measurements, demonstrating consistent trends and acceptable error rates. This research underscores the importance of dedicated simulation tools for validating new naval network architectures and enhancing network reliability and availability in wartime scenarios.

INDEX TERMS Naval ship network, network simulator, performance evaluation, naval combat system, DDS.

I. INTRODUCTION

The naval ship combat system is a complex weapon system that is equipped with various sensors and equipment to detect targets and perform armed control and engagement [1], [2]. From a systems perspective, these naval ship combat systems operate by mounting various systems such as sensor systems, strike systems, combat systems, and communication systems on one platform. From a control point of view, it can be divided into a combat management system, a C4I (Command, Control, Communications, Computer and Intelligence) system, an integrated institution control system, a communication system, and an administrative system [3]. As technology and information technology develop, the importance of tactical data links and networks is emerging in this naval ship system, and the war paradigm has changed to NCW (Network Centric Warfare) [4]. Therefore, in order to quickly acquire battlefield information, plan tactics, and use them in command, efforts have been made to meet increasing demands by integrating with ICT(Information and Communication Technology) [5]. Consequently, the concept of C4ISR systems has been used to support commanders in achieving information superiority by utilizing Communication and Computers as tools for effective Command and Control, and leveraging Intelligence, Surveillance, and Reconnaissance assets [6]. Today, the concept of C5ISR systems, which add Cyber to the C4ISR framework, has emerged. The goal of this system is to dominate the electromagnetic spectrum to command operations and create decisive effects anytime and anywhere [7]. This will increase

The associate editor coordinating the review of this manuscript and approving it for publication was Xiaojie Su.

the proportion of guided weapons, command control, and reconnaissance equipment. Additionally, network-connected command and control systems, real-time information sharing and data management systems, and precise and accurate battlefield visualization will be the key to the future battlefield. To establish such systems, advanced equipment and data must be integrated with technology, leading to increased complexity due to the deployment of numerous clients and servers [8]. The resulting increase in data can cause network load and delays in the process of performing tasks and affect actual wartime situations [9]. In maritime combat environments, the reliability of the network and its real-time data processing capability are critical for mission success. Increased network load can cause latency issues, which in turn slow down decision-making processes within combat systems, potentially leading to catastrophic outcomes for mission execution and survivability. To address these issues, research is being conducted on integrated management of the naval ship combat systems to enhance system accessibility and ensure availability. Furthermore, the need for research on the new network structure of the naval ships is also mentioned to efficiently integrate and manage these systems [10], [11], [12]. However, newly researched network architectures and integrated management solutions can increase system complexity from the early stages of design, leading to high costs and technical challenges during implementation [13]. In particular, technologies applied without sufficient validation may result in unexpected errors or vulnerabilities, compromising the reliability and stability of the entire system [14]. Therefore, efficient and reliable network performance analysis tools are necessary to explore and validate new network architectures. Existing network performance analysis tools fail to adequately reflect the unique characteristics of naval combat systems, underscoring the need for dedicated simulators that can overcome these limitations.

This paper identifies the need for a naval ship simulator to guide research on naval combat system networks [15]. Accordingly, a NSNS(Network Performance Evaluation Simulator for Naval Ship Combat System) was designed and implemented. This was designed to create an environment in which performance evaluation is possible by considering the characteristics of nodes, traffic, and the network that make up the naval ship combat system. Furthermore, the results of the NSNS were compared with actual measured data to analyze trends and error rates, thereby verifying how closely the NSNS outcomes align with real-world operational environments. Through this process, the reliability of NSNS was ensured, enabling users to have greater confidence in the results. Subsequently, a simulator for the final network model was developed, and its results were analyzed. This aims to provide a reliable environment for precise analysis and evaluation of performance changes based on the new network architecture and functionalities of naval vessels. Furthermore, it seeks to derive optimal solutions for the network of naval ship systems.

II. RELATED WORKS

A. LARGE-SCALE NETWORK SIMULATORS AND CASES

The naval ship system is a complex weapon system that integrates various sensors and weaponry to perform multiple functions, including combat management, target detection and tracking, and weapon control [16]. The focus of this paper is the development of a simulator for the new network research of the ship combat system. Network simulators perform an essential role in predicting and analyzing the performance of network environments. In particular, large-scale network simulators are used to model complex network behaviors and evaluate performance under various conditions. References [17] and [18] describe the development and application cases of large-scale network simulators. Reference [17] raised the issue that large-scale network simulations require a significant amount of time to manage multiple executions and efficiently allocate resources. To address this, SMO was proposed. It simplifies large-scale simulation executions for OMNet++ users and provides an automated analysis environment. Additionally, Python-based statistical tools are used to summarize and visualize simulation results, enhancing usability. Reference [18] proposed MB-IoT for evaluating the performance of IoT environments, enabling the simulation of large-scale IoT applications such as smart cities. It utilizes urban-based data and communication infrastructure data to model real-world environments and simulate packet transmission patterns, connection procedures, and transmission scheduling strategies. The simulation results provide performance metrics such as connection delay, transmission delay, and retransmission, offering directions for optimizing IoT network performance.

In this way, SMO and MB-IoT provide large-scale simulation and analysis environments for wireless and IoT networks, enhancing research productivity by offering user-friendly interfaces. These cases demonstrate that simulators in large-scale network environments must go beyond simple modeling and analysis, and provide a customized evaluation environment that reflects the specific characteristics of the system.

B. COMPARATIVE ANALYSIS AND UTILIZATION OF EXISTING NETWORK SIMULATORS

Reproducing network environments experimentally and analyzing their performance is an important means of verifying new technologies. Therefore, it is essential to select the appropriate tools considering the characteristics of the naval ship system. Reference [19] systematically analyzes and explores the features, advantages, and limitations of these tools. In doing so, it not only provides a guide for selecting the right tools but also suggests future directions. The network of the naval ship system has the characteristics of a complex system that requires high reliability and low latency. Additionally, it must meet the unique operational requirements of tactical networks, sensor networks, and others. Therefore, when selecting a simulation tool for the

naval ship system network, it is essential to consider the characteristics of the naval ship and choose an appropriate tool that can adequately reflect these characteristics.

1) OMNet++

OMNet++ is an object-oriented, event-driven simulation environment designed for testing communication protocols, multi-core applications, distributed systems, and more. It features a modular structure that allows for the development of customized models tailored to specific experimental purposes. Therefore, it is useful for designing and simulating data flow, traffic control, and inter-module communication structures in complex network environments. However, while it allows for intuitive and efficient management and analysis through a graphical user interface (IDE), the GUI can become overloaded in large-scale network environments.

2) NS-3

NS-3 is an open-source network simulator that provides a precise network simulation environment reflecting the realism and scalability of real-world network environments. It offers various wired and wireless network simulation models and allows for the definition of network topologies and events. It also supports packet capture (pcap) file analysis using tools such as Wireshark and tcpdump. Therefore, it is advantageous for large-scale network simulations and allows testing of data flow bottlenecks through realistic link performance and traffic modeling. However, as it does not provide a user-friendly interface, it may reduce work efficiency.

3) OPNET

OPNET is a commercial network simulator specialized in modeling and evaluating the performance of network components. It provides an intuitive user interface, fast simulation speed, and is well-suited for network planning and optimization. However, as a commercial software, its high cost may make it difficult to access for academic research or small organizations. Additionally, the closed-source nature of the software imposes limitations on customization.

4) GloMoSim

GloMoSim is a tool specialized for wireless network simulation, supporting both parallel and sequential modeling. It supports protocol stack-based hierarchical network modeling and provides libraries for detailed simulation of wireless and mobile networks. However, it has relatively low community support and lacks frequent updates, and additional implementation work is required for complex hybrid network simulations.

5) QualNet

QualNet is a commercial simulator based on GloMoSim, used for modeling large-scale networks and distributed applications. It provides precise models of various network devices and protocol stack layers, along with a user-friendly interface and real-time manipulation capabilities. However, as a commercial software, it has a high cost and its closed-source structure imposes limitations on customization.

C. THE NEEDS FOR PERFORMANCE EVALUATION IN THE EARLY STAGES OF LARGE-SCALE NETWORK DESIGN

In the early stages of large-scale network design, performance evaluation is challenging due to various constraints and uncertainties. As the network becomes more complex, the number of variables arising during the design process increases, often leading to evaluation and verification being deferred to the later stages of design. However, when issues are discovered in the later stages, the cost of making corrections increases significantly, and it negatively impacts the overall project timeline [20]. Therefore, various approaches are required to address this issue.

Reference [21] points out that existing evaluation methods do not adequately reflect various mission scenarios. Therefore, it evaluates the impact of network performance through scenarios such as enemy attacks, enabling the development of design strategies to maximize availability. This approach helps identify key issues that may arise in the early stages of design and provides directions for addressing them.

Reference [16] proposes a multi-layered network approach to evaluate the vulnerabilities of the ship combat system in the early stages of design, addressing this need. The paper points out that existing methodologies rely on detailed modeling and high-cost simulations in the later stages of design, failing to provide rapid evaluation in the early design phase. To improve this, it introduces a new methodology that applies network theory to assess the potential for integrated system failures. However, the paper does not adequately reflect the specific characteristics of the ship combat system, such as the combat and sensor systems. Additionally, it does not systematically present efforts to ensure consistency between the real-world environment and the simulation environment. Therefore, there is a need for methods to strengthen the reliability of simulators through comparison and validation with real-world environments.

III. ANALYSIS OF NAVAL SHIP COMBAT SYSTEM REQUIREMENTS

A. ANALYSIS OF NAVAL SHIP COMBAT SYSTEM

The naval ship system is an integrated weapon platform that monitors and evaluates tactical situations, delivering real-time tactical data from onboard equipment to enable rapid responses to detected and identified targets [22]. It is a system of systems comprising diverse components, including weapon, communication, engine, combat management, sensor, armament, data link, and navigation systems [23]. Figure 1 shows that functions and systems are configured to efficiently perform tactical situation evaluation, command judgment, armed allocation, and engagement by connecting and integrating radars, sensors, armament, and navigation

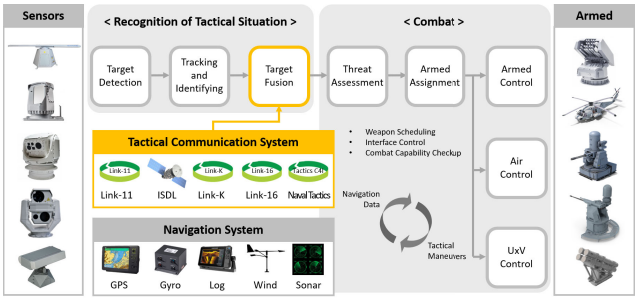


FIGURE 1. Function and structure of naval ship system.

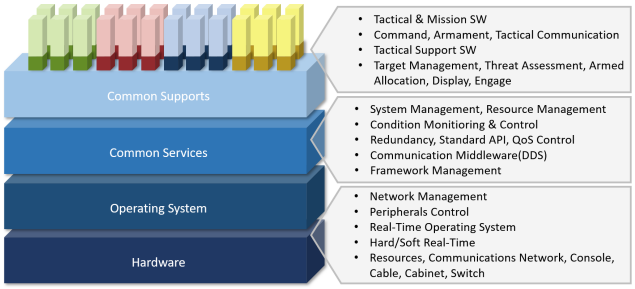


FIGURE 2. Layered architecture of the naval ship system.

support equipment through a network. Naval ships must respond to threats by quickly judging the situation in combat situations [24]. Therefore, it is equipped with various sensors, acquires them from the aforementioned data link, and compares and analyzes this information to provide reliable target information and information for precision strikes. It also provides a threat assessment function to evaluate priorities by analyzing acquired target information, and the ability to allocate optimal armed according to the response range of sensors and armed. And it provides the ability to respond quickly by integrating various armed forces onboard the ship. These features synthesize various data through a multi-functional console to recognize tactical situations and support responses tailored to mission roles. At this time, the systems constituting the existing naval ship were independently controlled for each function and mission. The ship combat system network is a complex system composed of different types of systems such as various weapon systems, communication systems, and engine systems. These systems were independently controlled. However, as demand specifications increase due to technological advances, complexity is increasing due to the installation of multiple clients and servers. The integrated ship computing environment aims to integrate various systems installed on ships through a network. This provides the basis for expanding the concept of integrating and controlling control systems such as combat systems and engine control systems, and at the same time becomes the basis for maximizing the automation capacity of networked systems. By integrating network-level systems with hardware and middleware, the ship's computing resources can be fully utilized, enabling easier development and installation of applications as needed [25].

B. ANALYSIS OF THE REQUIREMENTS OF THE NAVAL SHIP SYSTEM NETWORK

The current defense paradigm has changed to a NCW, which emphasizes the organic sharing and integration of information within and between platforms. And a naval combat system is a system that integrates and operates combat equipment installed on a warship. Accordingly, as shown in Figure 2, the naval ship system network is an interconnected network within the fleet for communication, information sharing, and operational coordination. It aims to achieve optimal combat efficiency by operating the entire fleet as a single system, rather than the naval ship combat systems operating individually. Therefore, several requirements exist [26]. Naval ships must be able to share sensor data, situation information and operational commands in real time. This requires an efficient system that can quickly process and transmit data. In addition, highly reliable communication functions that can stably transmit information are required. This demands to be robust even in combat situations and minimize situations in which the network is paralyzed [27]. Thus, to satisfy failure recovery of less than 1 second, the ship combat system network is deployed in a redundant structure. At this time, links and equipment such as core network equipment are duplicated and placed in physically separated locations. This protects the network and supports stable data transmission even in the event of a failure, disaster, or physical attack. In addition, the ship combat system consists of a complex structure and a large-scale distributed system of various equipment. Therefore, middleware is used for real-time communication based on the optimal data distribution function in a distributed system environment. This is adopted and operated as a real standard for the U.S. military and other military forces around the world through the DDS(Data Distribution Service) as shown in Figure 3 [28]. This is applied to various weapon systems, including combat systems, providing reliability and stability. And the naval ship combat system network shares data between multiple component equipment based on middleware, requiring efficient processing schemes for that data. Middleware-based networks may be transmitted simultaneously to multiple recipients who subscribe to a specific Topic. Thus, efficient mechanisms such as group management, flexible equipment addition and deletion are required, and measures to optimize them are needed. At this time, the exchanged data should take into account characteristics such as size, transmission time, and requirements, including image and armed information monitoring information. Therefore, it is necessary to configure and operate a separate network section according to the characteristics of the data. Accordingly, network management that can minimize the transmission delay of important data and optimize the overall network performance is required.

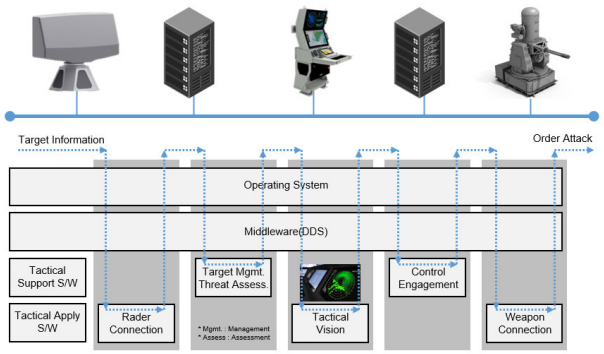


FIGURE 3. Communication structure of naval ship system.

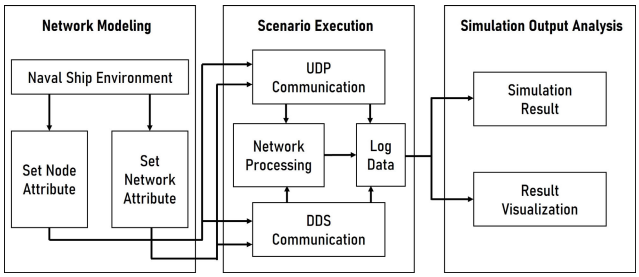


FIGURE 4. Hierarchy diagram of DDS.

IV. DESIGN OF NSNS(NETWORK PERFORMANCE EVALUATION SIMULATOR FOR NAVAL SHIP COMBAT SYSTEM)

A. ARCHITECTURE OF NSNS

Figure 4 shows the overall configuration of the simulator for the performance evaluation of the naval ship combat system network proposed in this paper. First, the network modeling part that constitutes the naval ship combat system network environment, it is composed of the part that establishes nodes in the naval ship combat system network and the part that defines the applied communication methods. A node is an entity representing a network device, such as a switch, information processing unit, radar, TV video processor, PC, or sensor. The Scenario Execution part includes a network processing process performed by the simulator based on the created nodes and scenarios. Among the communication methods used in the naval ship combat system network, it was composed of non-DDS (UDP) communication and DDS communication. It also includes a results extraction module to analyze simulation outcomes.

B. DESIGN OF NSNS

In this paper, essential functions were selected based on the requirements of the naval ship combat system network. And the class diagram of the simulator designed based on these functions is shown in Figure 5. The entire function is listed around the 'NSNS Main' class. First, it is implemented through the 'Node Manager' class to define and install each node that constitute the naval ship combat system and

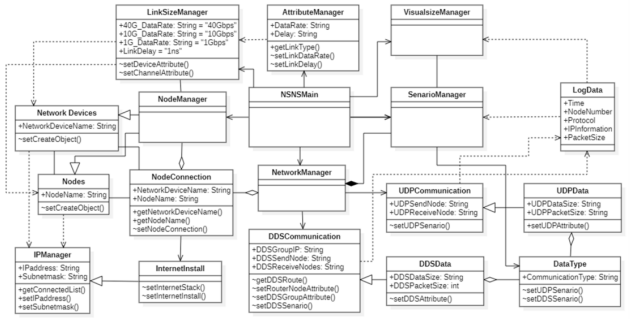


FIGURE 5. Class structure of NSNS.

subsystem. At this point, each node such as sensor, PC and switch constituting the naval ship subsystem is defined. The connection part of these nodes is implemented through the 'Link Manager' class. At this point, the characteristics of Link are defined through the 'Attribute Manager' class. In addition, the naval ship's network structure settings and communication plan transmit data through Non-DDS (UDP communication) and DDS middleware. Therefore, these functions were implemented in the 'Network Manager' class and classified into 'UDP Communication' class and 'DDS Communication' class according to the communication method. These nodes and network configurations and communication method are applied to set the scenario through the 'Scenario Manager' class. At this point, data including combat traffic was defined through the 'Data Type' class. Furthermore, when the simulation was executed, data on node information, packet size, and communication protocol information used were organized through the 'Log Data' class and used to analyze the results of the final simulation to form a simulator environment for the naval ship combat system. To implement this network simulation environment, this paper adopts NS-3 as the NSNS implementation tool. Each node in the naval ship combat system has a unique node ID and IP address. NS-3 supports this and allows for the clear identification of each node's location and role within the network simulation. Therefore, it is possible to clearly define the data transmission paths between the network devices(servers, switches) connected to each node, ensuring that data communication occurs correctly during network simulation. Additionally, by setting the attributes and latency of the links, a realistic data transmission environment can be implemented. Furthermore, since NSNS deals with the specific environment of the naval ship system, customization is essential. NS-3 can be flexibly adjusted to meet the requirements, and based on this, an optimized NSNS has been designed to enable realistic network performance analysis.

C. DESIGN OF DDS-BASED NETWORK ANALYSIS TECHNIQUES FOR STATIC COMBAT SYSTEMS

DDS is MOM(Message Oriented Middleware) that supports unbroken P2P(peer-to-peer)-based publication and subscription standardized by OMG. It supports asynchronous and

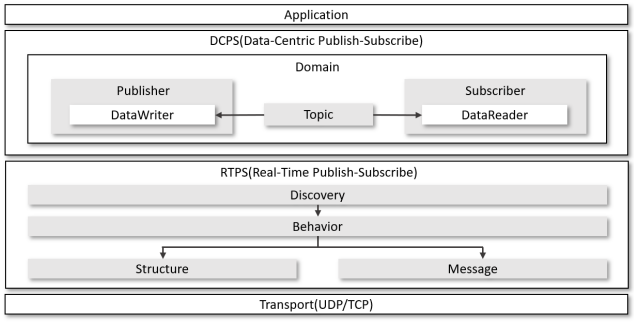


FIGURE 6. Hierarchy diagram of DDS.

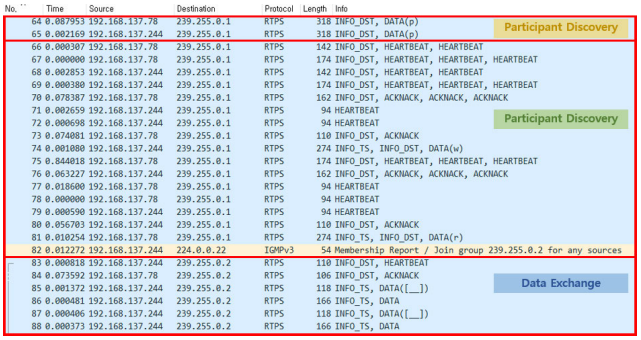


FIGURE 7. Analysis of DDS packets through Wireshark.

real-time data exchange through Topic in a distributed environment. These DDSs are composed of Transport, RTPS(Real Time Publicity Subscribe), DCPS(Data Centric Publicity Subscribe), and Application Layer structures as shown in Figure 6. At this time, the publisher sends the data by creating and submitting the data to provide the function of generating and distributing the data to be transmitted. The subscriber generates a DataReader corresponding to the DataWriter and receives data. At this time, the publisher and subscriber must be in the same DDS network domain, and the data is defined as Topic. This unbroken-P2P method is suitable for real-time systems because queue-based data exchange is not performed, and supports 22 QoS(Quality of Service) to facilitate end-point management. In addition, since there is no single point failure problem of the broker server, it is advantageous for finding stability and defects [29].

Figure 7 is the result of capturing data communication history through DDS using Wireshark, a packet analysis tool. At this time, packets exchanged through the RTPS protocol are listed in chronological order. RTPS is a key protocol for implementing DDS issuance/subscription-based data transmission and can also be used over best-effort transport layer protocols such as multicasting and UDP/IP. These RTPS supports DCPS interaction by managing registration, search, and communication of participants and endpoints for safe data transmission. It also handles discovery, message format, and transmission procedures

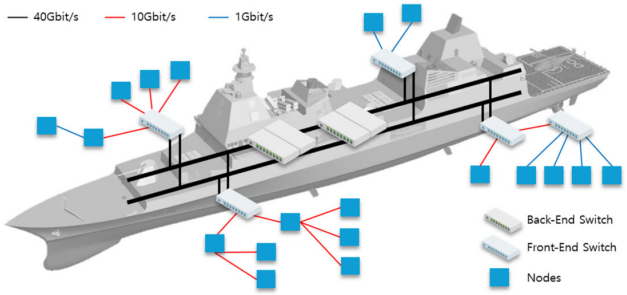


FIGURE 8. Prototype modeling of naval ship system network.

for real-time data communication in DDS to handle data flows and requirements in a real-time distributed system. Therefore, the main communication steps of RTPS are represented by Partisan Discovery, Endpoint Discovery, Data Exchange, and Heartbeat Exchange [30]. Partisan Discovery allows participants to recognize and register with each other on the network through the PDP(Participant Discovery Protocol). Endpoint Discovery manages registration and change between endpoints such as Topic, Publisher, and Subscriber through the EDP(Endpoint Discovery Protocol). Data Exchange efficiently transmits the data generated by the Publisher to the Subscriber with the Topic. At this time, data is exchanged through Unicast or Multicast through P2P communication. Heartbeat Exchange periodically checks and maintains the connection status between peers through Heartbeat messages. Therefore, in this paper, we intend to implement the DDS model in consideration of these four steps and the requirements of the ship combat system network. The ship combat system network's links operate at gigabyte capacities, handling approximately 35MB of data. Therefore, as confirmed by Wireshark, the data generated in the discovery section is 10 to 100B, which is a small percentage of about 0.000093%. Therefore, data generated during the discovery phase was omitted. Since node addition or deletion was not included, Heartbeat Exchange was limited to the framework's initial operation. And since DDS used in the ship combat system exchanges data through UDP-based Multicast, a structure was designed to simultaneously transmit data to the same Topic, that is, receivers belonging to a specific Multicast group, in consideration of the characteristics of the UDP Multicast communication method.

V. IMPLEMENTATION OF NSNS

A. MODELING THE ARCHITECTURE OF NAVAL SHIP COMBAT NETWORK

The naval ship combat system consists of various subsystems, and due to the increase in requirements for expanding automation and reducing response time, it is changing from a structure that individually controls the various systems mounted on the naval ship to a structure that integrates and controls them all. This provides an environment that facilitates rapid verification, judgment measures, and installation

of new equipment by connecting sensors and armed systems mounted on ships through a network [31]. In order to build such a naval ship system environment in a simulator, one of the networks, separated by system, was simplified and modeled as a prototype. The simulator's reliability was then evaluated by analyzing the results. Figure 8 shows the simulated environment of the naval ship system network configured for modeling in NSNS and the configuration of each node deployed. The naval ship system consists of a variety of weapon systems and surveillance equipment. Various target information and tactical information collected through various detection and tracking sensors and tactical data links are processed and fused in real time to display the integrated tactical situation. Therefore, the model used in this paper consists of nodes such as processing devices, PCs, radars, and various sensors, network devices that manage nodes, and integrated network devices that integrate and manage these devices. In this structure, a total of five network devices and lower nodes connected to them are arranged and communicated based on an integrated network device composed of redundancy. Here, the network device is composed of L3 switches, etc., and was modeled based on the benchmark for each switch. In addition, lower nodes were also modeled based on the benchmark. The communication technique that constitutes this was configured based on Ethernet and LAN, and the link was configured according to the equipment specifications and generated data capacity of the network. And the amount of data generated by each node is a large size of 40MByte, and the size of the link is configured in Gbyte units.

B. IMPLEMENTATION COMMUNICATION COMBAT NETWORK OF NAVAL SHIP

Algorithm 1 shows the process of modeling the naval ship combat system network structure including UDP-based communication functions and executing specific scenarios. The ship combat system network environment proposed in this paper requires multiple nodes to be deployed. Therefore, the algorithm was designed and implemented based on the function provided through NS-3 without a fixed limit on the number of nodes. This process is largely composed of network modeling, UDP communication, and result output stages. The network modeling stage defines the size of the link between nodes that make up the network to be built. It also defines the names and locations of the nodes that make up the network. After that, from the terminal node located at the bottom, it is connected to the parent node, P2P connection is progressed, and IP is assigned. The UDP communication stage was built to model data transmission patterns that periodically occur in the network and evaluate its performance. When the node definition is completed, each node exchanges a heartbeat message through Hearbeat Exchange. After that, preparation steps are taken to execute a UDP scenario that considers the combat traffic and data type of the ship's combat system. Since UDP communication is built based on the characteristics of unconnected inter-node communication, data transmission and reception nodes were simply defined based on the scenario. After that, the scenario was executed based on the DataRate and PacketSize required by each scenario, and at this time, the data was implemented to move according to the path.

Algorithm 1 UDP-based Naval Ship Network Modeling and Scenario Execution

```
Input: SendDataSize, SendNode, ReceiveNode
Output: Throughput, Goodput
Initialize LinkDataRate, LinkDelay, Node posisions(X,Y)
for each node (X, Y) in grid do
    connect node (X, Y) to (X, Y+1)
    assign IPv4 to each connection
end for
if Node Definition is finished then
    HeartbeatExchange();
end if
    generateUDPScenario(SendNode, ReceiveNode);
executeScenario(DateRate, PacketSize, StartTime, EndTime);
```

Algorithm 2 DDS-based Naval Ship Network Modeling and Scenario Execution

```
Input: SendDataSize, SendNode, ReceiveNode, DataRate, PacketSize,
    StartTime, EndTime
Output: Throughput, Goodput
Variables: LinkDataRate, LinkDelay, NodeName, NodeConnection, Node-
    Position_X, NodePosition_Y, p2pNodeConnection, ConnectNodePosi-
    tion
for each NodePosition_X in network grid do
    for each NodePosition_Y < max_Y do
        connect node (NodePosition_X, NodePosition_Y) to (NodePosition_X,
    NodePosition_Y+1)
        assign IPv4 to each p2p connection
    end for
end for
if Node Definition is finished then
    HeartbeatExchange();
end if
    generateUDPScenario(SendNode, ReceiveNode);
executeScenario(DateRate, PacketSize, StartTime, EndTime);
```

Algorithm 2 shows the process of modeling the ship combat system network structure including DDS-based communication functions and executing a specific scenario. In the DDS communication stage, as described in Section II, the DDS was constructed in consideration of the simulation plan. When the node definition is completed, each node exchanges a heartbeat message through HearbeatExchange(). After that, preparation steps are taken to implement a DDS scenario that considers the combat traffic and data type of the naval ship combat system. Each scenario refers to a structure in which nodes sharing the same Topic receive data. Therefore, the nodes that perform each scenario are nodes corresponding to a specific Multicast group, and they were grouped into setMulticastGroupIP(), and group IP was assigned. And based on the scenario, the data movement path was identified and the nodes for it were defined. After that, it was implemented to execute the scenario based on the data rate and packet size required by each scenario.

TABLE 1. Combat traffic transfer scenario.

No.	COMM.	Send	Receive	Data Size	Data Type
1	UDP	Node1	PC1	96	Video
2	UDP	Node2	PC2	96	Combat Information
3	UDP	Node3	PC3	96	Combat Information
4	UDP	PC3	Node1	96	Video
5	UDP	PC3	Node2	96	Surveillance
6	UDP	PC3	Node3	96	Combat Information
7	DDS	Node5	Node1 Node2 Node3	32	Engineering Control
7	DDS	Node3	PC1 Node6 Node7 Node7	24	Weapon Information

Nodes		Communication Performance			Reference Value	
num	Name	Throughput (Mbit/s)	Goodput (Mbit/s)	Data Sent (Mbit)	Goodput (Mbit/s)	Data Sent (Mbit)
0	Network Device 1_1	843.50	807.35	8044.31	807	8070
1	Network Device 1_2	842.69	806.62	8036.61	807	8070
2	Network Device 2	1851.55	1800.96	17244.08	1849	18490
3	Node 1	36.60	35.00	349.06	35	350
4	Node 2	0.00	0.00	-	-	-
5	Node 3	36.60	35.00	349.06	32	350
6	Node 4	0.00	0.00	-	-	-
7	Node 5	36.60	35.00	349.06	35	350
8	Node 6	0.00	0.00	-	-	-
9	Node 7	36.60	35.00	349.06	35	350
10	Node 8	0.00	0.00	-	-	-

FIGURE 9. Results and reference values through NSNS.

C. COMBAT TRAFFIC TRANSFER SCENARIO APPLIED TO NSNS

In this paper, the implemented NSNS is designed for the performance analysis of naval ship combat system networks. Therefore, it is necessary to estimate the amount of traffic considering the characteristics of combat traffic and select a scenario for transmitting such traffic. Table 1 shows portion of the scenarios, including the amount of combat system traffic and communication methods applied to NSNS. The traffic is composed of data that can be generated in the subsystems based on the existing naval ship combat system environment. At this point, communication methods that can be utilized were divided into Non-DDS(UDP) and DDS. Non-DDS consist of a simple 1:1 communication scenario with a UDP-based node-to-node structure. DDS is configured as a communication scenario in which data is exchanged using a Topic-based Pub/Sub structure. This involves creating multiple nodes with the same topic and receiving data from a specific node [9], [10].

D. SIMULATION RESULTS FOR PERFORMANCE ANALYSIS OF SHIP COMBAT SYSTEM NETWORK

Figure 9 shows portion of the results generated and the reference values that should be theoretically generated through conducting simulations through NSNS. These results were conducted based on the selected scenarios. The scenarios encompass the process of transmitting traffic, which is estimated by considering the characteristics of combat traffic, transmission protocols and data types. Therefore, the results

Algorithm 3 Performance Evaluation through Throughput and Goodput Calculation for Naval Ship System

```
Input: tracefile.tr
Output: Throughput, Goodput
 Variables: startTime, endTime, totalTime, totalDataSent, packetSize,
   packetCount, totalUsefulDataSent
struct Scenario{
   int num; string sendAddress; string receiveAddress;
}
while (getline(traceFile,line)):
   for each scenario in totalScenarios:
      sendAddress = getSendAddress();
      receiveAddress = getReceiveAddress();
      scenario = (sendAddress, receiveAddress)
      if (scenarioNum == scenario.num):
         firstTransmissionTime = getFirstTransmissionTime();
         lastTransmissionTime = getLastTransmissionTime();
         for each packet in line:
            totalDataSent += packetSize;
            totalUsefulDataSEnt += payloadSize;
            packetCoutn += 1;
         end for
         totalTime = lastTransmissionTime - firstTransmissionTime;
      end if
   end for
calculateThroughput(totalDataSent, totalTime);
calculateGoodput(totalDataSent, totalUsefulDataSent, totalTime);
```

that can run and verify simulations based on the selected scenario are throughput, Goodput, data transmission/reception amount, and the number of packets sent and received by each node for transmission and reception. When the simulation is completed, a TraceFile, which is a text-type log file that records all events that occurred during the simulation, is generated. TraceFile includes elements such as Timestamp, event type, node ID, packet ID, and packet size. Thus, the operation of the simulation was analyzed through these files, and the results were analyzed by calculating network performance indicators.

Algorithm 3 shows the process of calculating throughput, Goodput from these data. Throughput means the amount of data transmitted per unit time. Therefore, the size of all packets was calculated by selecting the amount of transmitted data according to the scenario in Tracefile. The total time required was calculated by extracting data based on the time interval from when the first packet was transmitted to when the last packet was transmitted. Goodput refers to the amount of useful data actually transmitted. Therefore, the amount of useful data was calculated by calculating the payload size of all packets from the data selected by scenario in Tracefile. And the total amount of data transmitted was summed up by the sizes of all packets. Also based on the scenario, values for the data throughput and total amount transmitted and received by each node that can be theoretically inferred are calculated and generated. As a result, the overall performance of the naval ship system was derived as a quantitative result. In addition, it provided an environment in which the performance results for this could be displayed directly.

And Algorithm 4 demonstrates the process of calculating the theoretical values of the throughput, as well as the total amount of data processed by each node, based on the

Algorithm 4 Theoretical Throughput and Node Data Calculation through Path Tracing in Naval Ship Network Traffic Scenarios

```
Input: SendDataSize, SendNode, EndpointNode, Count_EndpointNode
Output: path, ReferenceValue
if type(scenario) == dds:
  Count_EndpointNode += 1;
else:
  Count_EndpointNode = 1;
Function find_path_using_BFS(SendNode, EndpointNode):
  Start from SendNode
  while there are nodes to explore:
    for each neighbor of current node:
      if (neighbor == EndpointNode):
        return the constructed path;
      Mark neighbor as visited and continue exploration
    return empty path; # No valid path found
Function calculate reference_value(data transfer scenario):
  for each (SendNode, EndpointNode, SendDataSize) in data transfer scenario:
    path = find_path(SendNode, EndpointNode)
    if (path == empty):
      skip to next scenario # No valid path
    for node in path:
      if (type(scenario) == dds && node == parent_EndpointNode):
    add SendDataSize * Count_EndpointNode to ReferenceValue[node]
      else:
    add SendDataSize to ReferenceValue[node]
  return ReferenceValue;
```

given scenario. Each scenario includes information about the transmitting node, receiving node, and the amount of transmitted data, and this information was used for the calculations. A BFS queue and visit records were used to trace the path from the transmitting node to the receiving node by leveraging the connection information of the naval ship nodes configured in the simulator. During this process, the neighboring node is found by searching for p2p connection details including the current_node. If a discovered neighboring node has not been visited yet, it is added to the queue, the visit records are updated, and a path including the current_node is generated. This process is repeated until the neighboring node is the final node, at which point the path tracing is considered complete. Through this path information, the nodes in each path are calculated as much as the amount of transmission data in the scenario information, and all of these data are added up so that the reference value of each node can be finally represented. At this time, the DDS scenario was calculated considering multicast communication characteristics, and UDP was calculated considering node-to-node communication characteristics. As a result, the overall performance of the naval ship system was derived as a quantitative result. In addition, it provided an environment in which the performance results could be intuitively displayed.

VI. RELIABILITY ANALYSIS OF NSNS

A. NSNS RELIABILITY ANALYSIS BY COMPARISON WITH SURVEY DATA

In this paper, a network structure simplified as a prototype was constructed to evaluate the reliability of the results obtained from the developed NSNS. The same structure

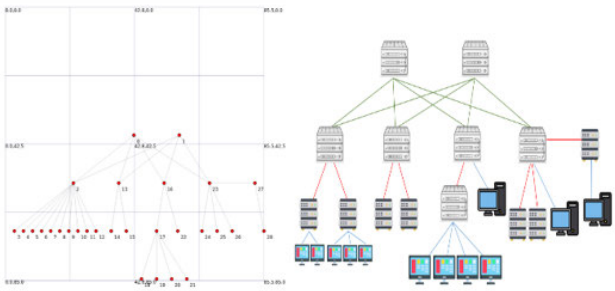


FIGURE 10. NSNS and real-world network architecture.

was implemented both in the real system environment and in the NSNS model, and the results were compared. The real system environment refers to an environment where actual equipment is connected to reproduce the naval ship network structure, which was also modeled in the same way in NSNS. The left side of the Figure 10 shows the network configured through NSNS, the right side is a picture of the network structure of the real system environment. Figure 10 illustrates the overall structure of the naval ship system network, simplified as a prototype, in a tree format. This network hardware configuration consists of network equipment and nodes. At the core of the network, it is composed of two large L3 switches. These two switches are connected in redundancy via 40Gbps links, allowing network operations to continue through the other switch if one fails. Numerous L3 switches are connected to the central switches, and these switches play a role in integrating various subsystems within the ship into the network. Each subsystem is connected by multiple nodes through 10Gbps / 1Gbps links, and these nodes represent various equipment such as radars, sensors, monitors, data processing devices, consoles, PCs, and more. The figure 9 shows the results that can be confirmed by NSNS and the results for the real system environment measured by Iperf. These results were based on the selected scenario. The scenario includes the process of calculating traffic considering the characteristics of combat traffic, transport protocols, and data types and transmitting this traffic. The left side of the figure shows the simulation through NSNS and some of the generated results and some of the reference values that should be generated theoretically. The right side of the figure shows some of the actual data results from the real system environment through Iperf, a test tool for network performance. This allows the network performance results to be confirmed by measuring network bandwidth, latency, and packet loss.

Figure 11 displays NSNS results alongside real-world system measurements from Iperf. These results were based on the selected scenario. The scenario includes the process of calculating traffic considering the characteristics of combat traffic, transport protocols, and data types and transmitting this traffic. The left side of the figure shows the simulation conducted through NSNS, displaying a portion

FIGURE 11. NSNS results and real-world environment measurement results with iperf.

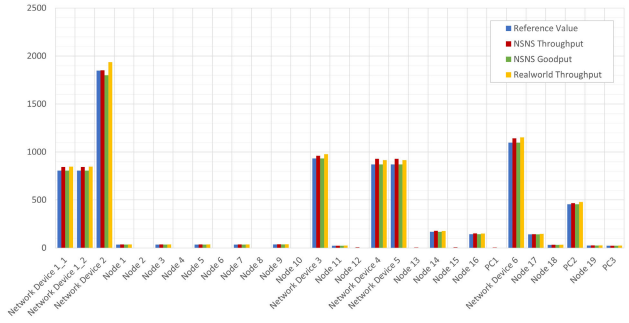


FIGURE 12. Comparison of results and reference values.

of the generated trace file. This file records network events that occurred during the simulation. Therefore, it contains network-related data such as the source and destination of packets, packet size, transmission time, packet routing, and delays. The right side of the figure shows a portion of the actual measured data results tested in the real system environment using Iperf, a tool for testing network performance. Using the Iperf tool, the traffic status test of the real system was conducted by deploying it on each node that composes the network from the central computer of the constructed environment. End-to-end traffic transmission and latency between nodes were measured to obtain the results. This allows the network performance results to be confirmed by measuring network bandwidth, latency, and packet loss.

Figure 12 shows the performance comparison of the between the results generated through the proposed NSNS and the measured data results obtained from a real system environment with the same structure and identical traffic scenarios created in the simulator. At this point, the reference values that should be theoretically generated based on the scenario were derived and compared together. The results of NSNS used here are throughput and goodput. And the results of the actual system environment can be confirmed through Iperf. This shows these results listed by nodes constituting the naval ship combat system network. As a result, it was confirmed that the results derived from NSNS were similar to the results of the actual system and the reference value that should be theoretically generated.

Figure 13 shows the error rate through comparison with theoretical reference values and comparison to verify the

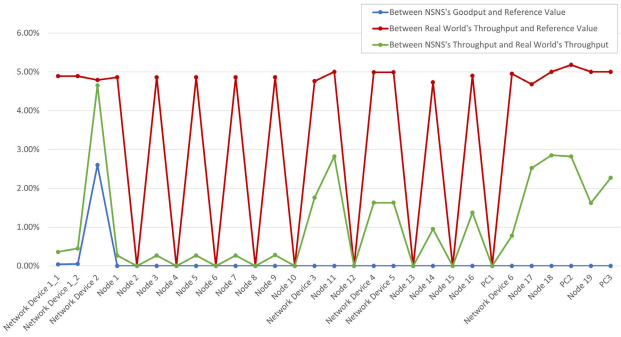


FIGURE 13. Error rate between results and ref. values.

reliability of NSNS results and actual system results through Iperf. Here, Goodput only handles substantially transmitted data, and if communication is completed normally, it should theoretically be equal to the data size that the node should send. However, because data transmission requires essential information such as headers, considering Throughput is more appropriate when applying it to the actual network communication environment. Therefore, among the two results of NSNS, the error rate for the actual system result was compared throughput, and the error rate for the reference value was compared through Goodput. In addition, the error rate between the Throughput results from the actual system and the reference values are shown in the graph. For this reason, it may appear to show values larger than the error rate when comparing NSNS with reference values. However, in this graph, it was used to observe similarity, not to compare values. Therefore, the error rate between NSNS's Goodput and the benchmark value was up to 2.6%, and the error rate between NSNS's Throughput and the actual system's Throughput was up to 4.65%. Furthermore, the error rate between NSNS and reference value exhibits a similar trend to the error rate between the actual system's Throughput and reference value.

B. NSNS RELIABILITY ANALYSIS BY COMPARISON OF OVERLOAD (BOTTLENECK) OCCURRENCE POINTS

It is necessary to analyze whether network equipment or nodes constituting the ship combat system network are being deployed and used efficiently through NSNS. This means that there is a need for an environment to check whether it is acceptable in the network environment of the ship battle system designed by the traffic and load selected based on the scenario. As a result, it is intended to provide an environment that can flexibly respond to the selection of naval ship combat system traffic and network configuration and deployment through simulation results based on parameter changes. It also aims to identify and resolve bottlenecks caused by overloading specific equipment and nodes beyond their maximum capacity. Therefore, the reliability of NSNS was analyzed by generating scenarios that deliberately overload

num	Node	NSNS Results			Reference Value	
		Throughput (Mbit/s)	Goodput (Mbit/s)	Data Sent (Mbit)	Goodput (Mbit/s)	Data Sent (Mbit)
20	Node 15	22.26	0.00	214.58	-	-
21	Node 16	738.16	707.35	7,115.69	715	7150
22	PC1	31.39	0.00	302.59	-	-
23	Network Device 6	4286.92	4148.31	40,356.25	5485	54850
24	Node 17	712.17	697.47	6,865.16	705	7050
25	Node 18	161.63	158.29	1,558.05	160	1600
26	PC2	999.99	977.56	9,413.75	2280	22800
27	Node 19	137.27	128.61	1,323.24	130	1300
28	PC3	121.88	118.72	1,174.91	120	1200

FIGURE 14. 5-fold increase of NSNS result and ref. value.

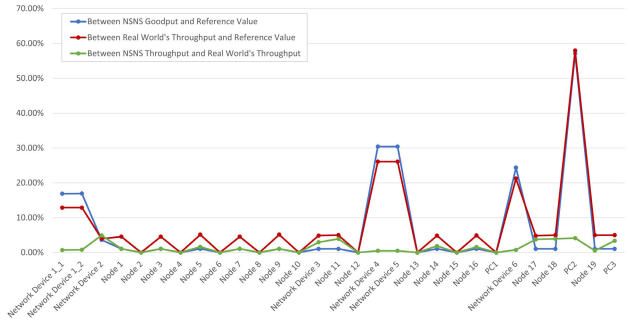


FIGURE 15. Error rate results of 5-fold increase.

and comparing them with NSNS results and actual system result data.

Figure 14 shows a portion of the results obtained by increasing the overall amount of data five times based on the scenarios conducted in Sections II and III for the maximum capacity test for traffic. And here, the point at which the overload occurs was identified based on the theoretical values, and the values were compared between the NSNS results and the actual system environment measurement results. This confirms that the NSNS results for the overload point show similarity with the actual system results where Throughput is generated up to the size of the link for that node. Furthermore, the results for each node, excluding the overload point, confirmed their similarity with the reference values that should be theoretically generated and the results from the actual system.

Figure 15 shows the NSNS results conducted by increasing the amount of data five times, the actual system results through Iperf, and the error rate through comparison with the reference value. At this time, the Goodput of NSNS was compared with the reference value, and the Throughput of NSNS was compared with the actual system results. The Goodput of NSNS was compared with the reference value, and the Throughput of NSNS was compared with the actual system results. Additionally, the comparison between the Throughput results of the actual system and the reference values allowed us to observe their similarity. Therefore, the error rate of NSNS' Goodput and reference value is up to 3.64%, excluding nodes and related network equipment, which are the points of overload. The reason for excluding

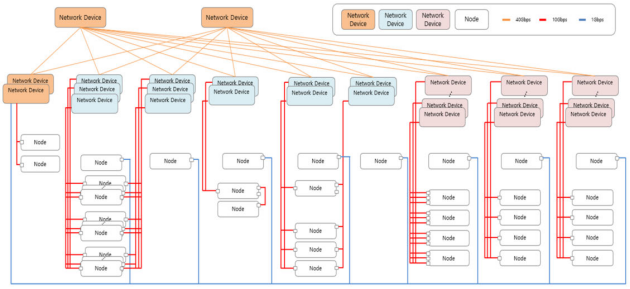


FIGURE 16. Final integrated naval ship combat network configuration.

the overload point is that the reference value does not take into account the maximum usage of the node, so the value increases without a limit. It means that the error rate naturally occurs significantly compared to the actual value set by the limit. Furthermore, the error rate between NSNS's throughput and the actual system's throughput was confirmed to be a maximum of 3.94%. Additionally, it was observed that the error rate between NSNS and the reference values exhibited a trend and similarity with the error rate between the actual system's Throughput and the reference values. As a result, NSNS was confirmed to be suitable for the environment that can develop and evaluate the network structure and functions of the naval ship as a simulator that satisfies reliability of behaving similarly to the actual system even at the overload point.

VII. FINAL INTEGRATED NAVAL SHIP COMBAT NETWORK SIMULATION AND RESULTS

A. COMPOSE OF FINAL INTEGRATED NAVAL SHIP COMBAT NETWORK

Similarity with the measured data was observed in the reliability verification through the prototype of the simulator for the performance analysis of the ship combat system network conducted in the previous section. As a result, the meaning and reliability of the simulator for ship network performance analysis implemented in this paper were proved. Based on this, a simulation for the final integrated ship network performance analysis was conducted. It configured an integrated ship combat system simulation environment with a separate network by system including multiple nodes and network devices. After that, the amount of traffic considering the characteristics of the battle traffic is newly calculated and a traffic transmission scenario is created including this to confirm the results of the network and processing performance of the integrated ship combat system. The figure 16 shows the configuration of the final integrated ship combat system to be implemented in NSNS. The network model of the final integrated ship combat system consists of a total of 42 network devices and 502 subsystem nodes connected to it, centering on the dual integrated network device. For security reasons, devices are labeled simply as network devices and nodes, but they include various FES (Front-End Switches) and BES (Back-End Switches).

TABLE 2. Combat traffic transfer scenario of final simulation.

No.	COMM	Send	Receive	Data Size	Data Type
1	UDP	Video Device1	Display Server	18Mbps	Video
2	UDP	Node2	PC2	35Mbps	Combat Information
3	DDS	Tactical Server1	OT 15 OT 10 OT 20 OT 150 OT 5200	32Mbps	Engineering Control
3	DDS	PC1	OT 5 Node6	24Mbps	Engineering Control

*OT : Operation Terminal

FIGURE 17. Visualization of final simulation.

The nodes consist of exhibitors, control devices, PCs, terminals, CCTVs, power control devices, tactical servers, exhibition servers, information storage devices, consoles, sensors, etc.

At this time, the traffic scenario is configured as shown in the figure and shows some of them. The scenario includes 512 non-DDS (UDP) cases and 45 DDS cases.

B. PERFORMANCE EVALUATION AND RESULT OF FINAL INTEGRATED NAVAL SHIP COMBAT NETWORK SIMULATION

The figure 17 visualized the network nodes, links, and packet transmission using Netanim supported by NS-3 in the scenario implemented through NSNS. Netanim is a visual representation tool of the results of NS-3 simulations. This allows intuitive observation of packet transmission, link formation and release between nodes, etc. that occur during simulation. And this visualization process allows us to ensure that the simulation settings work exactly as intended, and that the network is configured and works as expected. It also helps to analyze and correct the cause through visual cues in the event of a problem. Through this, visually confirmed whether the network was well configured through NSNS.

The figure 18 shows some of the results based on the combat traffic transmission scenario of the final integrated naval ship combat network simulation. At this time, normal communication was analyzed through the error rate of the Goodput value based on the theoretical value. And in the

Nodes		Communication Performance			Reference Value	Error Rate		Network Device Usage	
num	Name	Throughput (Mbit/s)	Goodput (Mbit/s)	Data Sent (Mbit)	(Mbit/s)	Throughput	Goodput	Maximum Throughput (Gbit/s)	Utilization Rate
6	Network Device 7	183.39	177.98	1748.98	178	3.03%	0.01%	18	1.02%
24	Network Device 25	2263.16	2209.96	21077.36	2210	2.41%	0.00%	40	5.66%
26	Network Device 27	1249.23	1216.96	11634.39	1217	2.65%	0.00%	40	3.12%
34	Network Device 35	1805.32	1750.89	16813.31	1750	3.16%	0.05%	405	0.45%
356	Node 315	10.23	10.00	97.55	10	2.29%	0.01%	-	-
401	Node 360	215.62	210.00	2056.27	210	2.67%	0.00%	-	-
513	Node 472	278.10	272.00	2652.21	272	2.24%	0.00%	-	-

FIGURE 18. Result of final simulation.

case of the network device, it is composed of FES and BES, so the point where the overload occurs based on the maximum capacity was confirmed through the Utilization Rate. At this time, as a result of the overloading point, it was confirmed that the maximum value of the error rate between the Goodput of each node and the reference value in the final integrated ship combat system simulation environment implemented by NSNS was 0.05%, and the amount of transmitted data was similar, confirming that communication was performed normally. In addition, based on the simulation results, since the network device usage rate is up to 5.66%, the traffic scenario implemented in NSNS can be expected to operate normally without delays or bottlenecks in operating in the ship-integrated network configuration implemented in NSNS.

VIII. CONCLUSION

In this paper, NSNS(Network Performance Evaluation Simulator for Naval Ship Combat System) was designed and implemented. In addition, reliability verification was performed through comparison with the results of implementing the same structure and traffic scenario in the actual environment. And this NSNS implemented an environment that combines network equipment with the lower nodes constituting the naval ship. Parameters, including communication method, transmitting and receiving nodes, data size, packet size, link capacity, and time, were used as inputs to create scenarios reflecting combat traffic characteristics. Subsequently, it conducts simulations based on the scenarios, and has the capability to derive results for the number of packets sent and received, Throughput, and Goodput for each node.

Based on the same traffic scenario, the performance was analyzed by comparing the environment in the actual system with that in NSNS. As a result, it was confirmed that a simulator with similar behavior to the real system was constructed in traffic processing. In particular, it was confirmed that throughput was observed to be equal to the maximum capacity of the overload point node at the point where overload occurred. It was confirmed that these results behaved similarly to the actual system. Therefore, the NSNS implemented in this paper allows for the prediction of network performance by reflecting various network loads and traffic conditions, enabling the identification of potential

bottlenecks and providing preemptive solutions to reduce system downtime. This approach allows for the verification of network delay and load issues caused by the network structure and system under various scenarios. Additionally, it can contribute to evaluating the practical applicability and proposing an optimal network structure. As a result, it can enhance the operational efficiency of the naval ship system and improve network availability in wartime situations. Furthermore, it can provide objective evidence for the verification and adoption of new naval ship systems.

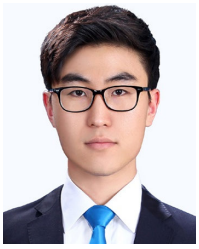
The NSNS proposed in this study focuses on network performance evaluation for the internal network of the Naval Ship Combat System. However, by being implemented based on NS3, it provides a flexible environment capable of simulating various network scenarios and traffic conditions. This enables high scalability, allowing for the modeling of not only mobile nodes such as UAVs but also various networks and systems. In addition, it offers the potential to apply algorithms for network performance optimization. Therefore, future research will aim to expand the scope of NSNS, creating an environment that can analyze performance by applying various external nodes and network algorithms, in addition to the internal systems of naval ships. Through this, a more advanced architecture for real-time performance analysis and optimization of networks will be proposed.

REFERENCES

[1] G.-S. Kim, M.-H. Jang, J.-Y. Ryu, B.-I. Jung, J.-M. Lee, and D.-S. Kim, "Edge computing based framework for next generation naval ship combat systems," *J. Korean Inst. Commun. Inf. Sci.*, vol. 47, no. 12, pp. 2078–2085, Dec. 2022.
[2] J. Y. Im and D.-S. Kim, "Performance evaluation of virtualization solution for next generation naval combat systems," *J. Inst. Electron. Inf. Eng.*, vol. 54, pp. 208–215, Feb. 2019.
[3] J. Y. Im, D. S. Kim, K. S. Song, and Y. S. Choi, "Design and realization of distributed real-time message management scheme for naval combat system development tool," *J. Inst. Control, Robot. Syst.*, vol. 22, no. 7, pp. 570–577, Jul. 2016.
[4] B. G. Kang, K.-M. Seo, and T. G. Kim, "Model-based design of defense cyber-physical systems to analyze mission effectiveness and network performance," *IEEE Access*, vol. 7, pp. 42063–42080, 2019.
[5] N. Tabish and T. Chaur-Luh, "Maritime autonomous surface ships: A review of cybersecurity challenges, countermeasures, and future perspectives," *IEEE Access*, vol. 12, pp. 17114–17136, 2024.
[6] Y.-D. Heo, "A study on the standardization of system support software in the combat management system," *J. Korea Soc. Comput. Inf.*, vol. 25, no. 11, pp. 147–155, Nov. 2020.
[7] C.-S. Baek and J.-H. Ahn, "A study of the standard interface architecture of naval combat management system," *J. Korea Soc. Comput. Inf.*, vol. 26, no. 1, pp. 147–154, Jan. 2021.
[8] M. Höyhtyä and J. Martio, "Integrated satellite–terrestrial connectivity for autonomous ships: Survey and future research directions," *Remote Sens.*, vol. 12, no. 15, p. 2057, Aug. 2020.
[9] H. J. Cha, H. K. Yang, Y. G. Jo, and H. B. Ryou, "A study on the analysis of U.S.A navy and R.O.K military C4I system and future development," *Converg. Secur. J.*, vol. 11, pp. 59–66, Dec. 2011.
[10] J. Han, Y. Kim, C. Jang, H. Lim, and J. Kim, "Conceptual design of infrastructure and framework for a futuristic surveillance imagery fusion system," *J. Korean Inst. Commun. Inf. Sci.*, vol. 46, no. 9, pp. 1426–1439, Sep. 2021.
[11] S. H. Jeong, H. G. Ji, S. W. Choi, N. S. Jeong, and J. G. Lim, "Development direction of integrated naval ship system," *J. Soc. Nav. Architects Korea*, vol. 57, pp. 15–20, Mar. 2020.
[12] S. W. Oh, "Integrated control and monitoring systems on naval ships," *Korean Inst. Inf. Technol. Mag.*, vol. 17, pp. 47–54, Nov. 2020.
[13] X. Yang, M. Liu, X. Wang, B. Hu, M. Liu, and X. Wang, "Ship network traffic engineering based on reinforcement learning," *Electronics*, vol. 13, no. 9, pp. 1710–1724, Apr. 2024.
[14] D.-C. Lee, K.-M. Seo, H.-M. Park, and B. S. Kim, "Simulation testing of maritime cyber-physical systems: Application of model-view-ViewModel," *Complexity*, vol. 2022, no. 1, pp. 1–14, Jan. 2022.
[15] H. J. Choi, "A study on the software standardization and simulator design for efficient reliability test in combat system," *J. Korea Soc. Comput. Inf.*, vol. 27, pp. 151–159, Dec. 2022.
[16] L. C. Brownlow, C. J. Goodrum, M. J. Sypniewski, J. A. Coller, and D. J. Singer, "A multilayer network approach to vulnerability assessment for early-stage naval ship design programs," *Ocean Eng.*, vol. 225, Apr. 2021, Art. no. 108731.
[17] P. A. B. Bautista, L. F. Urquiza-Aguiar, L. L. Cárdenas, and M. A. Igartua, "Large-scale simulations manager tool for OMNeT++: Expediting simulations and post-processing analysis," *IEEE Access*, vol. 8, pp. 159291–159306, 2020.
[18] V. Therrien, H. Mellah, V. Boutin, and B. Sanso, "A large-scale simulator for NB-IoT," *IEEE Access*, vol. 10, pp. 68231–68239, 2022.
[19] J. Gomez, E. F. Kfoury, J. Crichigno, and G. Srivastava, "A survey on network simulators, emulators, and testbeds used for research and education," *Comput. Netw.*, vol. 237, Dec. 2023, Art. no. 110054.
[20] H. Wen, P. Di, and T. Chen, "Warship mission reliability modeling and simulation from the perspective of equipment support resource," *J. Mar. Sci. Eng.*, vol. 11, no. 3, pp. 504–523, Feb. 2023.
[21] Y. Wang, J. Tao, X. Zhang, G. Bai, and Y. Zhang, "Mission-oriented capability evaluation for combat network based on operation loops," *Defence Technol.*, vol. 42, pp. 156–175, Dec. 2024.
[22] D. Lan, P. Xu, J. Nong, J. Song, and J. Zhao, "Application of artificial intelligence technology in vulnerability analysis of intelligent ship network, international journal of computational intelligence systems," *Int. J. Comput. Intell. Syst.*, vol. 17, pp. 1–12, Jun. 2024.
[23] S.-C. Yun and T.-S. Shon, "A study on security requirements of shipboard combat system based on threat modelling," *J. Korea Inst. Mil. Sci. Technol.*, vol. 26, no. 3, pp. 281–301, Jun. 2023.
[24] F. Li, Y. Guo, Z. Wang, Y. Chen, and J. Gu, "Hybrid dual-link data transmission based on Internet of Vessels," *Sensors*, vol. 25, no. 6, p. 1899, Mar. 2025.
[25] K. Hwang, K. Ok, Y. Kim, B. Choi, H. Oh, and K. Choi, "A study on development direction of next-generation naval combat system architecture," *J. Korea Inst. Mil. Sci. Technol.*, vol. 19, no. 1, pp. 105–118, Feb. 2016.
[26] J. H. Cha, J. M. Lee, and D.-S. Kim, "Design and implementation of a hybrid blockchain for the prevention of forgery and tampering in defense test and evaluation reports," *J. Korea Inst. Mil. Sci. Technol.*, vol. 21, pp. 103–114, Feb. 2018.
[27] D. Falcão, R. Salles, and P. Maranhão, "Performance evaluation of disruption tolerant networks on warships' tactical messages for secure transmissions," *J. Commun. Netw.*, vol. 23, no. 6, pp. 473–487, Dec. 2021.
[28] Y. K. Go and C. S. Kim, "Cryptographic overhead of DDS security for naval combat system security," in *Proc. Korea Comput. Congr.*, Jun. 2017, pp. 1217–1219.
[29] J. H. Cha, H. J. Kim, J. M. Lee, and D. S. Kim, "Trend of message-oriented middleware research for smart defense distributed systems," *J. Comput. Sci. Eng.*, vol. 41, pp. 32–39, Mar. 2023.
[30] J. H. Cha, J. W. Lee, J. M. Lee, and D.-S. Kim, "A study on the real-time middleware application and trend for ICT convergence technology in civil and military," *Inf. Commun. Mag.*, vol. 37, pp. 47–54, Sep. 2020.
[31] S. W. Oh, "An integrated architecture for control and monitoring systems on naval surface combatants," *J. Korea Inst. Mil. Sci. Technol.*, vol. 21, pp. 103–114, Feb. 2018.

MIN-HUI JANG received the B.Eng. and M.Eng. degrees in electronic engineering and IT convergence engineering from the Kumoh National Institute of Technology, South Korea, in 2019 and 2021, respectively, where she is currently pursuing the Ph.D. degree in IT convergence engineering. Her research interests include industrial networked systems, middleware, and real-time systems.

HYEONG-JIN KIM received the B.Eng. and M.Eng. degrees in electronic engineering and IT convergence engineering from the Kumoh National Institute of Technology, South Korea, in 2020 and 2021, respectively, where he is currently pursuing the Ph.D. degree in IT convergence engineering. Since February 2022, he has been the Research and Development Team Leader of NSLab Inc. His research interests include industrial blockchain applications, middleware, and real-time systems.



YOUNG-KEUN GO received the B.S. and M.S. degrees from Korea University, South Korea, in 2014 and 2016, respectively. Since October 2016, he has been a Researcher with the Agency for Defense Development. His research interests include systems, software engineering, and combat systems development.

NAK-JUNG CHOI received the B.S., M.S., and Ph.D. degrees from the School of Computer Science and Engineering, Kyungpook University, South Korea, in 2011, 2013, and 2020, respectively. Since May 2018, he has been a Researcher with the Agency for Defense Development. His research interests include naval combat systems, combat system architecture, and networks for naval combat architecture.

JAE-HO LEE received the B.Eng. degree in electronic engineering from the Kumoh National Institute of Technology, South Korea, in 2007, and the M.Eng. degree in sensor and display engineering from Kyungpook University, South Korea, in 2009. He was a Researcher with LG Display, from 2009 to 2011. Since February 2011, he has been a Researcher with Hanwha Systems. His current research interests include naval combat systems, systems engineering, electronic engineering, and image processing.

JAE-HAK YU received the B.S. degree in electronic engineering from Pusan National University, South Korea, in 2023. He has been a Researcher with Hanwha Systems. His current research interests include naval combat systems, systems engineering, electronic engineering, and communication engineering.

JAE-MIN LEE (Member, IEEE) received the Ph.D. degree in electrical and computer engineering from Seoul National University, Seoul, South Korea, in 2005. From 2005 to 2014, he was a Senior Engineer with Samsung Electronics, Suwon, South Korea. From 2015 to 2016, he was a Principal Engineer with Samsung Electronics. Since 2017, he has been an Assistant Professor with the School of Electronic Engineering and Department of IT-Convergence Engineering, Kumoh National Institute of Technology, Gyeongsangbuk-do, South Korea. His current main research interests include industrial wireless control networks, performance analysis of wireless networks, and TRIZ.

TAE-SOO JUN received the B.S. and M.S. degrees from Seoul National University, Seoul, South Korea, in 1998 and 2000, respectively, and the Ph.D. degree in computer engineering and ECE from The University of Texas at Austin, Austin, TX, USA, in 2009. From 2010 to 2022, he was with the Samsung Research, Samsung Electronics Inc., Seoul, South Korea, for smart device software platform (Tizen Project) as a Principal Engineer and IoT/AI software platform as the Director. He has been an Assistant Professor of computer software engineering with the Kumoh National Institute of Technology (KIT), South Korea, since 2022. His research interests include distributed computing in a smart environment, intelligent system design for pervasive computing, and real-time systems.

DONG-SEONG KIM (Senior Member, IEEE) received the Ph.D. degree in electrical and computer engineering from Seoul National University, Seoul, South Korea, in 2003. From 1994 to 2003, he was a full-time Researcher with ERC-ACI, Seoul National University. From March 2003 to February 2005, he was a Postdoctoral Researcher with the Wireless Network Laboratory, School of Electrical and Computer Engineering, Cornell University, NY, USA. From 2007 to 2009, he was a Visiting Professor with the Department of Computer Science, University of California at Davis, CA, USA. He was the Dean of IACF, from 2019 to 2022. He is currently a Professor with the Department of IT Convergence Engineering, School of Electronic Engineering, Kumoh National Institute of Technology, Gumi, South Korea. He is also the Director of the KIT Convergence Research Institute and the ICT Convergence Research Center (ITRC and NRF Advanced Research Center Program) supported by Korean Government at Kumoh National Institute of Technology and the Director of NSLab Company Ltd. His primary research interests include real-time IoT and smart platforms, industrial wireless control networks, networked embedded systems, Fieldbus, metaverse, and blockchain. He is a Senior Member of ACM.