

Review

Underwater Network Management System in Internet of Underwater Things: Open Challenges, Benefits, and Feasible Solution

Delphin Raj K. M ¹, Jinyoung Lee ², Eunbi Ko ³, Soo-Young Shin ², Jung-Il Namgung ⁴, Sun-Ho Yum ¹ and Soo-Hyun Park ^{3,*}

¹ Department of Financial Information Security, Kookmin University, Seoul 02707, Korea; delphinraj@kookmin.ac.kr (D.R.K.M.); junsan86@kookmin.ac.kr (S.-H.Y.)

² Special Communication Research Center, Kookmin University, 77 Jeongneung-ro, Seongbuk-gu, Seoul 02707, Korea; jylee9018@kookmin.ac.kr (J.L.); sy-shin@kookmin.ac.kr (S.-Y.S.)

³ Department of Computer Science, Kookmin University, 77 Jeongneung-ro, Seongbuk-gu, Seoul 02707, Korea; sinaa821@naver.com

⁴ Technical Engineer, BLUnomous, Kyunggido 12285, Korea; greenji@naver.com

* Correspondence: shpark21@kookmin.ac.kr

Received: 15 May 2020; Accepted: 9 July 2020; Published: 14 July 2020



Abstract: As oceans cover the majority of the earth's surface, it becomes inevitable in extending the concepts of Internet of Things (IoT) to ocean bodies, thereby tiling the way for a new drift in the digital world, the Internet of Underwater Things (IoUT). The primary objective of IoUT is the creation of a network of several smart interconnected undersea things, to digitally link water bodies by using devices such as autonomous underwater vehicles. Since the traditional ideas of IoT cannot be merely expanded to underwater, due to the difference in environmental characteristics, this puts forward a variety of challenges for scientists to work with IoUT, and one such challenge is the network management with IoUT. This paper gives an overview on (1) underwater network management systems (U-NMS) using acoustic communication in IoUT; (2) the challenges and benefits and use cases of U-NMS; (3) fault, configuration, accounting, performance, security and constrained management (FCAPSC) functionalities of U-NMS and (4) a comparison between network management system in IoT and U-NMS system in IoUT. Additionally, this paper shows the prototype design and implementation setup of U-NMS in a laboratory environment, using lightweight machine to machine (LWM2M) and acoustic communication technology for IoUT. This paper will contribute much to the profit of researchers and industry players in uncovering the critical areas of the Internet of Underwater Things.

Keywords: Internet of Things (IoT); Internet of Underwater Things (IoUT); network management system (NMS), underwater network management system (U-NMS); fault, configuration, accounting, performance, security and constrained management (FCAPSC); underwater management information base (U-MIB)

1. Introduction

The existing Internet of Things (IoT) system is coped with various heterogeneous devices that are installed with hardware and software components [1–4]. Due to the continuous growth in the number and variety of network elements, the complexity of the network management activity has become progressively stronger. In such a case, the network management system (NMS) becomes mandatory to monitor and control the devices in a heterogeneous network [5]. Most of the relevant technologies of the network management system ought to have the following characteristics: (1) it should be automatic and frequently monitor the activities of the terrestrial area network; (2) it should rapidly notify the problems

to the network management station; (3) it should be intelligent enough to handle all the resources that are available in the network; (4) it should be smart enough to identify the exact fault location in the network; (5) it should track the changes in the network for finding the major cause of the problem [6]. In wireless sensor networks, providing support to proper and effective network management is critical. The fault, configuration, and performance management functions are monitored, with the help of WSN Management [7]. NMS is the process of managing and controlling all the functions inside the network, so that it is possible to ensure that the components are functioning, and delivers trustable information to the management system [8–10]. In the 1990s, FCAPS, a framework created by ISO, distinguishes the working purposes of network management into five various levels, such as fault management (F), the configuration level management (C), the accounting level management (A), the performance level management (P) and the security level management (S) [11]. Here, a simple network management protocol (SNMP) is used for the communication between devices in IoT. In IoT, the network management system is separated into two steps: (1) device management and (2) network management; for device management and network management, numerous protocols are developed [12]. The NMS components, such as management station, manager, agent, a management protocol, and management information base (MIB) are well defined in Figure 1 [13].

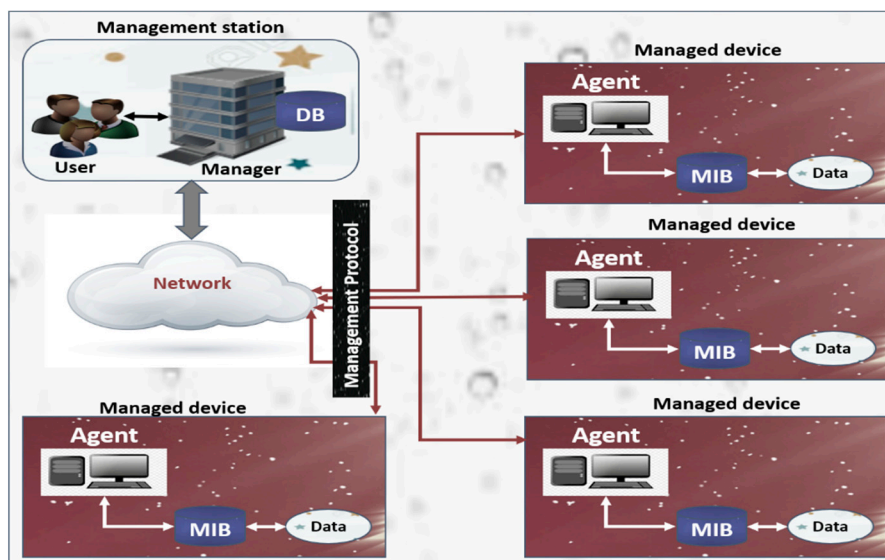


Figure 1. Network management system components.

The IoT network management system plays a significant role in the industry. Many researchers in the past have discussed their numerous ideas for designing and developing the terrestrial NMS, such as network management platform, device management protocol, network management protocol, etc., [14] and other NMS systems adapted with terrestrial IoT are reviewed underneath.

The simple network management protocol (SNMP) [15–17] is used in the terrestrial area network, to identify and solve the bugs or errors of networks. SNMP was developed by the team Internet Engineering Task Force (IETF). Abstract syntax notation one (ASN.1) [18] is the language used for defining the management information base in the network management system. User Datagram Protocol (UDP) is the supporting protocol, and OpenNMS, OpenDayLight, and Zabbix are the management platforms used in SNMP. Network configuration protocol (NETCONF) [19] was developed primarily for network configuration, and the monitoring of the terrestrial network management system. Yet Another Next Generation (YANG) is the modeling language used in NETCONF and Extensible Markup Language (XML); JavaScript Object Notation (JSON) is the supporting language of NETCONF [20,21]. The common information management protocol (CMIP) [22,23] was developed by International Organization for Standardization (ISO), which is used for the management of telecommunication networks. Guidelines for the Definition of Managed Objects (GDMO) is the

modeling language used for defining the managed objects (MOs) in CMIP. Transmission Control Protocol (TCP)/User Datagram Protocol (UDP) are the supporting protocols of CMIP. The LoWPAN Network Management Protocol (LNMP) [24–26] management architecture is suitable for Internet Protocol (IPv6) and low-power wireless personal area networks. The LNMP used the adaptation layer protocol with SNMP for the management of network devices. Message translation is possible in LNMP. UDP is the supporting protocol, and OpenNMS is the management platform of LNMP. IETF developed the restful network configuration protocol (RESTCONF) [27,28]. RESTCONF is the extraction of NETCONF, by adding a simple interface for restful communication. YANG is the data modeling language used in RESTCONF. Open vSwitch Database (OVSDB) [29–31] was developed by IETF to operate in a software-defined network (SDN). It comprises the open database server (OVSDB) and an open virtualized switch (OVS) for fast-forwarding. OVSDV supports the easy creation of interfaces by the user.

Open Mobile Alliance Lightweight M2M (OMA-LwM2M), is utilized as the device management protocol in the M2M network environment [32–34]. XML, JSON is the language used for modeling and utilizes the CoAP methods like GET, PUT, POST, and DELETE for binding in OMA-LwM2M. OMA Device Management (OMA-DM) [35–37] functions are used to provide the communication between server and client in devices, via the device management tree. This is the finest solution for remotely managing connected devices. The XML format is used for communication. Constrained application protocol (CoAP) [38–40] was developed by IETF. CoAP was specially designed for constrained devices, which is used with lower-level protocols, but it is particularly adapted over UDP/Internet Protocol Version 6 (IPv6). Its focus is mainly on finding unreachable devices. universal resource identifier (URI) is the technique used to identify the resources. Datagram Transport Layer Security (DTLS) is the supporting protocol used for high-level security in CoAP. Things Management Protocol (TMP) [41] uses the operations get/set, like the SNMP operations, to allow the interface to transmit and communicate between the things and things to the application. Web Service Definition Language (WSDL) is the language used for supporting TMP. Web API is the supporting protocol of TMP. The characteristics comparison of network management and device management protocols in the terrestrial Internet of Things (IoT) environment are presented in Table 1.

For accessing the network traffic information, the multi router traffic grapher (MRTG) is widely used in the IoT based network [42]. In Reference [43], an NMS for the large scale area network (WSNMS) is proposed to manage more than 215 nodes. The experimental results show that the performance of WSNMS is high-level. In Reference [44], in order to make a reliable management system for telecommunication networks, the Japanese company extended the framework of the telecommunication management network (TMN), with high reliability and a good quality of service. In Reference [45], the TMN based network management system with Common Object Request Broker Architecture (CORBA) and the integration are made through SNMP/CMIP for gateway-based management. In Reference [46], the ad-hoc NMS is proposed, based on the new network management technique and an upgraded embedded ARM-Linux terminal. The experimental result proved that efficiency is improved for small scale networks. In Reference [47], an intelligent agent for the telecommunication management system is proposed, by reactivating the properties of agents. Additionally, the system-level diagnosis method is used to increase the reliability in the network.

Table 1. Characteristics comparison of network management and device management protocols in terrestrial Internet of Things (IoT) environment.

Network Management Protocols in Terrestrial IoT Environment						
Protocols	Developed by	Data Modeling Language	Supporting Protocol	Management Platform	Advantages	Disadvantages
SNMP	IETF	ASN.1 and SMI	UDP	OpenNMS, OpenDayLight and Zabbix	Pros: The development of SNMP is simple and easily expandable.	Cons: In SNMP, the configuration module is not available. So, network reconfiguration is not possible. Cons: Not suitable for underwater networks
NETCONF	IETF	YANG, XML and JSON	Secure Shell (SSH)/TCP	OpenNMS and OpenDayLight.	Pros: NETCONF protocol is so strong, and its security features are high.	Cons: Data modeling and architecture for implementation was not well designed Cons: Architecture is not suitable for U-NMS
CMIP	ISO	GDMO	TCP/UDP	Solaris	Pros: Security features are high when compare to SNMP	Cons: Suitable only for wide area networks
LNMP	IETF	-	-	OpenNMS and Zabbix	Pros: Strong and architecture is suitable for reducing the cost and increasing the lifespan of networks	Cons: Delay due to the conversion between protocols
RESTCONF	IETF	YANG	SSH/Secure Sockets Layer (SSL)/Hypertext Transfer Protocol HTTP/TCP	OpenNMS and OpenDayLight.	Pros: strong security and delivers restful communication	Cons: The architecture is not well-designed for implementation support Cons: Complex architecture not suitable for solving underwater configuration problem
OVSDB	IETF	JSON	HTTP/SSL	OpenDayLight	Pros: Strong interoperability support for SDN based networks	Cons: Low-security features and not suitable for a constrained environment such as underwater networks
Device Management Protocols in Terrestrial IoT Environment						
Protocols	Developed by	Data Modeling Language	Supporting Protocol	Resource Accessibility	Advantages	Disadvantages
LwM2M	IETF	XML and JSON	UDP/HTTP/SSH/TCP	URLs	Pros: Communication model and security are very strong and can be applied for device management in U-NMS.	Cons: Does not support the heterogeneity networks.
OMA-DM	IETF	XML and JSON	UDP/HTTP/SSH/TCP	URLs	Pros: Provides the standard communication model and can be suitable for U-NMS	Cons: Security features are not supported.
CoAP	IETF	JSON	DTLS/UDP/HTTP/SSH/TCP	URLs	Pros: Supports standard communication model and security features are good	Cons: Does not support the heterogeneity networks.
TMP	IETF	WSDL	HTTP/SSH/TCP	URLs	Pros: connection is simple for requesting information	Cons: No security features are included and not powerful

The exponential growth of devices in networks becoming a big challenge in the case of infrastructure, management, device maintenance, device security, power conserving, etc. [48] The constant progress in the amount and the variety of network elements has also added to the fact that the activities of network management increase greatly [49]. In this case, the NMS in Internet of Underwater Things (IoUT) plays an important role in managing various IoUT use cases. It supports one to access the network management and device management information of IoUT networks [50]. In the last few decades, many researchers spotted wide attention in the area of IoUT, by developing numerous IoUT use cases [51]. Therefore, various underwater sensing devices such as sensors, actuators, autonomous underwater vehicles, gateways are utilized for monitoring a wide variety of IoUT use cases. The major IoUT use case, such as environmental monitoring, resource exploration, diver's safety, military security, etc. [52–69], are pointed out in Figure 2.

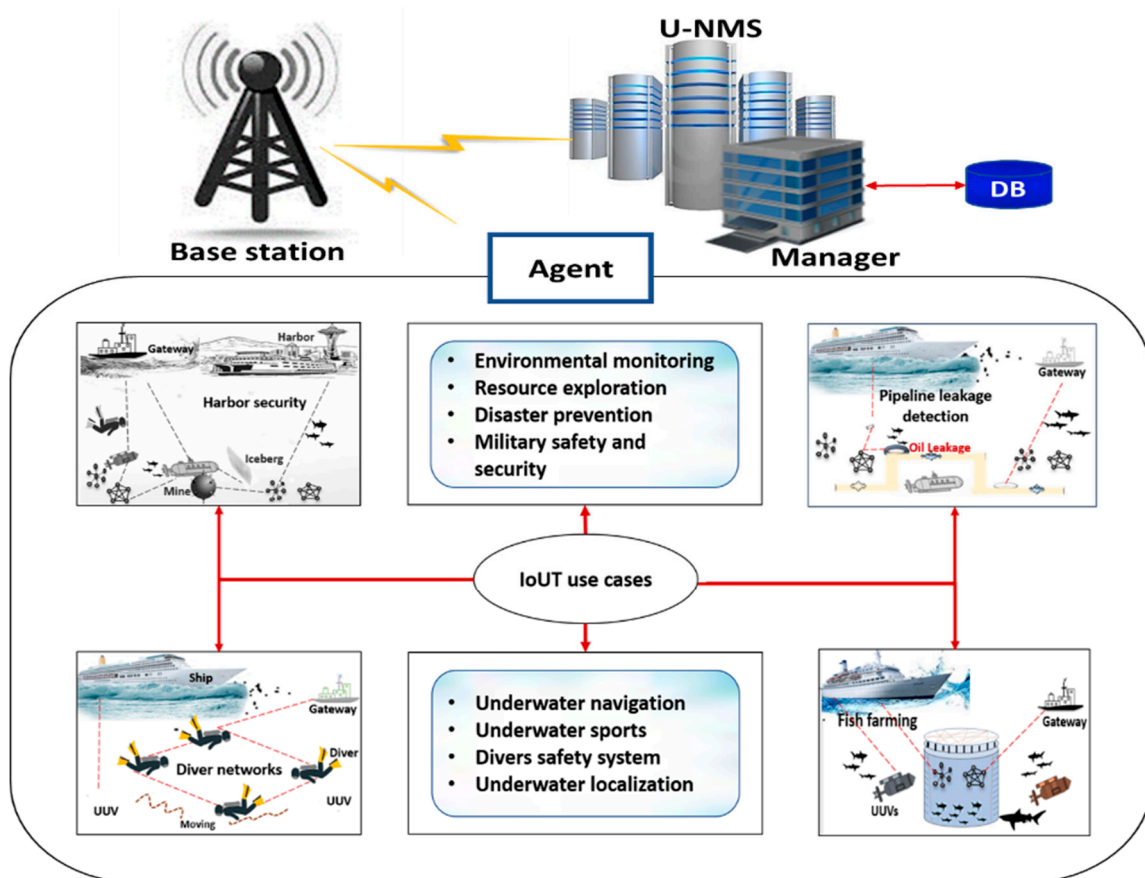


Figure 2. Internet of Underwater Things (IoUT) use cases.

However, owing to the lack of research in the area of the network management system (NMS) in IoUT [70], and the technical limitation in the IoUT environment such as connection, security management, localization, power management, memory management, low data rate, etc. are noted in [71]. The researchers focus did not touch the area of the network management system in IoUT. In the 21st century, the IoT based industries are likely to be increased [72]. Therefore, it is essential to develop the underwater network management system (U-NMS), to deliver quality services for IoUT use cases. The primary contribution of this paper can be summarized as follows:

- Overview of U-NMS based on acoustic communication technology in IoUT.
- Identify and outline the open challenges of NMS in IoUT and illustrate the benefits of U-NMS.
- Core FCAPSC functions of U-NMS in IoUT.

- The comparative analysis of the network management and device management protocols of IoT, along with the U-NMS.
- Requirements supporting U-NMS in IoUT.
- Propose a unique prototype design for U-NMS in a laboratory environment for IoUT, using acoustic communication technology.

The rest of the paper is organized as follows. Section 2 addresses the conceptual architecture of U-NMS, along with its open challenges, benefits, literature review, functions of U-NMS, and describes the comparison between NMS in IoT and U-NMS in IoUT. Section 3 describes the prototype design and implementation setup of U-NMS in a laboratory environment for IoUT. Section 4 concludes the paper, along with future work and direction.

2. Background

In this section, this paper provides an overview of U-NMS, along with an elaboration of the literature review, and discusses the different challenges, benefits, and functions of U-NMS system, along with the comparison between NMS and U-NMS.

2.1. Overview of U-NMS

U-NMS is the sub-clause derived from IoUT [73–76]. Therefore, the U-NMS extends the various unique characteristics of IoUT [77–83].

- **Intelligence:** This is concerned with merging algorithms, as well as computation, along with hardware plus software, which makes it smart enough to interact with various underwater and surface devices in an efficient manner.
- **Connectivity:** Connectivity aids in network availability and compatibility.
- **Sensing:** The main function of underwater IoT is sensing, which enables sensors in detecting any changes in the underwater environment and reporting the status or communicate with the matching environment.
- **Energy:** For the intelligent power ecosystem that is made for underwater IoT, the essential parts such as Energy harvesting, charging infrastructure, and power efficiency must be present.
- **Security:** Securing the devices, networks, and data transmission comprise a very important part of underwater IoT.
- **Node mobility:** Due to the constrained underwater environment, there is no permanent location of the sensor nodes present under sea level in the underwater wireless sensor network. The nodes are mobile, which results in improper communication.
- **Time synchronization:** Time synchronization is an essential yet challenging problem in underwater sensor networks (UWSNs). This challenge can be ascribed to (1) messaging timestamping, (2) node mobility; and, (3) the Doppler scale effect.
- **Multi-path effect:** The internal stratification of the ocean causes multi-path effects such as distortion and severe frequency-dependent fades. A multi-path medium results in a temporal spread of the transmitted pulses, as well as time-variant propagation delays and attenuation factors.
- **Stratification effect:** Localization is one of the essential tasks for underwater acoustic sensor networks (UASNs). The sound speed in water environments varies with depth; therefore, the sound waves do not propagate along a straight line. This phenomenon is described as the stratification effect.
- **Underwater noises:** Various noises underwater, such as ship noise, bio noise, wind noise, rain noise, etc. can make considerable changes in data underwater data transferrer.
- **Heterogeneity:** The major design requirements for IoUT include scalability, modularity, extensibility, and interoperability.

- Dynamic Changes: Various IoUT devices state alters dynamically according to the environment, and the system should catch up with these.

The conceptual U-NMS architecture in IoUT is the combination of different networks, such as terrestrial IoT networks and IoUT networks, as shown in Figure 3. Initially, the management station is set up in the terrestrial IoT networks, where the U-NMS can be installed. The terrestrial IoT networks are attached with components, such as management station, manager, database, etc., and devices such as base station (BS), surface gateway (S-GW), etc. which is used for monitoring the underwater networks via radio frequency (RF) and acoustic communication technologies. The IoUT network consists of intelligent underwater devices such as underwater-sensor node (UW-SNode), unmanned underwater vehicle (AUV), underwater-cluster head (UW-CH), etc. These are also installed with agent software, that is used for sensing, collecting and transferring data.

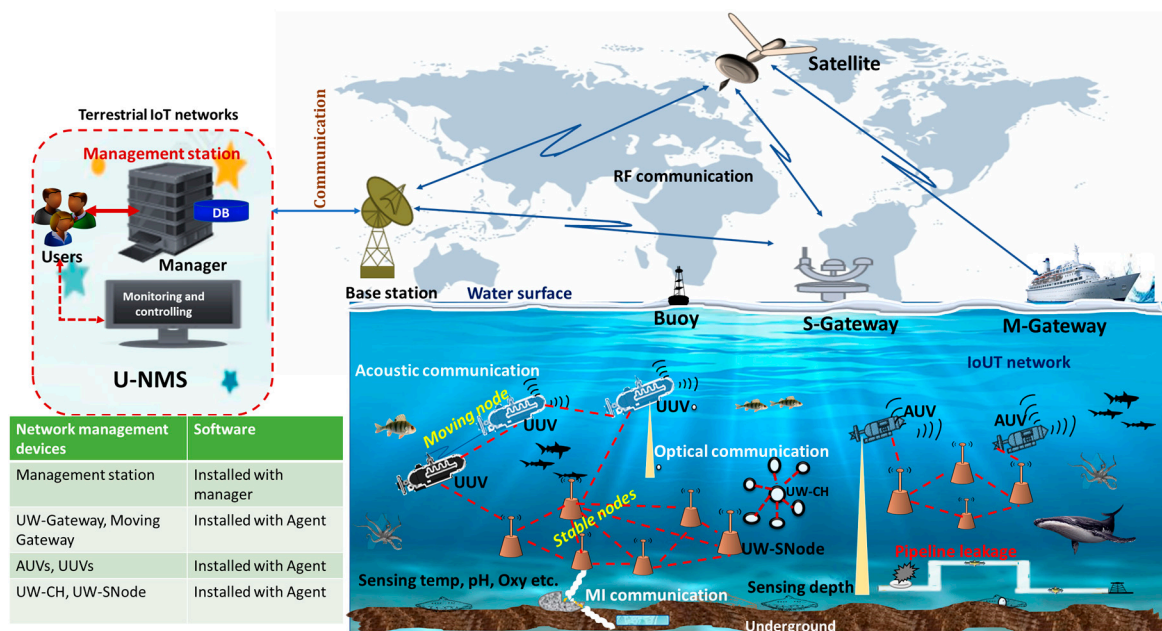


Figure 3. Conceptual architecture of underwater network management systems (U-NMS) in IoUT [79].

- S-GW: Surface gateway can be the moving gateway or fixed gateway, which is established with the Global Positioning System (GPS) to obtain their location and time for references. Especially, the task of S-GW is to provide self-localization and synchronization services to UUVs and AUVs [84–86].
- AUVs/UUVs: In U-NMS, the AUV or UUV acts as the “mobile nodes” in the underwater communication system, which provides the degree of spatial reuse mechanism for localization tasks in U-NMS. The locations of AUVs/UUVs can be estimated through direct interaction with the S-GW in the water body. The acoustic medium can be used for collecting and transferring data in U-NMS [84–86].
- UW-CH: In U-NMS, the UW-CH is used for collecting the network management information from UW-SNode and transferring those data to AUVs or UUVs via acoustic communication.
- UW-SNode: In U-NMS, the UW-SNode is used for transferring the critical message of the UW-SNode itself to UW-CH or UUVs.

2.2. Literature Review on U-NMS

To date, the literature lacks comprehensive reviews and studies on the role that U-NMS plays in the context of IoUT. The existing literature review indicates the necessity of a network management system in IoUT [50]. In Reference [70], the high-level U-NMS architecture and topology discovery mechanism is proposed for integrating terrestrial management systems into an underwater constrained

environment. The proposed architecture comprises numerous U-NMS components, such as manager, master-agent, sub-agent, management station, underwater SNMP (u-SNMP), manage objects, etc., for collecting and managing network information. It utilizes methods such as to Request, Response, and Trap to collect network management information in U-NMS, as displayed in Figure 4.

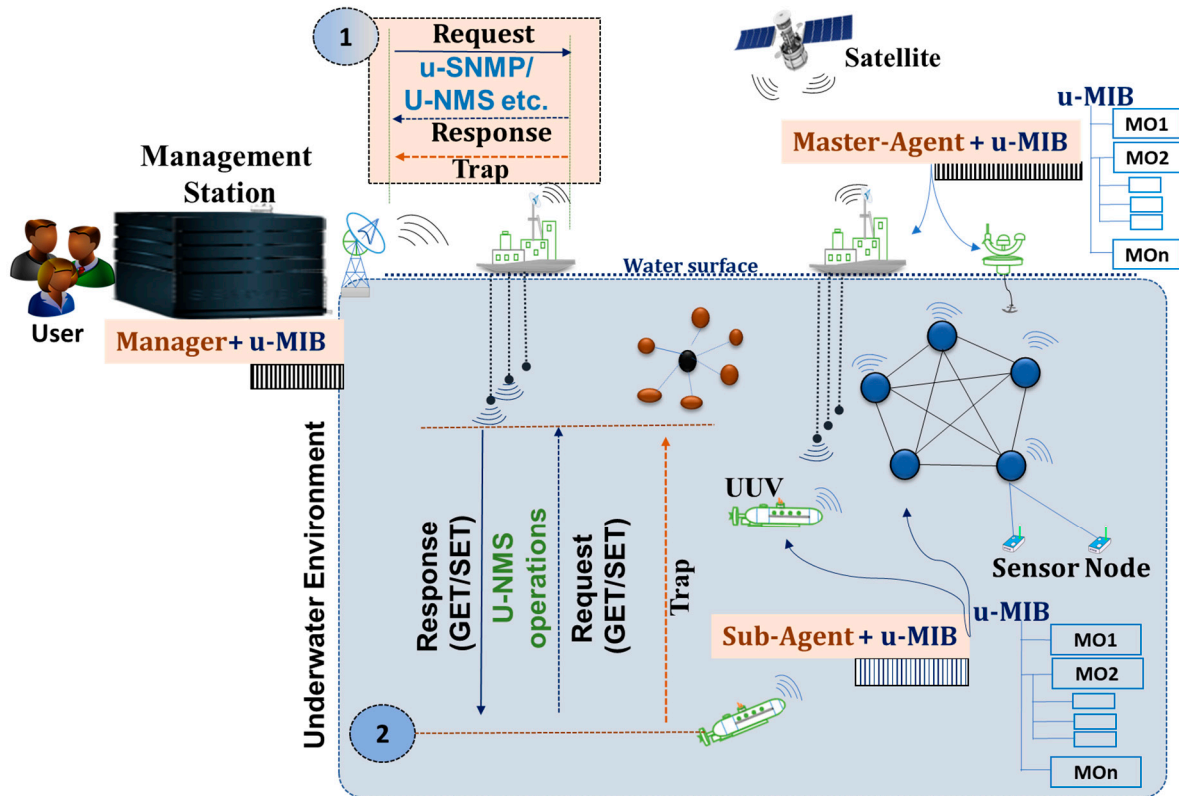


Figure 4. U-NMS components and request/response operation in IoUT [87].

In Reference [87], a couple of studies discussed the lightweight network management system with a specific IoUT function. Additionally, the underwater network management system (U-NMS) for the constrained environment was proposed. This includes newly designed architecture, the never stop mechanism, the integration of the underwater management information base (u-MIB) [88], and u-SNMP as the network management protocol, by compressing legacy SNMPv2c, carried out by Hamdamboy Urnov et al., as shown in Figure 5.

Based on the manager-agent concept in [89], the u-SNMP based network management is presented in Figure 6. In order to controll, as well as monitor, the underwater network, the u-SNMP manager makes an interface that co-operates with u-SNMP agent using u-SNMP. Based on the request from u-SNMP manager, the u-SNMP agent will send a response back. To represent the resources in the network, objects that act as data variables are used. These objects are collectively known as underwater management information base (u-MIB); this signifies features of the managed device. Almost all the actions, such as monitoring, controlling, and configuring, can be remotely done using u-MIBs that comprise definitions of management information. The u-SNMP manager monitors and controls the functions of U-NMS, by carrying the value of u-MIB via underwater networks. It also helps in modifying the configuration settings, by altering the value of certain variables, or triggering a particular action that will take place at an agent side. The operations performed using u-SNMP are depicted in Table 2.

Table 2. Operations OF u-SNMP [89].

Methods	Operations	Explanation	Value of Bits
Get Request	Read	The Get Request and Get Response method is necessary to retrieve the underwater management information base (u-MIB) value from the master agent or sub-agent of U-NMS.	00000
Get Response			00010
Set Request	Write	Necessary to change the u-MIB value of the variable using the Set Request method.	00011
Set Response			00101
Trap	Notification (Notify)	<ul style="list-style-type: none"> • A voluntary notification sent to the manager by an agent. • e.g., when a critical situation occurs, such as battery failure, memory full, etc. 	00111

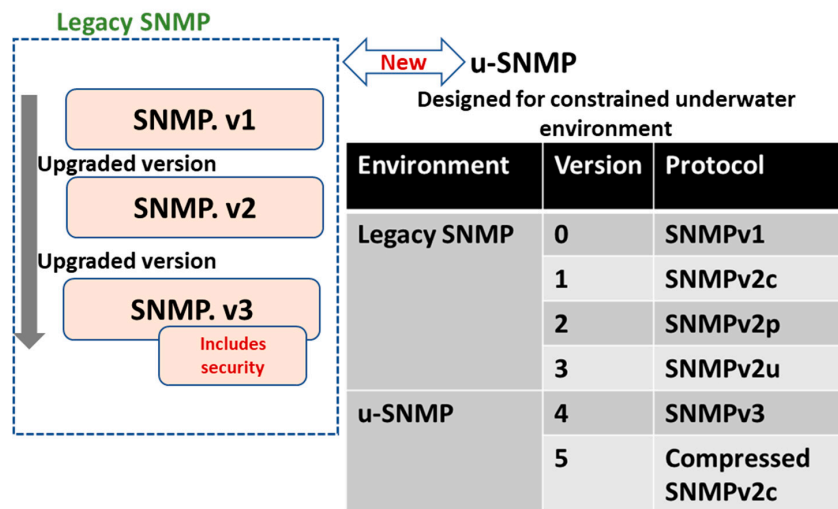


Figure 5. Generating lightweight underwater simple network management protocol (u-SNMP) by compressing legacy SNMPv2c version [89].

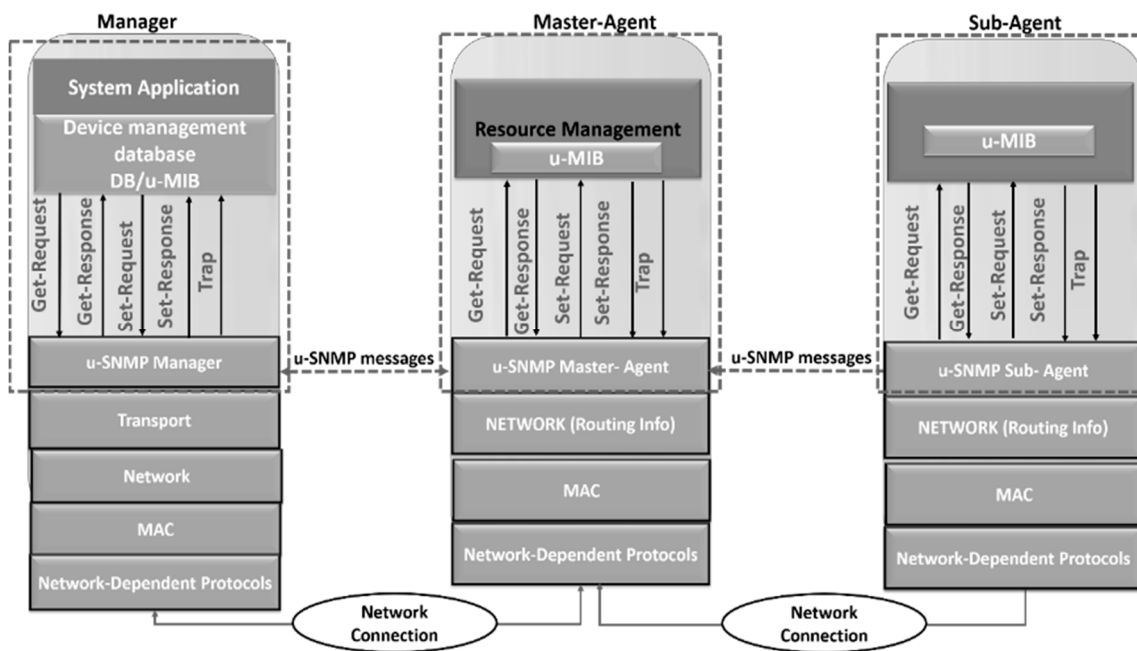


Figure 6. u-SNMP protocol in U-NMS [89].

The lightweight underwater management information base (u-MIB) module is designed based on legacy SNMP in U-NMS, as shown in Figure 7. The technical focus is to integrate the lightweight u-MIB to the manager and agent [90].

In Reference [91], the Never-stop mechanism is integrated into the constrained IoUT environment, which helps for the reliable communication in U-NMS. Figure 8 illustrates the operating of the Never-stop mechanism in U-NMS. Based on the environmental condition, bandwidth level, and device condition, the data value is dynamically measured in U-NMS. For example, the battery power level and memory space level of each device. If the data value ≥ 1 is YES (depends on the threshold value), then the process enables the Local Manager Module to perform the U-NMS functions using OID, object value, time, etc. In this case, the functional integration will perform the operation based on the request handler, and performs the response handler operation (Msg value, Req.ID, VarBind, etc.). After completing one cycle, the never-stop mechanism is used by the manager to request the MOs

repetitively in U-NMS. In contrast, data value less than one (1) converting module starts to activate (SNMP to u-SNMP), and varBind can use OPC, ODC algorithms. In progress, the sub-agent will be active based on the request handler. Functional integration has two function modules, as the fault and the constrained. Finally, the response handler will activate and respond to the master agent. Indeed, more roles are activated based on the converting module.

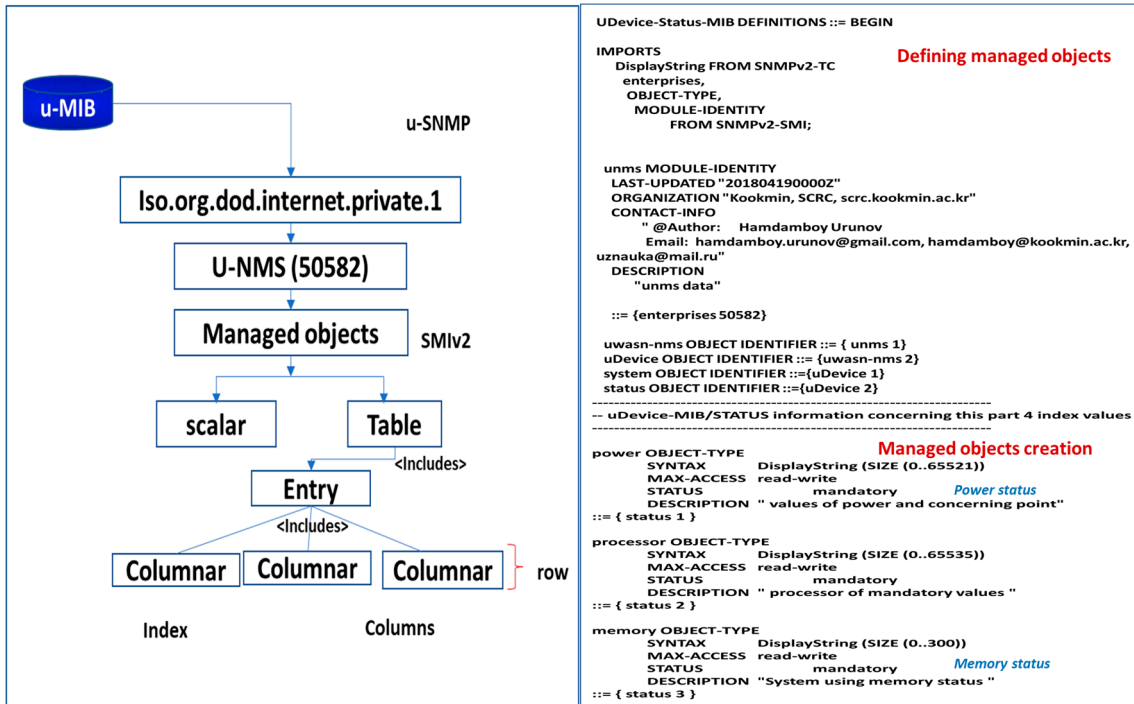


Figure 7. u-MIB designing and defining of managed objects (MOs) in u-SNMP [88,90].

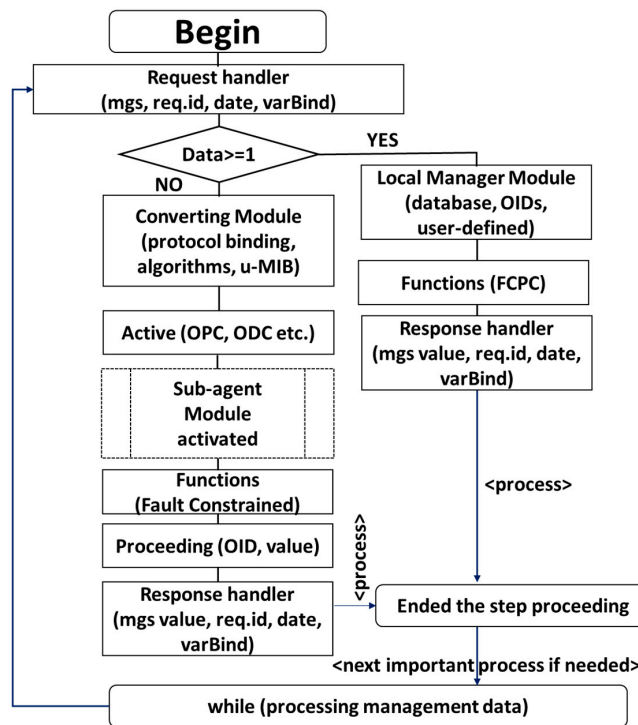


Figure 8. Never-stop mechanism [91].

2.3. Functions Supporting U-NMS

FCAPS, a framework created by ISO, distinguishes the working purposes of network management into five various levels, such as fault-management (F), the configuration level management (C), the accounting level management (A), the performance level management (P) and the security level management (S) [11]. An extension of FCAPS, FCAPSC, is proposed for network management with IoUT [89]. The detailed description of FCAPSC function, along with the research carried out so far in IoUT domain with respect to network management, is given in this section. The FCAPSC functions are detailed in Figure 9.

- Fault Management (F) techniques for IoUT: In this level, network problems are identified and corrected. Possible future issues are identified, and actions are taken to stop them from recurring or occurring. Moreover, with the help of this, the network remains operational, and downtime will be minimized.
- Configuration Management (C) techniques for IoUT: In this level, network operation is observed and controlled. Programming and hardware changes, including new equipment addition and programs, the reform of existing systems, programs, and obsolete systems removal, are coordinated. In addition to this, equipment inventory and programs are retained and updated regularly.
- Accounting Management (A) techniques for IoUT: This level, also known as the allocation level, is dedicated to optimally as well as fairly distribute resources among several network subscribers. This makes use of the systems available in an effective way, reducing the operation cost. This level also ensures that the users are appropriately billed.
- Performance Management (P) techniques for IoUT: In this level, the management system collects network statistics and evaluates the system's performance under different underwater condition.
- Security Management (S) techniques for IoUT: In this level, the network is secured against hackers, physical sabotage, unauthorized users, and electronic sabotage. The privacy of user information is preserved where it is warranted/necessary. Security systems also permit network administrators to resist what each authorized user can or cannot do to a system.
- Constrained Management (C) techniques for IoUT: This management is divided into 2 modules: (1) constrained network management and (2) constrained device management. In constrained management techniques, several steps are used to monitor and collect information from a constrained environment, such as battery power, memory level, connection status, etc., as summarized below.
 1. Lightweight mechanism: This module required a lightweight mechanism of system integration. Especially, the sustainability of message transmission should be lightweight and reduce message transmissions.
 2. The u-MIB integration: This module acts as a management database, such as lightweight enterprise MIB.
 3. Network coverage: This module integrated several modules of the function based on underwater network coverage specifications.
 4. System durability: This module can capture system durability and network possibilities, less error handling, and long-period system disabilities.

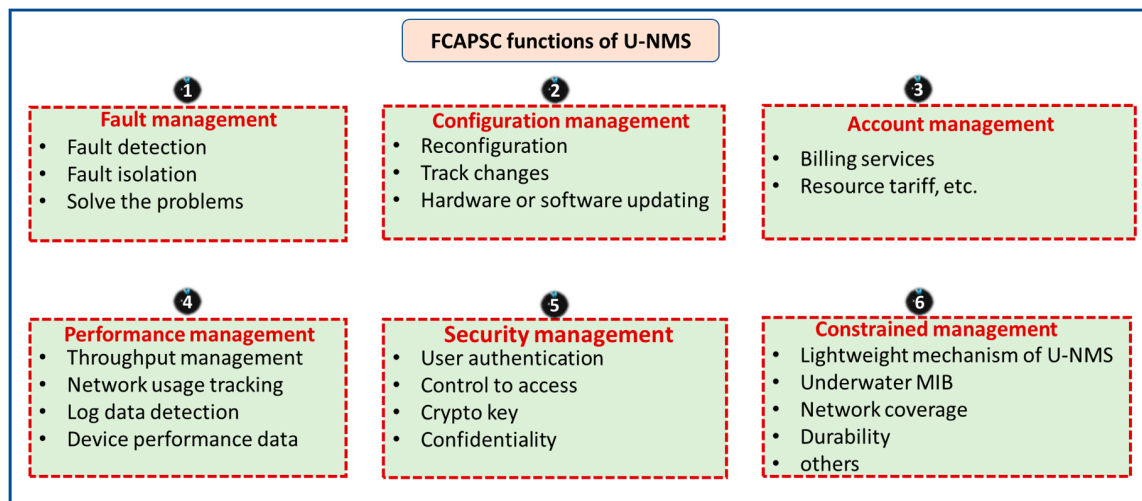


Figure 9. Functions of U-NMS [89].

2.4. Challenges and Benefits of U-NMS

As the environmental conditions of the terrestrial area network and underwater area network differ significantly, it poses numerous challenges if one tries to extend the traditional concepts of NMS to U-NMS simply. Some of the practical technical challenges are listed below:

- **Low data rates:** Using low frequencies in underwater communication is the major cause of the low data rate. In addition, there are several problems inherent to medium like reflections, energy dispersion, refraction, etc., that significantly degrade the communication among devices [92].
- **Localization in UWASN:** Since radio frequency waves are highly attenuated, localization in underwater is highly challenging. Because of this, employing technology in devices such as GPS is not viable [93].
- **Topology control:** It is used to improve the technologies in underwater communication such as localization, time synchronization, mobility management, data reduction, energy-efficient network operation, and routing. Therefore, the topology control mechanism faces a massive challenge in underwater communication [94].
- **Routing problem:** The underwater communication generally relays on the acoustic and optical medium. This consumes a high battery for long-range acoustic communication and short-range optical communication. Therefore, for energy saving, it becomes a huge challenge to build energy-efficient routing protocols for underwater communication [95].
- **Data collection:** The data-gathering methods in UWSNs are considerably different than those in WSNs, due to high battery energy consumption, high propagation delay, and so on. Many of the proposed schemes still face difficulties in reliable data collection [96].
- **Learning mechanisms:** Deep learning or machine learning mechanisms require a massive amount of data support. With the wireless communication technology, along with a huge amount of data, there is an inherent benefit of applying a deep learning mechanism. However, the broad application of deep learning mechanisms in wireless sensor networks is still not fully developed, particularly in underwater acoustic sensor networks [97]. The convolutional neural network (CNN) is the widely used deep learning technology for image recognition and signal processing in wireless communication technology. However, the various factors of underwater communication, such as low contrast, turbidity, and complex light propagation, cause difficulty in obtaining the color of underwater objects [98,99].
- **Transmission range:** Only less distance can be covered by nodes in UWASN while considering the battery power planning and network coverage. Moreover, signals are generally transmitted in fewer frequencies underwater, so that there is less chance of being absorbed by water. This

allows higher transmission ranges. However, this causes the chance for higher collision and interference [100].

- Data delivery rate: The successful delivery ratio of packets is affected considerably due to various factors such as traffic. The link reliability is typically unstable and low [100].
- Attenuation: The main reason for why attenuation is incited by absorption is the acoustic energy into heat energy conversion, which rises with distance, as well as frequency. Additionally, it is caused by reverberation and scattering (on the bottom and surface of the rough ocean), dispersion, and refraction (due to the reflection point of dispersion by surface wind). Attenuation is also determined by the depth of water depth [101].
- Depth of deployment: Underwater deployment is sparser compared with the earth bounded sensor networks (here, the lethargic deployment of sensors is done) due to the expenditure complexity [102].
- Antenna size: Due to the small size of the antenna deployed, network coverage issues persist.
- Data transmission delay: The propagation speed of underwater sensor networks is typically slow when compared to terrestrial networks. In the case of IoUT, bounded end-to-end delay ensuring can be a serious issue [103].
- Mobility: Underwater sensors can move due to water currents, and this leads to dynamic network topology changes, which is yet another challenging issue with IoUT [103].
- Cost: Compared with sensors underwater, terrestrial nodes are cheaper. This is due to the underwater transceiver's complexity and the protection of hardware in an extreme underwater environment. Moreover, sensors are highly expensive, as the suppliers give low scale economies [103].
- Limitation in performance: In the underwater network management system, there is a limitation in performance, related to the measurement such as bandwidth, throughput, latency, error rate, etc. [50]
- Problem with network configuration: In the underwater network management system, when the network size increases, there is a difficulty to update the software inside the devices. This leads to errors in the network management system [50,70].
- Problem with device security: In the underwater network management system, some attacks in devices can make the whole network collapse. So, the management system needs the security mechanism to prevent the attacks [50,70].
- Memory problem in IoUT devices: In the underwater network management system, the memory size is limited for all smart sensing devices. So, memory management is necessary for storing and retrieving information [50,70].
- Battery problem in IoUT devices: In the underwater network management system, load balancing is an essential part of smart sensing devices, since there is only limited power backup in underwater sensor nodes, and, a higher power has to be needed in the case of higher distance data transferring and complicated signal processing at the receiver. Therefore, high recharging cost is one of the major issues underwater [50,70].

To overcome the U-NMS challenges, such as battery problems, memory problems, bandwidth problems, etc., the underwater device management system and underwater network management system should be designed by considering various factors, as shown in Figure 10.

- Underwater network monitoring and controlling, utilized for managing and controlling the underwater network-based functions, such as link quality, connectivity, network security, etc., in U-NMS.
- Underwater device monitoring and controlling utilized for controlling the underwater device-based functions, such as device name, ID, etc., in U-NMS.
- Function monitoring: utilized for monitoring all FCAPSC functions of U-NMS.

- Easy problem solving utilized for finding and solving the problems of networks and devices in U-NMS.
- Location management: utilized for finding the location of each device in U-NMS.
- Resource management: utilized for managing resources such as power level, memory space, etc., in U-NMS.
- Connectivity management: utilized for controlling the connection and communication between the devices underwater.
- Others: others required for extending other functionalities such as traffic management, security management, performance management, etc., in U-NMS.

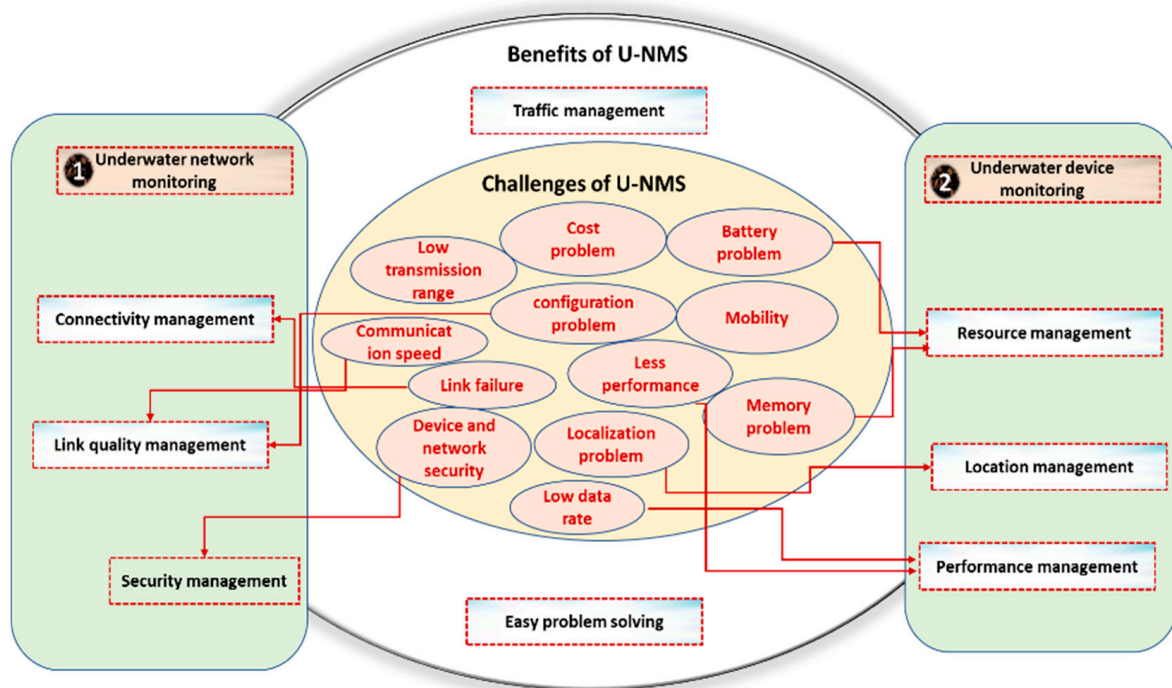


Figure 10. Benefits of using U-NMS.

2.5. Comparison Between NMS and U-NMS

This subsection depicts the comparison between NMS and U-NMS: Table 3 describes the environment setup comparison between NMS and U-NMS, Table 4 describes the network management protocol comparison between NMS and U-NMS, and Table 5 describes the device management protocol comparison between NMS and U-NMS.

Table 3. Environmental setup between NMS and U-NMS [73–75,89].

Parameters	OMA-LwM2M	Device Management Consideration in Underwater (u-SNMP)		
		Sub-Agent	Master-Agent	Metrics Value
Environment	Terrestrial environment: Adopted with wire/wireless communication	Underwater environment: adopted with acoustic communication		Speed of the signal ≈ 1.5 km/s
Communication delay	Low Radio Frequency (3.1 ns)	High in acoustic signal		>0.64 ms
Doppler spread	Not considered	Deeply considered		$\Delta f \approx 21$ Hz
Reflection	Low	powerful		≈ 22 ms
Noise	no impact	Impact (various factors)		>17 db
Battery used/power resources	Yes/ Rechargeable	Yes/ Lithium battery	few/ Lithium battery	>4000 mAh– $10,000$ mAh/ 10–18 V
Memory usage	limited	extremely limited	limited	>500 MB/ 2 GB
Transport layer protocol	TCP/UDP	none	UDP/none	-
Addressing	IPv6/ 6LowPAN (16 bytes)	None	UID	>1 byte
Bandwidth usage	Narrow	Extremely Narrow		$<\approx 10$ kHz
Range of communication	Small distance (<1 km)	Long-distance		≈ 1000 km

Table 4. Comparison between network management protocols of NMS and U-NMS. [15–18,87–89].

Areas	NMS Protocols		u-SNMP
	SNMP	TMN	
Reliability	This is based on the User Datagram Protocol and cannot assure proper transmission of data	TCP and UDP are supported, and it guarantees message delivery	Bundle protocols can support the reliable transmission of data in an underwater environment
MIB language	An adapted subset of ASN.1, Structure of Management Information (SMI) is used in SNMP for defining the objects	It is based on the object-oriented paradigm. The standard that helps in defining objects is Guidelines for Definition of Managed Objects (GDMO)	ASN.1 and SMI is used for defining the objects
Complexity	Simple design as well as architecture	Comprehensive, but complex data modeling and abstraction	Complexity for designing a framework for underwater management protocol design
Cost	Cost-effective because architecture is simple	Costly due to complex architecture	Deployment in an underwater environment is quite expensive
Protocol stack	SNMP is lightweight because limited operations are used in SNMP	CMIP is the protocol stack used in TMN; it is heavy because more operations are used in CMIP	Lightweight and limited operations for U-NMS
Operations	M Get, M action, M delete, M set, M create, M event report	Get request, Set Request, Get Response, Get next Request, Trap, Get next Response, Get Set Response	Get Request, Trap, Get Response, Set Request, Set Response
Supported Functions	FCAPS	FCAPS	FCAPSC

Table 5. Device management protocol comparison with U-NMS [32–37,87,89].

Areas	Device Management Protocols of IoT		U-NMS
	OMA-DM	OMA-LwM2M	
Devices applicable	Mobile phones, tablets, and M2M gateways.	Constrained local wireless or M2M cellular based devices	Constrained IoUT devices such as UW-SNode, surface Gateway, UUVs, etc
Data model	DM tree with very compound management objects. Moreover, Server has to know each device’s management tree structure it can manage	Flat as well as simple objects which have uniform URL between devices	DM tree should be lightweight with limited managed objects (MOs) by considering U-NMS. u-MIB defined for underwater is used to find the objects in underwater. A resource can be located using URLs
Communication model	EXEC, GET, PUT, POST, SHOW, DELETE	DELETE, GET, PUT, POST	Get Request, Trap, Get Response, Set Request, Set Response
Transport layer protocol	UDP	UDP	Bundle protocol
Network layer	IPv6, 6LoWPAN	IPv6, 6LoWPAN	u-6LoWPAN
Process layer stack	HTTPS SSH/TCP	HTTPS SSH/TCP	HTTPS SSH/UDP/TCP
Data encoding	XML, JSON	XML, JSON	XML, JSON, ASN.1
Important Factors	Integrates RESTful, management and scalability	This protocol depends on Bootstrap server URI and also, access control objects Object/Device. Configuration: selective configuration is supported, and application-specific objects from a large pool of OMA/IPSO objects are also available	It is necessary to reduce the message size structure and PDU header. Make lightweight MIB (u-MIB) structure for U-NMS

3. Proposed U-NMS System

This section presents a detailed description of the new prototype design for U-NMS, with general architecture, requirements supporting the proposed U-NMS, laboratory environmental test, and emulation results.

3.1. Conceptual Architecture of the Proposed System

Figure 11 shows the proposed U-NMS architecture based on LWM2M [32–34]. This architecture consists of four main components: U-NMS server (LWM2M-based server), U-NMS client (LWM2M-based gateway), U-NMS application, and IoUT devices. A group of IoUT device sensors, which are connected to LWM2M-based gateway, provide underwater sensor services established on device information, such as device identity, device name, memory status, temperature level, battery level, etc. U-NMS client (LWM2M-based gateway) has dual characters: it controls underwater devices, and acts as the client to U-NMS server (LWM2M-based server). U-NMS server maintains a collection of managed objects from U-NMS client, performs management operations, and is stored inside the U-NMS server database (DB). U-NMS is the kind of application used in U-NMS server for the management of networks and devices underwater. The users can collect the network and device state information of IoUT devices. Table 6 shows the requirements supporting the design of U-NMS.

Table 6. Requirements of proposed U-NMS.

Requirements of U-NMS	Description
User/Manager console	It can be the user or machine that can control and manage all the operations of the manager via management station
U-NMS	Application use in the U-NMS server
Network management station	The placeholder for database management, user and manager
Database (DB)	The data storage of U-NMS
U-NMS server (Manager)	The software installed in the management station for interacting with the agent in U-NMS
Agent	The application which is generally installed in the network management devices
U-NMS client (Master agent)	The application which is installed in the surface gateway
sub agent	The application which is installed in IoUT devices
Managed Objects (MOs)	MOs are the collects of objects in underwater management information base that can be accessed using object identifiers (OIDs)
Object Identity (OID)	Unique identity number of each objects in U-NMS
Localization methods	The localization methods need to be applied in U-NMS for identifying the position of devices in underwater environment
Device configuration	In U-NMS, device configuration mechanism is used for making the hardware and software configuration in U-NMS
Operations	These are the methods used in U-NMS for sending or receiving the management information from underwater networks. The methods that can be used are GET, GETNext, SET, SETNext, and TRAP
FCAPSC	FCAPSC is the function supportable for underwater network management system
Resource availability checking	To avoid the failure of IoUT devices, the resources such as power status, link status, memory status, etc. need to be checked frequently
Timeliness	Timeliness is used to check the service quality in U-NMS
Network status monitoring	To identify the connection between the devices in IoUT environment
Performance analysis	In U-NMS, it is necessary to identify the performance of each IoUT device
Device information collection	It is necessary to collect the device information such as ID, name, address, etc
Self-resource discovery	Identify the resource availability of each IoUT device and notify that information to U-NMS server via U-NMS client

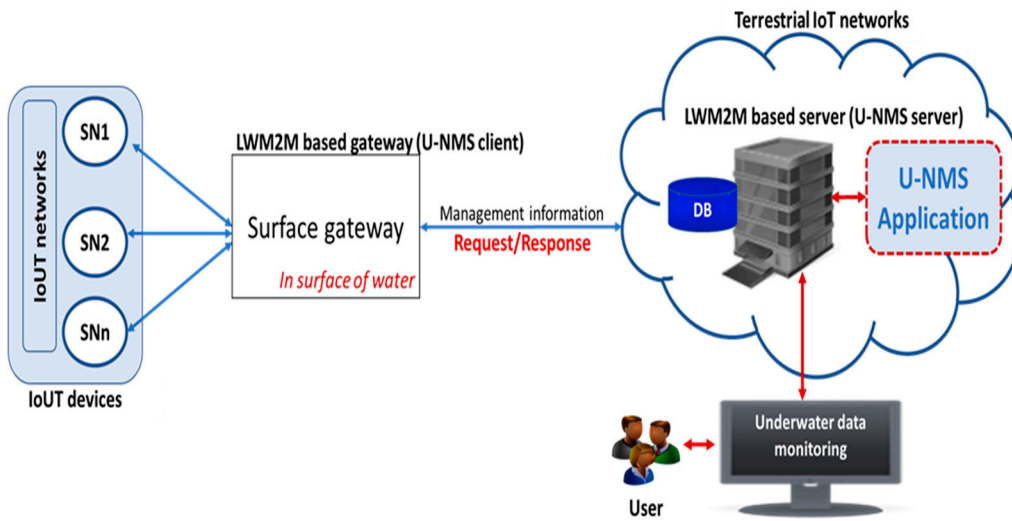


Figure 11. Conceptual architecture of LWM2M based U-NMS.

3.2. Designing Mechanism of the Surface Gateway in U-NMS

Figure 12 illustrates the software block diagram of the LWM2M based gateway in U-NMS. This block diagram consists of three compartments: IoUT networks and devices, which is installed with subagent and provides the device resource information, such as device ID, device name, device battery, etc., or (R1, R2, R3) to the U-NMS client. The U-NMS client is installed with the master agent, which comprises the object manager and resource manager. The object manager is stored with existing objects, and has space for storing new objects using OIDs. The resource manager will collect the resources provided by IoUT devices and store them in the pool as R1, R2, R3, etc. The object manager assigns OIDs for the newly found resources in the pool and store as the new MOs in U-NMS client. The U-NMS server that is installed with the manager is used to collect or manage the MOs of U-NMS client, using GET/SET methods.

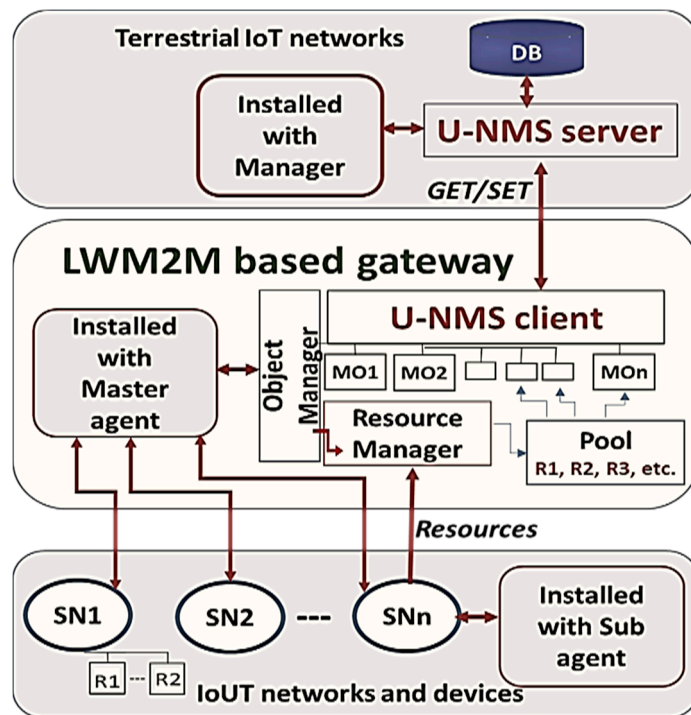


Figure 12. Block diagram of LWM2M based gateway design in U-NMS.

Figure 13 shows the message sequence process of the LWM2M based gateway (U-NMS client). When it begins, the U-NMS client registers its existence and objects to U-NMS. Once the connection is established between U-NMS client and IoUT devices, the object manager in the U-NMS client will start the resource discovery process. In accordance with the results, the object manager dynamically creates OIDs for the newly discovered resources. A U-NMS server can retrieve information about created objects by searching for the objects in U-NMS client list. After creating OIDs for the new resources, the U-NMS server can GET/SET messages regarding specific resources of the managed objects. These messages will be mapped to the subsequent operation in the object manager. As a result, the U-NMS client obtains the most recent data generated by the IoUT devices. The received information is finally delivered to the U-NMS server.

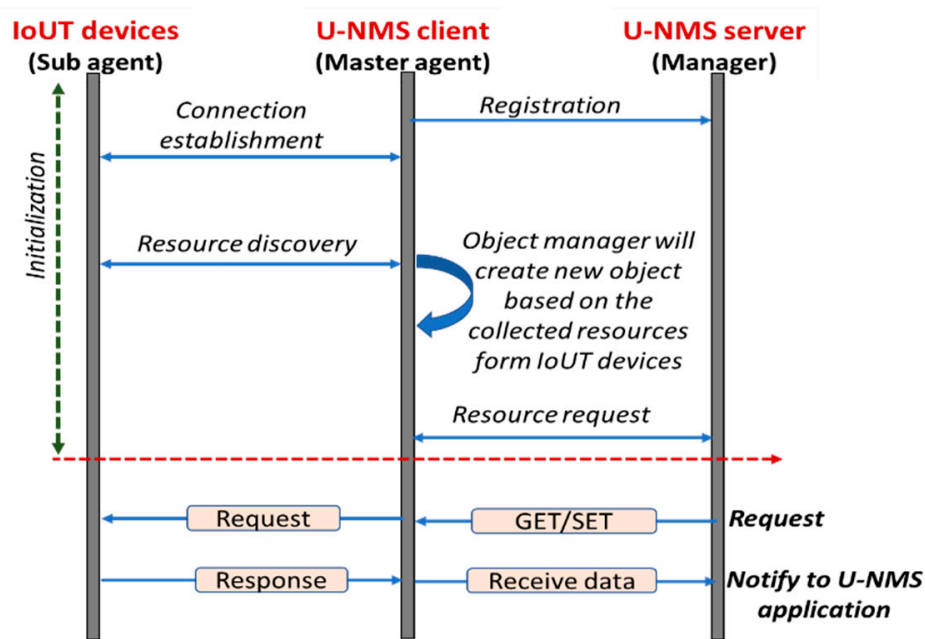


Figure 13. Sequence diagram for creating GET/SET operation in U-NMS.

3.3. Prototype Design and Managed Objects Defining

U-NMS prototype is designed with different components, such as U-NMS client, U-NMS server, manager, agent, underwater devices, etc. The components used for the design process are as follows:

1. U-NMS server: It consists of the manager program, which is used to manage the underwater networks and devices. For example, the operations such as GET/SET are used to get the information from U-NMS client in U-NMS.
2. U-NMS client: It consists of a master agent program, which is used to send notification messages when some critical event occurs, and process the response message based on the request from the U-NMS server.
3. Service Provider: In the proposed design, the LTE cat. M1 is the service provider that acts as the interface between user and surface-gateway to connect IoUT devices.
4. Surface gateway: In our approach, the surface gateway acts as the U-NMS client.
 - a. Device profile: It is the acoustic modem connected with acoustic transducer.
 - b. MOs: The U-NMS client consists of a collection of managed objects (MOs). The managed objects can be accessed using the object identifier (OID).
5. Underwater devices: The underwater devices are installed with the master agent, which can send the current resource information to U-NMS client.

Figure 14 depicts the prototype design of U-NMS. The ‘a’ side of the figure consists of the ‘manager’ part that contains the U-NMS server, which is used to obtain the necessary information, or to update the information using the methods such as GET/SET in U-NMS, and the ‘b’ side of the figure consists of the ‘master agent’ part that contains LTE Cat.M1 modem emulator, which uses UDP based LwM2M (CoAP) for the communication between U-NMS server and U-NMS client.

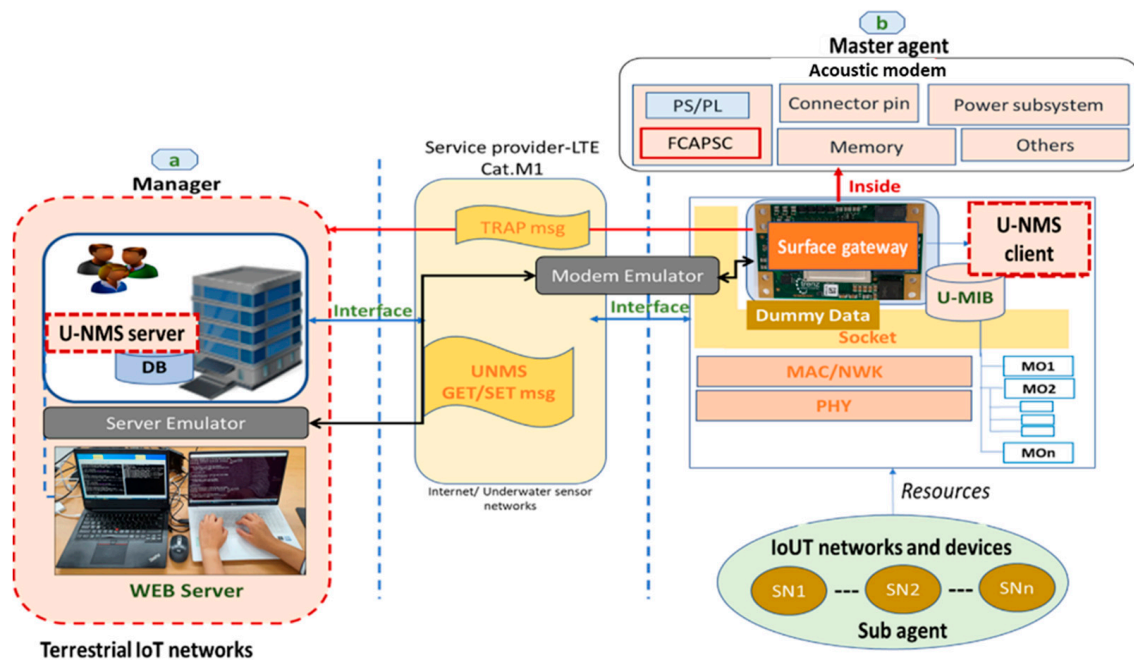


Figure 14. Fundamental prototype design for U-NMS.

Figure 15 shows the extended managed objects of U-NMS. In this paper, the MO structure is modified based on OMA-LwM2M. The newly defined underwater MO for U-NMS is Underwater network objects. These managed objects can be accessed using the address “Object ID/Object Instance ID/Resource ID/Resource Instance ID”. The description of defined MOs is summarized as follows:

- Network ID: Unique identity of underwater networks.
- Network status: Quality of underwater networks (0 indicates a weak connection, 1 indicates a normal connection, 2 indicates strong connection).
- Number of devices: Indicates the number of devices connected to underwater networks.
- Device ID: Unique identity of underwater devices.
- Device Name: Indicates the name of the device, such as sensors, gateway, UUVs, etc.
- Devices resource: Indicates the availability of the resources, such as battery, memory, temperature status, etc., of underwater devices.

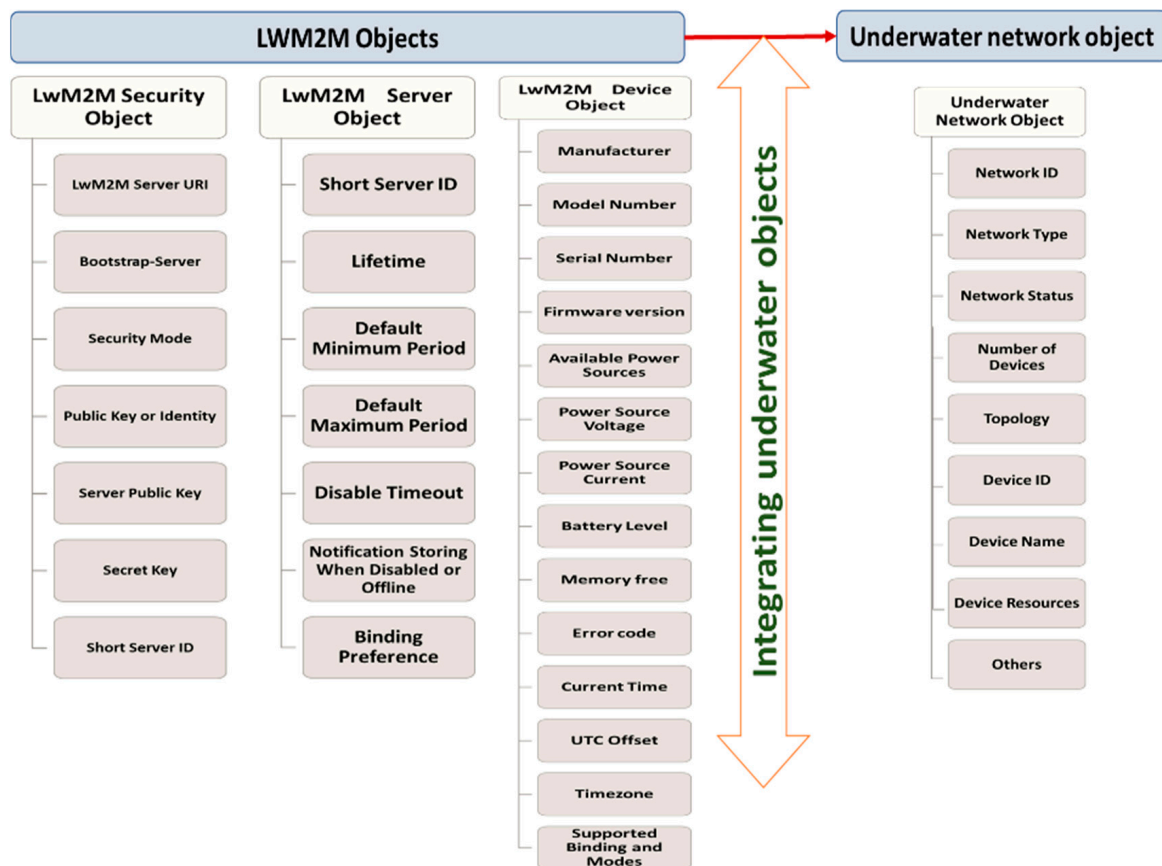


Figure 15. Defining underwater objects based on Open Mobile Alliance lightweight Machine to machine (OMA-LwM2M) in U-NMS.

3.4. Testbed Description

In this section, we describe the U-NMS testbed in the laboratory environment. We start with the hardware parts used, and then describe the protocol stack used in the U-NMS.

3.4.1. Hardware Description

1. **Acoustic modem:** The acoustic modem is designed with low power operation, which includes the MCU Cortex M3 processor, the operating frequency of 70 kHz, the maximum data rate of 200 bps, the maximum operational range of 50 m, and the dimension of radius 70 mm and height 40 mm. The serial peripheral interface (SPI) is used for the connection between the acoustic modem and the mainboard.
2. **The mainboard for U-NMS:** STM32F103CB is utilized as the mainboard for U-NMS, which comprises of MCU Cortex M3 processor, installed with U-NMS module over the firmware, width, and height of 30 mm respectively and power consumption of 3.3 V and 5 V. The universal asynchronous receiver/transmitter (UART) interface is used for the communication between the mainboard and PC.
3. **Acoustic Transducer:** The acoustic transducer supports the frequency range of 70 kHz, with the input power of 190 watts, an operating depth of 1500 m, and uses the cable type of polyurethane 6 mm low noise coaxial.
4. **Water tank:** The water tank is filled with water to create the underwater in lab environment, and the size is 75 cm × 75 cm × 100 cm—width, depth, and height, respectively.
5. **Manger/master agent:** The PC is utilized for monitoring and controlling the operations of the U-NMS system. In this U-NMS system, the modem is connected to the serial port to

GET-REQUEST, GET-RESPONSE, SET REQUEST, and, SET RESPONSE, from the manger and master agent.

Figure 16 shows the laboratory environment test of U-NMS. The U-NMS server and U-NMS client are developed using the acoustic modem, and the acoustic transducer connected with it. The manager is used to send the request and, the master agent is used to send the response back to the U-NMS server. The experimental laboratory results are shown in sub-clause 3.5.

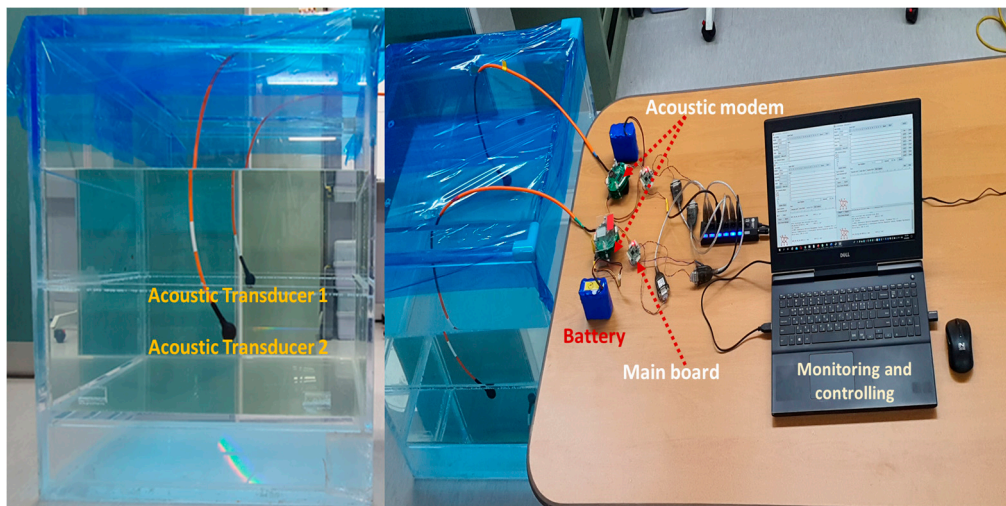


Figure 16. Testbed setup of U-NMS in a laboratory environment.

3.4.2. Protocol Stack of U-NMS

The protocol stack of U-NMS is divided into two layers: (1) lower layer and (2) upper layer as shown in Figure 17. The lower layer is the combination of U-NMS physical layer and U-NMS data link layer, and the U-NMS application layer is partitioned under the upper layer of U-NMS.

1. **Application layer for U-NMS:** The U-NMS application is installed inside the application layer of U-NMS. The U-NMS application module is employed here for the management of functions in U-NMS. The methods, such as GET REQUEST, GET RESPONSE, SET REQUEST, and SET RESPONSE, are used for transferring the U-NMS information.
2. **Data Link layer of U-NMS:** The mainboard of U-NMS is connected to the U-NMS data link layer. The U-NMS module is installed in this layer for transferring the information base on the request. The functions such as data processing, error detection, and handling, data framing, schedule management, etc. are used in the data link layer of U-NMS.
3. **Physical layer of U-NMS:** The functions, such as signal generation, binary transmission, acoustic medium for transmission, etc., are performed in the physical layer of U-NMS.

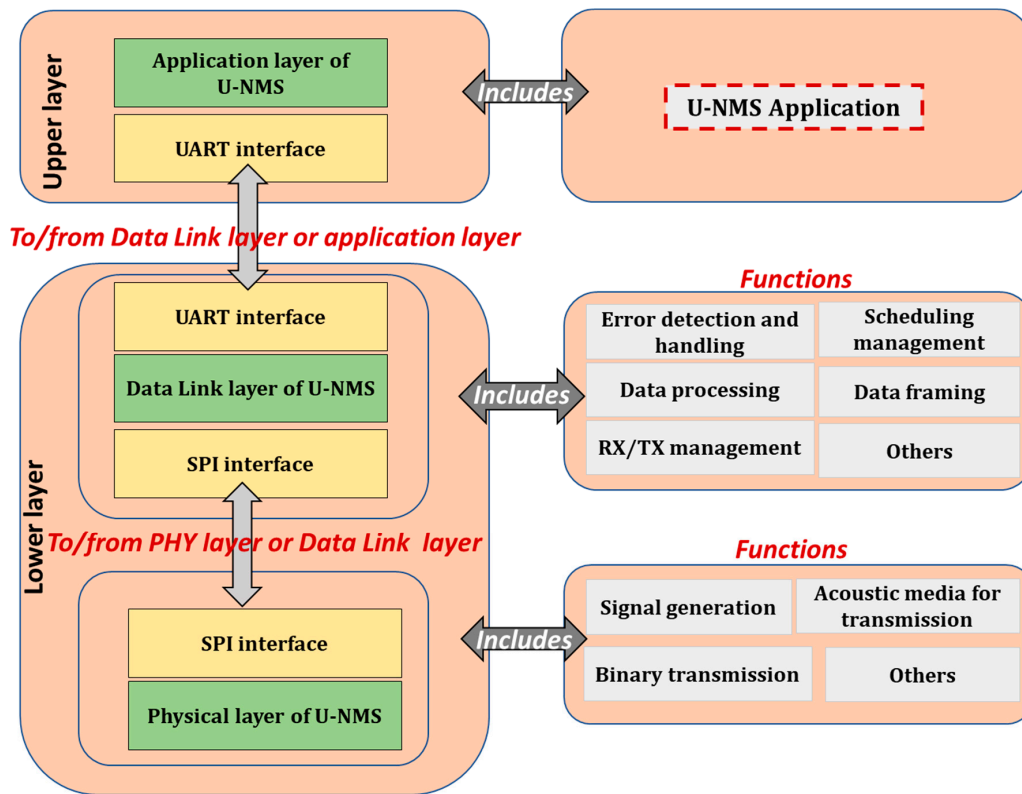


Figure 17. Protocol stack of U-NMS.

3.5. Laboratory Experiment Results

Figure 18 shows the laboratory experimental setup of the underwater network management system for processing GET and SET U-MIB objects in U-NMS. It consists of two parts: (1) manager (U-NMS server) and (2) master agent (U-NMS client). The manager requests the managed objects of underwater devices, such as network ID, device ID, device name, etc. The master agent receives the request from the manager and sends the values back to the manager. The commands used for GET_REQUEST, GET_RESPONSE, SET_REQUEST, and SET_RESPONSE are “91.92.11.XX.00”, “91.92.12.XX.00”, “91.92.13.XX.00”, “91.92.14.XX.00”, respectively, in U-NMS.

Figure 19 shows the packet format of U-NMS with, 5 bytes for sending and receiving data. The packet format consists of a Source ID, Destination ID, Packet Type, Resource Type, and Resource Value. The description of the packet format is summarized below.

- Source ID and Destination ID are used to identify the source and destination in U-NMS.
- Packet Types are the methods used in the U-NMS, such as GET_REQUEST: 0 × 11, GET_RESPONSE: 0 × 12, SET_REQUEST: 0 × 13, and SET_RESPONSE: 0 × 14.
- Resource Type is the various MO values used in the U-NMS, such as Network ID, Network status, Device ID, Device name, Device Resources, etc.
- Resource Value is the MO values used for GET_REQUEST/GET_RESPONSE and SET_REQUEST/SET_RESPONSE in U-NMS.

Figure 20 shows how the GET_REQUEST and GET_RESPONSE method works on the manager side and master agent side in U-NMS. The ‘a’ side of the figure shows the working of the U-NMS server. For example, the manager sends the GET_REQUEST value as “92 26241/21” and receives the GET_RESPONSE value as “91 26241/21/51”. Therefore, ‘51’ is the MO value received from the U-NMS client. The ‘b’ side of the figure shows the working of the master agent. In this case, the master agent receives the GET_REQUEST message from the manager, and transmits the GET_RESPONSE value, including the requested object.

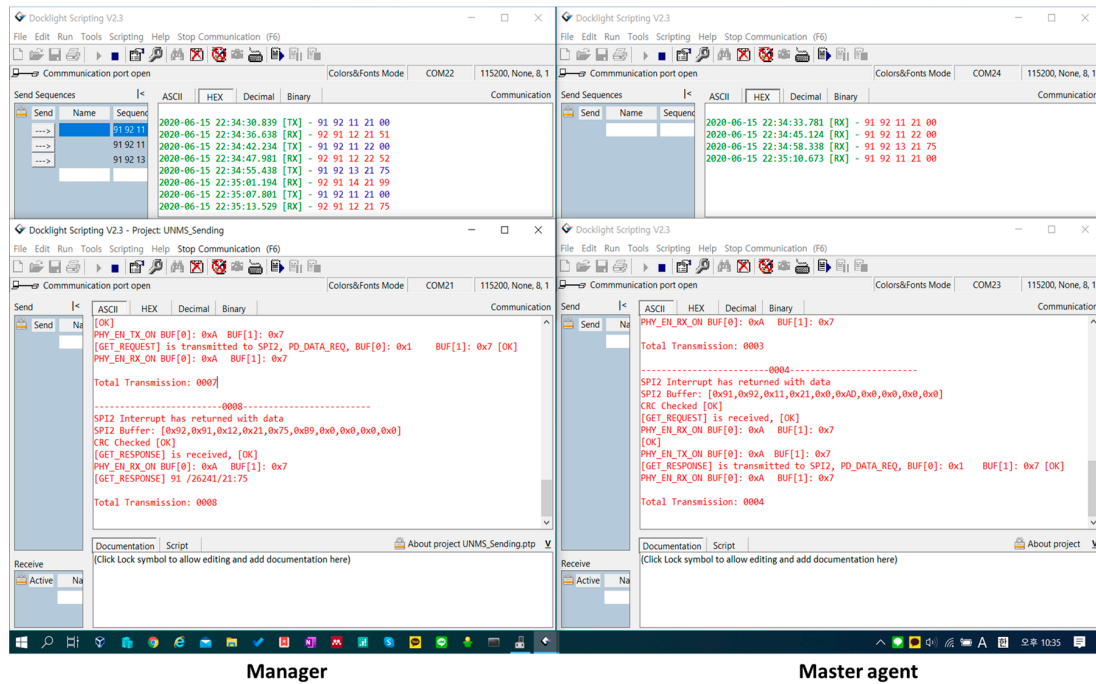


Figure 18. Laboratory experimental setup of U-NMS.

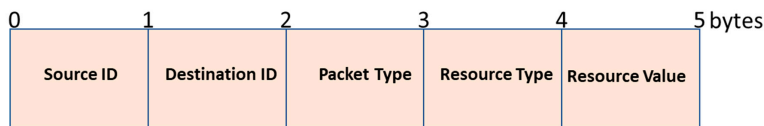


Figure 19. The packet format of U-NMS.

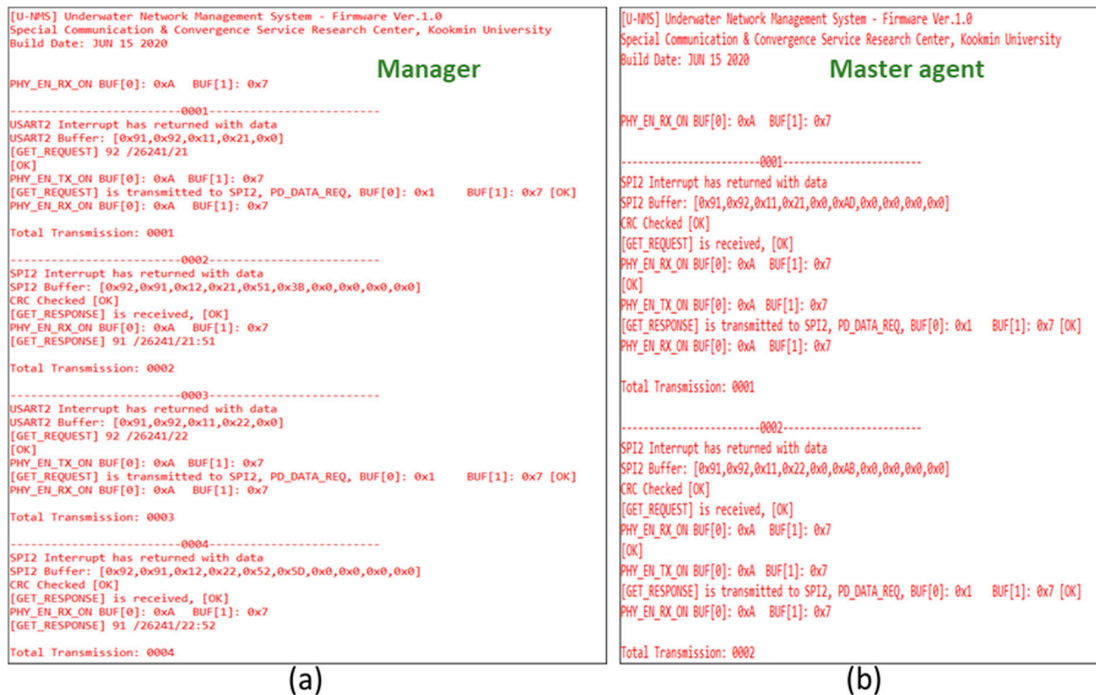


Figure 20. (a) Results of GET_REQUEST and GET_RESPONSE methods in the manager side of U-NMS. (b) Results of GET_REQUEST and GET_RESPONSE methods in the master agent side of U-NMS.

Figure 21 shows the working SET_REQUEST and SET_RESPONSE methods in the manager and master agent sides of U-NMS. The existing value of “92/26241/21” was ‘51’. The SET_REQUEST method is used to change the value of MOs in U-NMS. The ‘a’ side of the figure shows the working of SET_REQUEST and SET_RESPONSE methods in the manager side. In this case, the manager sends the SET_REQUEST method for changing the MO value from “91 26241/21/51” to “91 26241/21/75”, and received the SET_RESPONSE value successfully. The ‘b’ side of the figure shows the working of SET_REQUEST and SET_RESPONSE methods in the master agent side. In this case, the SET_REQUEST and SET_RESPONSE methods are received and transmitted successfully to the manager.

Manager	Master agent
<pre>-----0005----- USART2 Interrupt has returned with data USART2 Buffer: [0x91,0x92,0x13,0x21,0x75] [SET_REQUEST] 92 /26241/21 75 [OK] PHY_EN_TX_ON BUF[0]: 0xA BUF[1]: 0x7 [SET_REQUEST] is transmitted to SPI2, PD_DATA_REQ, BUF[0]: 0x1 BUF[1]: 0x7 [OK] PHY_EN_RX_ON BUF[0]: 0xA BUF[1]: 0x7 Total Transmission: 0005 -----0006----- SPI2 Interrupt has returned with data SPI2 Buffer: [0x92,0x91,0x14,0x21,0x99,0xB7,0x0,0x0,0x0,0x0] CRC Checked [OK] [SET_RESPONSE] is received, [OK] PHY_EN_RX_ON BUF[0]: 0xA BUF[1]: 0x7 [SET_RESPONSE] The value is successfully changed. Total Transmission: 0006 -----0007----- USART2 Interrupt has returned with data USART2 Buffer: [0x91,0x92,0x11,0x21,0x0] [GET_REQUEST] 92 /26241/21 [OK] PHY_EN_TX_ON BUF[0]: 0xA BUF[1]: 0x7 [GET_REQUEST] is transmitted to SPI2, PD_DATA_REQ, BUF[0]: 0x1 BUF[1]: 0x7 [OK] PHY_EN_RX_ON BUF[0]: 0xA BUF[1]: 0x7 Total Transmission: 0007 -----0008----- SPI2 Interrupt has returned with data SPI2 Buffer: [0x92,0x91,0x12,0x21,0x75,0xB9,0x0,0x0,0x0,0x0] CRC Checked [OK] [GET_RESPONSE] is received, [OK] PHY_EN_RX_ON BUF[0]: 0xA BUF[1]: 0x7 [GET_RESPONSE] 91 /26241/21:75 Total Transmission: 0008</pre>	<pre>-----0003----- SPI2 Interrupt has returned with data SPI2 Buffer: [0x91,0x92,0x13,0x21,0x75,0x8A,0x0,0x0,0x0,0x0] CRC Checked [OK] [SET_REQUEST] is received, [OK] PHY_EN_RX_ON BUF[0]: 0xA BUF[1]: 0x7 [OK] PHY_EN_TX_ON BUF[0]: 0xA BUF[1]: 0x7 [SET_RESPONSE] is transmitted to SPI2, PD_DATA_REQ, BUF[0]: 0x1 BUF[1]: 0x7 [OK] PHY_EN_RX_ON BUF[0]: 0xA BUF[1]: 0x7 Total Transmission: 0003</pre>

(b)

Figure 21. (a) Results of SET_REQUEST and SET_RESPONSE methods in the manager side of U-NMS. (b) Results of SET_REQUEST and SET_RESPONSE methods in the master agent side of U-NMS.

4. Conclusions and Future Direction

At present, the network management system (NMS) in IoT is one of the booming technologies used for the management of IoT devices in terrestrial IoT networks. The technologies applicable for the network management system in IoT, such as the network management protocol and device management protocols, cannot be applied in IoUT networks, due to various factors, such as the limitation of battery, memory, bandwidth, etc. Hence, it is essential to create a reliable NMS for IoUT. This paper describes the open challenges, benefits, and solutions for using a network management system, using acoustic communication technology in IoUT. Moreover, this paper provides various factors concerning U-NMS, such as (1) identify and outline open challenges in adopting NMS to U-NMS and illustrate the benefits of U-NMS; (2) comparative analysis of the network management and device management protocols of IoT along with the underwater network management system. (3) Requirements of an underwater network system in IoUT. (4) Lists necessary use cases of U-NMS and (5) propose a new prototype design and implementation setup of U-NMS in a laboratory environment for IoUT, using acoustic communication technology.

Many researchers have proposed various ideas related to the network management system in IoT, such as network management protocol, device management protocol, etc. However, by studying the limitation underwater, the researchers find difficulties in applying the IoT network management system to underwater. So, the U-NMS in IoUT has a huge space in the emerging world. Therefore,

the suggested prototype can be the basement for the developers in the underwater environment to develop the U-NMS. The current prototype is based on GET/SET methods, and only limited MOs are designed inside the U-MIB, such as network ID, device ID, network status, device resources, etc.

Currently, the proposed prototype is being tested in the laboratory environmental setup. In the future, the proposed prototype will be extended to the real field test in the sea environment, by employing different communication technologies in IoUT environment, such as acoustic, optical, and IR. Moreover, the management resources/managed object inside the U-MIB will be extended by adding more MOs. The method TRAP will be expanded in the future, to direct the notification message if any critical situation occurs in IoUT devices.

Author Contributions: Conceptualization, S.-H.P., D.R.K.M. and E.K.; Software, J.L. and S.-H.Y.; Methodology, D.R.K.M. and E.K.; Formal analysis and Visualization, S.-Y.S.; Investigation and Validation, J.-I.N.; Resources, project administration, Supervision and Proofreading, S.-H.P.; Writing—original draft, D.R.K.M.; Writing—review & editing, D.R.K.M. and J.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research is supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF), funded by the Ministry of Education (NRF-2019R1D1A1B03028903).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Qiu, T.; Chen, N.; Li, K.; Qiao, D.; Fu, Z. Heterogeneous ad hoc networks: Architectures, advances and challenges. *Ad Hoc Netw.* **2017**, *55*, 143–152. [[CrossRef](#)]
2. Chen, S.; Xu, H.; Liu, D.; Hu, B.; Wang, H. A vision of IoT: Applications, challenges, and opportunities with China perspective. *IEEE Internet Things J.* **2014**, *1*, 349–359. [[CrossRef](#)]
3. Zanella, A.; Bui, N.; Castellani, A.; Vangelista, L.; Zorzi, M. Internet of things for smart cities. *IEEE Internet Things J.* **2014**, *1*, 22–32. [[CrossRef](#)]
4. Chettri, L.; Bera, R. A Comprehensive survey on Internet of Things (IoT) toward 5G wireless systems. *IEEE Internet Things J.* **2020**, *7*, 16–32. [[CrossRef](#)]
5. ISO/IEC 10040:1998. *Recommendation X.701—Information Technology—Open Systems Interconnection—Systems Management Overview*; ITU-T: Geneva, Switzerland, 1997.
6. Jia, L.; Zhu, W.; Zhai, C.; Du, Y. Research on an integrated network management system. In Proceedings of the Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD 2007), Qingdao, China, 30 July 30–1 August 1 2007; pp. 311–316.
7. Dragomir, D.; Voinescu, A.; Draghici, A.; Tapus, N. WSN management in a multi-user secure context. In Proceedings of the 11th RoEduNet International Conference, Sinaia, Romania, 17–19 January 2013; pp. 1–4. [[CrossRef](#)]
8. Pras, A.; Schönwälder, J.; Burgess, M.; Festor, O.; Martinez Perez, G.; Stadler, R.; Stiller, B. Key research challenges in network management. *IEEE Commun. Mag.* **2007**, *45*, 104–110. [[CrossRef](#)]
9. Goers, W.; Brenner, M. Implementing a management system architecture framework. *Bell Labs Tech. J.* **2002**, *5*, 31–43. [[CrossRef](#)]
10. Bush, S.F.; Kalyanaraman, S. Management of active and programable networks. *J. Netw. Syst. Manag.* **2006**, *14*, 1–5. [[CrossRef](#)]
11. Rao, U.H. Challenges of Implementing Network Management Solution. *Int. J. Distrib. Parallel Syst.* **2011**, *2*, 67–76. [[CrossRef](#)]
12. Ma, Y.-W.; Chen, J.-L.; Huang, Y.-M.; Lee, M.-Y. An efficient management system for wireless sensor networks. *Sensors* **2010**, *10*, 11400–11413. [[CrossRef](#)]
13. Kurose, J.F.; Ross, K.W. *Computer Networking: A Top-Down Approach Featuring the Internet*, 6th ed.; Pearson: London, UK, 2010.
14. Silva, J.C.; Rodrigues, J.J.P.C.; Al-Muhtadi, J.; Rabêlo, R.A.L.; Furtado, V. Management platforms and protocols for internet of things: A survey. *Sensors* **2019**, *19*, 676. [[CrossRef](#)]
15. Case, J.; Fedor, M.; Schoffstall, M.; Davin, J. RFC 1157—Simple Network Management Protocol (SNMP). Network Working Group. May 1990. Available online: <https://tools.ietf.org/html/rfc1157/> (accessed on 1 May 2020).

16. Specialski, E.S. Management of Computer Networks and Telecommunications. Master's Thesis, University of Santa Catarina, Florianópolis, SC, USA, 2002.
17. SNMP Research. Simple Network Management Protocol. SNMP Research International. Available online: <http://www.snmp.com/protocol/> (accessed on 1 May 2020).
18. ASN.1 PROJECT. Introduction to ASN.1. ITU-T. Available online: <http://www.itu.int/en/ITU-T/asn1/Pages/introduction.aspx> (accessed on 1 May 2020).
19. Enns, R.; Bjorklund, M.; Schoenwaelder, J.; Bierman, A. RFC 6241—Network Configuration Protocol (NETCONF). Internet Engineering Task Force (IETF). June 2011. Available online: <https://tools.ietf.org/html/rfc6241> (accessed on 1 May 2020).
20. Bjorklund, M. RFC 6020—YANG—A Data Modeling Language for NETCONF. Internet Engineering Task Force (IETF). October 2010. Available online: <https://tools.ietf.org/html/rfc6020> (accessed on 1 May 2020).
21. Chappell, C. *White Paper Creating the Programmable Network: The Business Case for NETCONF/YANG in Network Devices*; Heavy Reading: New York, NY, USA, 2013.
22. Warrior, U.; Corporation, U.; Besaw, L. Hewlett-Packard RFC 1189—The Common Management Information Services and Protocols for the Internet. Network Working Group. October 1990. Available online: <https://tools.ietf.org/html/rfc1189> (accessed on 1 May 2020).
23. Ying, D.; Feng, Q.; Luoming, M. Implementation of a CMIP-based management interface for optical access network. In Proceedings of the Fifth Asia-Pacific Conference on Communications and Fourth Optoelectronics and Communications Conference on Communications, Beijing, China, 18–22 October 1999; Volume 1, pp. 87–90.
24. Sehgal, A.; Perelman, V.; Kuryla, S.; Schönwälder, J. Management of resource constrained devices in the internet of things. *IEEE Commun. Mag.* **2012**, *50*, 144–149. [[CrossRef](#)]
25. Kushalnagar, N.; Montenegro, G.; Schumacher, C. RFC 4919—6LoWPAN: Overview, Assumptions, Problem Statement and Goals; Network Working Group: Gatineau, QC, Canada, August 2007.
26. Mukhtar, H.; Kang-myoo, K.; Chaudhry, S.A.; Akbar, A.H.; Ki-hyung, K.; Yoo, S. LNMP—Management Architecture for IPv6 based low-power Wireless Personal Area Networks (6LoWPAN). In Proceedings of the Network Operations and Management Symposium, Salvador Da Bahia, Brazil, 7–11 April 2008; pp. 417–424.
27. Bierman, A.; Bjorklund, M.; Watsen, K. RFC 8040—RESTCONF Protocol. January 2017. Available online: <https://tools.ietf.org/html/rfc8040> (accessed on 1 May 2020).
28. Prieto, A.G.; Leung, A.; Rockwell, K. Automating the testing of RESTCONF agents. In Proceedings of the 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM), Ottawa, ON, Canada, 11–15 May 2015; pp. 984–989.
29. Pfa, B.; Davie, B. RFC 7047—The Open vSwitch Database Management Protocol. Internet Engineering Task Force (IETF). December 2013. Available online: <https://tools.ietf.org/html/rfc7047> (accessed on 1 May 2020).
30. Davie, B.; Koponen, T.; Pettit, J.; Pfa, B.; Casado, M.; Gude, N.; Petty, T. Public review for a database approach to SDN control plane design. *ACM SIGCOMM Comput. Commun. Rev.* **2017**, *47*, 15–26. [[CrossRef](#)]
31. Sharma, S.; Staessens, D.; Colle, D.; Palma, D.; Goncalves, J.; Figueiredo, R.; Morris, D.; Pickavet, M.; Demeester, P.; Sharma, S.; et al. Implementing quality of service for the software defined networking enabled future internet. In Proceedings of the 2014 Third European Workshop on Software Defined Networks, London, UK, 1–3 September 2014; pp. 49–54.
32. Klas, G.; Rodermund, F.; Shelby, Z.; Akhouri, S.; Höller, J. Lightweight M2M: Enabling device management and applications for the internet of things. In *Ericsson and Vodafone White Paper*; Ericsson: Stockholm, Sweden, 2014.
33. Ocak, M.C. Implementation of an Internet of Things Device Management Interface. Master's Thesis, School of Electrical Engineering, Alto University, Greater Helsinki, Finland, November 2014.
34. Rao, S.; Chendanda, D.; Deshpande, C.; Lakkundi, V. Implementing LWM2M in constrained IoT devices. In Proceedings of the IEEE ICWiSe 2015, Melaka, Malaysia, 24–26 August 2015.
35. Open Mobile Alliance (OMA). *OMA DM Device Description Framework*; Version 1.3; OMA: San Diego, CA, USA, October 2012.
36. Open Mobile Alliance (OMA). *OMA Device Management Tree and Description*; Version 1.3; OMA: San Diego, CA, USA, October 2012.
37. Open Mobile Alliance (OMA). *OMA Device Management Protocol*; Version 1.3; OMA: San Diego, CA, USA, May 2016.

38. Lamaazi, H.; Benamar, N.; Jara, A.; Ladid, L.; El Ouadghiri, D. Internet of thing and networks management: Lnmpp, snmp, coman protocols. In Proceedings of the First International Workshop on Wireless Networks and Mobile COMMunications (WINCOM 2013), Fes, Morocco, 25 December 2013; pp. 1–5.
39. Sheng, Z.; Wang, H.; Yin, C.; Hu, X.; Yang, S.; Leung, V.C. Lightweight management of resource-constrained sensor devices in internet of things. *IEEE Internet Things J.* **2015**, *2*, 402–411. [[CrossRef](#)]
40. Castro, M.; Jara, A.J.; Skarmeta, A.F. Enabling end-to-end CoAP based communications for the web of things. *J. Netw. Comput. Appl.* **2016**, *59*, 230–236. [[CrossRef](#)]
41. Dai, G. Design and implementation on SOAP-based things management protocol for internet of things. In Proceedings of the 10th World Congress Intelligent Control and Automation (WCICA), Beijing, China, 6–8 July 2012; pp. 4305–4308.
42. Oetiker, T. MRTG—The Multi Router Traffic Grapher. In *LISA 1998: Proceedings of the 12th USENIX Conference on System Administration*; USENIX Association: Berkeley, CA, USA, 1998; pp. 141–148.
43. Apel, U. Intelligent network management systems. In Proceedings of the X International Symposium on Subscriber Loops and Services, Vancouver, BC, Canada, 27 September–1 October 1993; pp. 257–264. [[CrossRef](#)]
44. Otani, T. An integrated management system architecture for utility telecommunication networks. Integrated Network Management VI. Distributed Management for the Networked Millennium. In Proceedings of the Sixth IFIP/IEEE International Symposium on Integrated Network Management. (Cat. No.99EX302), Boston, MA, USA, 24–28 May 1999; pp. 933–934.
45. Kim, J.-Y.; Lim, S.-D.; Hong, J.W.; Kim, S.-B.; Lee, H.-Y. TMN-based integrated network management using Web and CORBA. Integrated Network Management VI. Distributed Management for the Networked Millennium. In Proceedings of the Sixth IFIP/IEEE International Symposium on Integrated Network Management. (Cat. No.99EX302), Boston, MA, USA, 24–28 May 1999; pp. 947–948.
46. Zhang, T.; Xu, C.; Wu, M.; Zhen, Y.; Zhang, Q.; Wen, J. Implementation of Ad Hoc Network management system based on embedded ARM-Linux platform. In Proceedings of the 2010 International Conference on Networking and Digital Society, Wenzhou, China, 30–31 May 2010; pp. 167–170.
47. Tsatsoulis, C.; Soh, L.-K. Intelligent agents in telecommunication networks. In *Computational Intelligence in Telecommunications Networks*; CRC Press: Boca Raton, FL, USA, 2000.
48. Atzori, L.; Iera, A.; Morabito, G. The internet of things: A survey. *Comput. Netw.* **2010**, *54*, 2787–2805. [[CrossRef](#)]
49. Singh, K.J.; Kapoor, D.S. Create your own internet of things: A survey of IoT platforms. *IEEE Consum. Electron. Mag.* **2017**, *6*, 57–68. [[CrossRef](#)]
50. Urunov, K.; Shin, S.-Y.; Park, S.-H.; Lim, Y.K. Analysis of the network management system with constrained underwater devices. In Proceedings of the Symposium of the Korean Institute of Communications and Information Sciences, Seoul, Korea, 21–23 June 2017.
51. Urunov, K.; Shin, S.-Y.; Namgung, J.-I.; Park, S.-H. Designing of use cases for the U-NMS. *J. Inst. Electron. Eng. Korea* **2018**, *55*, 47–59.
52. Kao, C.-C.; Lin, Y.-S.; Wu, G.-D.; Huang, C.-J. A comprehensive study on the internet of underwater things: Applications, challenges, and channel models. *Sensors* **2017**, *17*, 1477. [[CrossRef](#)]
53. Duraibabu, D.B.; Leen, G.; Toal, D.; Newe, T.; Lewis, E.; Dooly, G. Underwater depth and temperature sensing based on fiber optic technology for marine and fresh water applications. *Sensors* **2017**, *17*, 1228. [[CrossRef](#)] [[PubMed](#)]
54. Mellin, T.A.; Ravik, O. Autonomous underwater system for pipeline leak detection and inspection. In Proceedings of the 6th International Symposium on Unmanned Untethered Submersible Technology, Durham, NH, USA, 12–14 June 1989; pp. 15–24.
55. Prasad, A.N.; Mamun, K.; Islam, F.; Haqva, H. Smart water quality monitoring system. In Proceedings of the 2nd Asia-Pacific World Congress on Computer Science and Engineering (APWC on CSE), Nadi, Fiji, 2–4 December 2015. [[CrossRef](#)]
56. Qin, Y.; Alam, A.U.; Pan, S.; Howlader, M.; Ghosh, R.; Hu, N.-X.; Jin, H.; Dong, S.; Chen, C.-H.; Deen, J. Integrated water quality monitoring system with pH, free chlorine, and temperature sensors. *Sens. Actuators B Chem.* **2018**, *255 Pt 1*, 781–790, ISSN 0925–4005. [[CrossRef](#)]

57. Khan, A.; Jenkins, L. Undersea wireless sensor network for ocean pollution prevention. In Proceedings of the Communication Systems Software and Middleware and Workshops, 2008. COMSWARE 2008. 3rd International Conference on, Bangalore, India, 6–10 January 2008. [CrossRef]
58. Delphin Raj, K.M.; Shin, S.-Y.; Namgung, J.-I.; Park, S.-H. The RIL based approach for predicting the growth of pearl spot fish using-UWAC. *J. Inst. Electron. Inf. Eng.* **2018**, *55*, 32–40.
59. Ainbridge, S.; Eggeling, D.; Page, G. Lessons from the field—two years of deploying operational wireless sensor networks on the great barrier reef. *Sensors* **2011**, *11*, 6842–6855. [CrossRef]
60. Davis, A.; Chang, H. Underwater wireless sensor networks. In Proceedings of the OCEANS 2012 MTS/IEEE: Harnessing the Power of the Ocean, Virginia Beach, VA, USA, 14–19 October 2012. [CrossRef]
61. Chuang, M.-C.; Hwang, J.N.; Ye, J.-H.; Yeh, W.-c.; Williams, K. Underwater fish tracking for moving cameras based on deformable multiple kernels. *IEEE Trans. Syst. Man Cybern. Syst.* **2016**, *47*, 1–11. [CrossRef]
62. Udo, E.; Isong, E. Flood monitoring and detection system using wireless sensor network. *Asian J. Comput. Inf. Syst.* **2014**, *1*, 2321–5658.
63. Kumar, P.; Kumar, P.; Priyadarshini, P. Underwater acoustic sensor network for early warning generation. In Proceedings of the 2012 Oceans, Hampton Roads, VA, USA, 14–19 October 2012; pp. 1–6. [CrossRef]
64. Wang, Y.; Dong, J.; Xie, F. A design of simple underwater wireless communication system. In Proceedings of the 2010 International Conference on Internet Technology and Applications, Wuhan, China, 20–22 August 2010; pp. 1–4. [CrossRef]
65. Rao, C.; Mukherjee, K.; Gupta, S.; Ray, A.; Phoha, S. Underwater mine detection using symbolic pattern analysis of sidescan sonar images. In Proceedings of the 2009 American Control Conference, St. Louis, MO, USA, 10–12 June 2009; pp. 5416–5421. [CrossRef]
66. KM, D.R.; Yum, S.-H.; Ko, E.; Shin, S.-Y.; Namgung, J.-I.; Park, S.-H. Multi-Media and Multi-Band based adaptation layer techniques for underwater sensor networks. *Appl. Sci.* **2019**, *9*, 3187. [CrossRef]
67. Sanap, M.; Chaudhari, S.; Vartak, C.; Chimurkar, P. HYDROBOT: An underwater surveillance swimming robot. In Proceedings of the 2018 International Conference on Communication information and Computing Technology (ICCICT), Mumbai, India, 2–3 February 2018; pp. 1–7. [CrossRef]
68. Bernardina, G.R.D.; Cerveri, P.; Barros, R.M.L.; Marins, J.C.B.; Silvatti, A.P. Action sport cameras as an instrument to perform a 3D underwater motion analysis. *PLoS ONE* **2016**, *11*, e0160490. [CrossRef]
69. Han, Y.; Zheng, C.; Sun, D. Accurate underwater localization using LBL positioning system. In Proceedings of the OCEANS 2015-MTS/IEEE Washington, Washington, DC, USA, 19–22 October 2015; pp. 1–4. [CrossRef]
70. Urunov, K.; Shin, S.-Y.; Namgung, J.-I.; Park, S.-H. High-Level architectural design of management system for the internet of underwater things. In Proceedings of the 2018 Tenth International Conference on Ubiquitous and Future Networks (ICUFN), Prague, Czech Republic, 3–6 July 2018; pp. 326–331.
71. Akylidiz, I.; Pompili, D.; Melodia, T. Challenges for efficient communication in underwater acoustic sensor networks. *ACM SIGBED Rev.* **2004**, *1*. [CrossRef]
72. Yang, H.; Alphones, A.; Xiong, Z.; Niyato, D.; Zhao, J.; Wu, K. Artificial Intelligence-Enabled Intelligent 6G Networks. *arXiv* **2019**, arXiv:1912.05744.
73. Kao, C.; Lin, Y.; Wu, G.; Huang, C. A study of applications, challenges, and channel models on the Internet of Underwater Things. In Proceedings of the 2017 International Conference on Applied System Innovation (ICASI), Sapporo, Japan, 13–17 May 2017; pp. 1375–1378.
74. Liou, E.; Kao, C.; Chang, C.; Lin, Y.; Huang, C. Internet of underwater things: Challenges and routing protocols. In Proceedings of the 2018 IEEE International Conference on Applied System Invention (ICASI), Chiba, Japan, 13–17 April 2018; pp. 1171–1174.
75. Domingo, M.C. An overview of the internet of underwater things. *J. Netw. Comput. Appl.* **2012**, *35*, 1879–1890. [CrossRef]
76. Available online: <https://www.linkedin.com/pulse/internet-things-iot-characteristics-kavyashree-g-c/> (accessed on 12 July 2020).
77. Patel, K.K.; Patel, S.M. Internet of things (IoT): Definition, characteristics, architecture, enabling technologies, application and future challenges. *Int. J. Eng. Sci. Comput.* **2016**, *6*, 6122–6131.
78. Available online: <https://designmind.frogdesign.com/2014/08/internet-things-six-key-characteristics/> (accessed on 12 July 2020).

79. Agarwal, K.; Rakesh, N. Node mobility issues in underwater wireless sensor network. In *Advances in Computer and Computational Sciences*; Springer: Berlin/Heidelberg, Germany, 2017.
80. Pallares, O.; Bouvet, P.; del Rio, J. TS-MUWSN: Time synchronization for mobile underwater sensor networks. *IEEE J. Ocean. Eng.* **2016**, *41*, 763–775. [[CrossRef](#)]
81. Bessios, A.G.; Caimi, F.M. Multipath compensation for underwater acoustic communication. In Proceedings of the OCEANS'94, Brest, France, 13–16 September 1994; Volume 1, pp. I/317–I/322. [[CrossRef](#)]
82. Zhang, B.; Hongyi, W.; Xu, T.; Zheng, L.; Yang, Q. Received signal strength-based underwater acoustic localization considering stratification effect. In Proceedings of the OCEANS 2016-Shanghai, Shanghai, China, 10–13 April 2016. [[CrossRef](#)]
83. Krishnan, K.; Gauni, S.; Manimegalai, C.T.; Malsawmdawngliana, V. Ambient noise analysis in underwater wireless communication using laser diode. *Opt. Laser Technol.* **2019**, *114*, 135–139, ISSN 0030-3992. [[CrossRef](#)]
84. Kantarci, M.E.; Mouftah, H.T.; Oktug, S.F. A survey of architectures and localization techniques for underwater acoustic sensor networks. *IEEE Commun. Surv. Tutor.* **2011**, *13*, 487–502. [[CrossRef](#)]
85. Yan, J.; Guo, D.; Luo, X.; Guan, X. AUV-Aided localization for underwater acoustic sensor networks with current field estimation. *IEEE Trans. Veh. Technol.* **2020**, 1–1. [[CrossRef](#)]
86. Yan, J.; Gong, Y.; Chen, C.; Luo, X.; Guan, X. AUV-Aided localization for internet of underwater things: A reinforcement learning-based method. *IEEE Internet Things J.* **2020**, 1–1. [[CrossRef](#)]
87. Urunov, K.; Shin, S.-Y.; Namgung, J.-I.; Park, S.-H. Lightweight constrained management for the underwater—network management system. *Sens. Lett.* **2018**, *16*, 698–711. [[CrossRef](#)]
88. Urunov, K.; Shin, S.-Y.; Namgung, J.-I.; Park, S.-H. Guidelines of u-MIB Design for IoUT. In Proceedings of the 2018 Korean Society of Communication Sciences Fall Conference, Seoul, Korea, 17 November 2018.
89. Urunov, K.; Shin, S.-Y.; Park, S.-H.; Yi, K. u-SNMP for the internet of underwater things. *Int. J. Control. Autom.* **2017**, *10*, 199–216. [[CrossRef](#)]
90. Urunov, K.; Shin, S.-Y.; Namgung, J.-I.; Park, S.-H. Key-Factors of the constrained management for the internet of underwater things. *Int. J. Internet Technol. Secur. Trans.* **2018**, *1*, 1. [[CrossRef](#)]
91. Urunov, K.; Shin, S.-Y.; Namgung, J.-I.; Park, S.-H. Underwater: Network management system on the internet of underwater things. In Proceedings of the Thirteenth ACM International Conference on Underwater Networks & Systems (WUWNet '18), Association for Computing Machinery, Shenzhen, China, 3–5 December 2018.
92. Kashiwara, S.; Sahara, T.; Kaneda, S.; Ohta, C. Rate adaptation mechanism with available data rate trimming and data rate information provision for V2I communications. *Mob. Inf. Syst.* **2019**, *2019*, 9. [[CrossRef](#)]
93. Moradi, M.; Rezazadeh, J.; Ismail, A.S. A reverse localization scheme for underwater acoustic sensor networks. *Sensors* **2012**, *12*, 4352–4380. [[CrossRef](#)]
94. Coutinho, R.W.L.; Boukerche, A. Topology control for internet of underwater things. In Proceedings of the 15th ACM International Symposium on QoS and Security for Wireless and Mobile Networks (Q2SWinet'19), Association for Computing Machinery, New York, NY, USA, 25–29 November 2019; pp. 79–83.
95. Ayaz, M.; Abdullah, A. Underwater wireless sensor networks: Routing issues and future challenges. In Proceedings of the MoMM'2009-The 7th International Conference on Advances in Mobile Computing and Multimedia, Kuala Lumpur, Malaysia, 14–16 December 2009; pp. 370–375. [[CrossRef](#)]
96. Jindal, H.; Saxena, S.; Singh, S. Challenges and issues in underwater acoustics sensor networks: A review. In Proceedings of the 2014 International Conference on Parallel, Distributed and Grid Computing, Solan, India, 11–13 December 2014; pp. 251–255. [[CrossRef](#)]
97. Wang, Y.; Zhang, H.; Sang, Z.; Xu, L.; Cao, C.; Gulliver, T.A. Modulation Classification of Underwater Communication with Deep Learning. *Comput. Intell. Neurosci.* **2019**, *2019*. [[CrossRef](#)] [[PubMed](#)]
98. Moniruzzaman, M.d.; Islam, S.; Bennamoun, M.; Lavery, P. Deep learning on underwater marine object detection: A survey. In Proceedings of the 18th International Conference, ACIVS 2017, Antwerp, Belgium, 18–21 September 2017; pp. 150–160. [[CrossRef](#)]
99. Jiang, R.; Sun, C.; Zhang, L.; Tang, X.; Wang, H.; Zhang, A. Deep learning aided signal detection for spad-based underwater optical wireless communications. *IEEE Access* **2020**, *8*, 20363–20374. [[CrossRef](#)]
100. Gao, M.; Foh, C.H.; Cai, J. On the selection of transmission range in underwater acoustic sensor networks. *Sensors* **2012**, *12*, 4715–4729. [[CrossRef](#)] [[PubMed](#)]

101. Qureshi, U.M.; Shaikh, F.K.; Aziz, Z.; Shah, S.M.Z.S.; Sheikh, A.A.; Felemban, E.; Qaisar, S.B. RF path and absorption loss estimation for underwater wireless sensor networks in different water environments. *Sensors* **2016**, *16*, 890. [[CrossRef](#)] [[PubMed](#)]
102. Alhumyani, H.; Ammar, R.; Albarakati, H.; Alharbi, A. Deployment strategies for underwater sensing and processing networks. In Proceedings of the 2016 IEEE Symposium on Computers and Communication (ISCC), Messina, Italy, 27–30 June 2016; pp. 358–363.
103. Yang, G.; Dai, L.; Wei, Z. Challenges, threats, security issues and new trends of underwater wireless sensor networks. *Sensors* **2018**, *18*, 3907. [[CrossRef](#)]



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).