International Journal of Control and Automation
Vol. 10, No. 10 (2017), pp.199-216
http//dx.doi.org/10.14257/ijca.2017.10.10.17

U-SNMP for the Internet of Underwater Things

Khamdamboy Urunov[1], Soo-Young Shin[2], Soo-Hyun Park[3] and Kwan Yi[4]

[1, 2] *Special Communication Research Center, Kookmin University Seoul, South Korea*
[3]*School of Computer Science, Kookmin University, Seoul, South Korea*
[4] *Department of Curriculum and Instruction, University of Eastern Kentucky, KY, USA*
[1]*hamdamboy.urunov@gmail.com,* [2]*sy-shin@kookmin.ac.kr,*
[3]*shpark21@kookmin.ac.kr,* [4]*kwan.Yi@eku.edu*

Abstract

Network Management System (NMS) is an application setting on the important role of a manager-console and dominates independent components of the network. The NMS is adjusting software and hardware into the unconstrained network. The system devices are using terrestrial area networks, and this based on possible a manager and an agent. In order to, a Simple Network Management Protocol (SNMP) is the major protocol of the network management and widely used NMS. In the real sector of the world which is using Management Information Technology, it relies on several factors. This means the quality of management system and they are saving money, time and increases productivity. Basically, growing number of IoT sensors and embedded devices are widely interconnecting each sector of social life. Hence, network constrained management system is not an easy using and enforcing every area. Indeed, legacy NMS system cannot integrate limitation environment, because it must become lightweight and easy adapt to a heterogeneous system. In our primary focus on also a constrained environment, like an underwater network. However, we should design and deploy for underwater application management protocol. It needs to become lightweight device management protocol for the IoUT. In this paper, our main goal relates to developing underwater management application protocol, like an underwater SNMP (u-SNMP). We also explain details of Managed Information Base (MIB) structure and registration IANA unique number for a private U-MIB. IoUT architectural model and implementation have the processing step on it.

Keywords*: SNMP, Structure Managed Information, MIB, u-SNMP, NMS, Internet of Things, Internet of Underwater Things*

1. Introduction

NMS is adjusting management objects and consequently, interconnect other system devices for retrieving data. It usually manages data a remote control of the network carry out central reporting to a system manager-console (see Figure 3). Indeed, the NMS [1] system wrapped several factor elements. Those are network devices discovery, device-management system, and intelligent notifications as so on [3]. Those mentioned elements depicted the Figure 1 categories of the NMS and their components. The main classification of NMS system relies on those steps. Primary core definition is devices discovery, this process of finding a device and synchronizing device inventory, configure with device management databases. Network device discovery can maintain Simple Network Management Protocol (SNMP) [2] and use for security SSH/SSHv2 [10]. **Network device monitoring** is a subset of the management system and this carries out

ISSN: 2005-4297 IJCA

through software application and tools. The major effort of the monitoring system may detect or overloaded, crashed-server, failing gateway, a composition of device status, and alert to the manager-console. **Network performance** endeavors to gather network statistics, metrics and define quality of services by the underlying network. In this case, network statistic components related to network bandwidth, capacity, network delay, data loss and network errors and other components. **Network performance analysis** is primarily delivering the quality of service to system-users. The important factor of a management system is intelligent notifications and this is including distributed network (IoT/IoUT). A purely distributed network is growing unstructured and so simple computation. The IoUT system should be horizontal integrate of services like the monitoring and intelligent notifications. The Figure 1 shown intelligent notifications and customizable alerts are closer to IoT technology. That means those are constrained environment network elements.

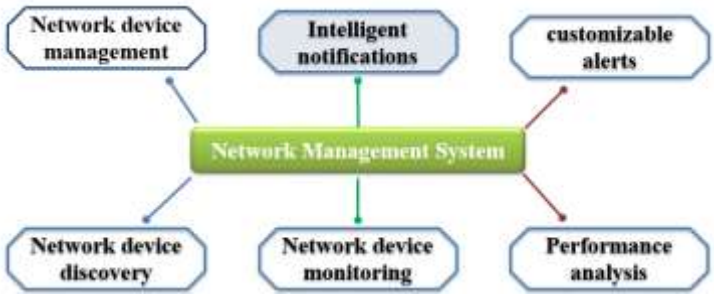


Figure 1. NMS Components

Additionally, Scientific's predicted near a decade, such as million or billions of embedded devices will coherence to the Internet. For instances, intelligent things, smart gear, smart energy, multimedia devices, home and building automation, sport-healthcare, thermometers, heart rate monitors, alarms and so on. This is way integrating embedded nodes, intelligence devices and other system instruments connect to together, it is called the Internet of Things (IoT) [5]. The underwater case, they will be connected heterogeneous underlying network and underwater sensor nodes, Autonomous Underwater Vehicles (AUV), cluster nodes and other possible connection, so it is called Internet of Underwater Things (IoUT) [4]. The underwater network or Internet of Underwater Things (IoUT) consist of an underline communication system. In that case, this underwater-system protocol stack and terrestrial network protocols are significantly different. Indeed, as a mentioned, management system need to integrate and cope with issues in the constrained network. Basically, the constrained underwater network need to a management system, this system will be lightweight and easy discover devices, services. However, there have several obstacles executing management system for limitation environment. They consist of several problem statements and requirement following steps:

- limited resources and CPU power
- it is not easy self-configuration and deploy
- constricted power-charge
- smaller memory footprint
- energy-efficient communication
- constrained hardware and software
- difficult to dynamic changing devices
- legacy protocol SNMP so heavyweight
- difficult compatibility of architectural design in the heterogeneous underwater network (acoustic, optical, fiber optic, wire)
- some applications based on time-synchronizer (manager and agent)

Furthermore, the SNMP should enable to terrestrial protocol, this cannot directly implement the underwater network community as well. This protocol is not only limited to a physical instrument, but also software like web server and device management databases. Moreover, one of underwater network component is a surface (see Figure 15). The surface devices are attaching terrestrial network elements like moving gateways, buoy, and proxy agent, and DMD, servers as so on and easily deploy manager software directly, based on SNMP without modifications. However, manager SNMP methods need to integrate underwater agents and using interface also need to lightweight and interoperable to the system. Our main goal is how to apply SNMP protocol to IoT/ IoUT devices, and a manager is terrestrial area network devices, but an agent can be constrained devices. Most of the essential base component obtain those underwater sensor devices acting a legible role in the IoUT.

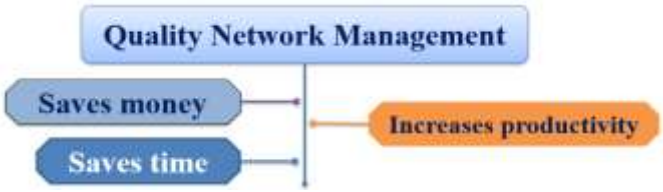


Figure 2. Quality Network management

Essential parameters have quality network management in depicted Figure 2. Especially saves money is manager console is required at a single location to a managed accessing network system, and struggle to decrease hiring expenditures. Save time is the major role of the management system, manager console or team members can simply enter using data own management station. Finally, increase productivity is helping people every aspect of hardware and software. Our primary focus on making a u-SNMP new solution protocol. In that case, our paper should analyze all possibility of application protocol as a u-SNMP and also designing details of MIB hierarchic tree which means our private MIB (U-MIB). Additionally, u-SNMP stand for two means: lowercase character of the (underwater - u) is a lightweight device management. The next one is constrained environment protocol of system networks. As much as possible we strive to describe following steps integration of private U-MIB to the standard-MIB hierarchic tree. Moreover, we should investigate objects like domination devices, those objects have identity possibility value and it is called Object ID, or OID. We should design IoT manager and agent architecture, attempt to the implementation of the underwater management system.

2. Related Works

Primary investigation solution is concerning usability of the NMS. In small networks, such as an office or home network is not necessary the NMS. However, it is important for any large network, such as those in an enterprise configuration. For instance, server farms, data center, and corporate networks, bank, central control elector power as so on. Additionally, the NMS system is engaged IoT world also. The IoT and IoUT are widely dependent on intelligent sensor nodes. The main role of the intelligent sensor is detecting responsible of the position, light, pressure, and temperature. In that case, we should investigate several research solutions rely on SNMP and IoT integration. During exploration, we obtain several key points of the SNMP and implementation process. The initial step is how to apply SNMP protocol for the IoT system [6]. The SNMP also implement agent part to the constrained environment [7]. The SNMP protocol is using virtual machine/virtual interface for constricted environment [11]. They are using Kernel and Xen managing Domain-Table contains Virtual Machine's data. Most of the limitation

sensors are using embedded OS and implementing lightweight SNMP Agent. This [12] is analyzing and obtaining embedded SNMP agent results. The closer result is an attempt to investigate the implementation of embedded SNMP agent [8]. In our exploring achievement also related to an embedded system, as a Linux OS or embedded application design using a Real time OS (RTOS) so porting lightweight SNMP (U-SNMP). Much quantity of scalable sensors and actuators are using SNMP-based protocol [13]. Hence, the SNMP is one of the most common use monitoring systems. Next one is implemented and deploying management system constrained devices in the IoT. Truly, design and analysis embedded system for the [5], SNMP agent. Indeed, research result is sensor node using IoT and monitoring based on SNMP [9]. Implementation for real testbed in a constricted network. IF such kind of sensor devices connect discovery-algorithm way to IoT also possible methods [14]. It is assumed that we just using monitoring IoT enabling devices [15]. Moreover, the entities managed with some system which is important to secure and protect the network as well. The previous version of SNMP v1/v2 is the biggest weakness of the security. Those SNMP versions cannot provide encryption data into the connection manager and agent. Some of the users who are not related management system like eavesdrop on management information, they can easily get data from the system without any authentication permission. So how to tackle this issue, one of the factor is SNMPv3. That version SNMPv3 is more secure and encrypted data in the system, not only managed. Likewise, create SNMP client in Java using SNMP4j. Actually, there are lots of [16] open source library for SNMP is available, even java have the library for the same.

3. Analysis of Lightweight Device Management U-SNMP Protocol

Legacy SNMP protocol is heavyweight to deploy underwater environment because we need to make a lightweight U-SNMP protocol. It is a UDP-based network protocol. The initial step, we should explain SNMP based manager and agent model. Next step is MIB structure based on SMI, how to create new-MIB. There is core advantage exploring the various type of the agent and manager. The SNMP manager brilliant way of the manger-console and agent interfaces. Hence, different level of the hardware devices like a server, router, proxy agent, sensor nodes, gateway, and buoy connect to the network. Basically, the agent is one of the main management system element. In this paper, more focus on proxy-agent interconnection to the IoT devices, constrained nodes. The interesting fact is why becoming u-SNMP protocol and MIB hierarchic tree registration process of IANA. We also explain whole of the arising questions following categories.

3.1. Manager and Agent Considerations

Basically the manager and the agent structure rely on core element of the management system hardware and software usability. The SNMP database of management information (MIB) is wrapping of managed object data. Each managed object need to be unique and reliable identity number. The Figure 3 illustrated Manager and Agent architecture based on SNMP protocol.

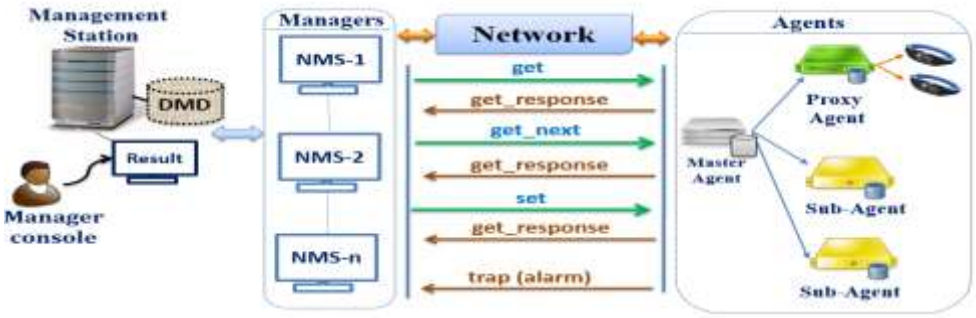


Figure 3. Manager and Agent architecture

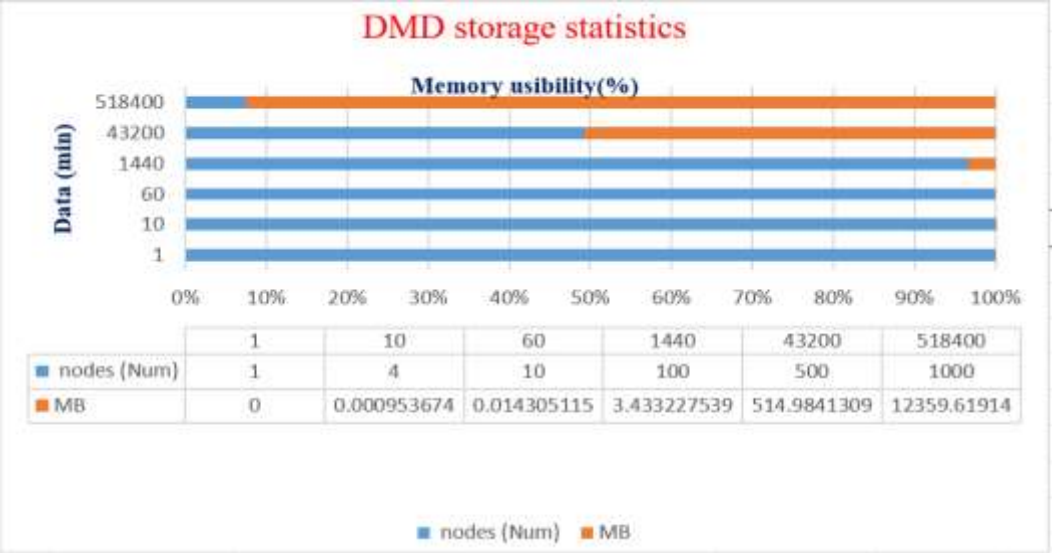
A manager-console is a like administrator and responsible whole of the system. Responsibility factors are developing network system topology and central data collection from system, fixed on problem sector. If a manager – console attempts to adjust retrieve process of the data (CPU, hard disc level, energy, cold start, warm start). Most of information are coming central based to the manager-console from mange devices, most of data aggregate and storage in device management database (DMD) which getting statistical analysis and status of devices implementation. In that case, Figure 3 depicted several NMS system (managers) interconnection each other and using notification method based on SNMP protocol. NMS (Manager) is capable of querying any managed device – via polling and decision in Network. Normally runs on very few systems compared to SNMP agents. Truly, DMD can regularly gather storage-data as well and following those steps:

1. Management facilities should recognize pure data collection, minimal data come from underwater sensor devices.
2. Hardware and software can relatively requirement of data collection.
3. Let's look at the assumption that 500 or 1000 nodes in the management system, additionally each device status value is 25 Byte data per minute. In this calculation 14 GB in a year.
4. Underwater sensor board relies on beagle-board black or Raspberry PI, using RTOS or embedded Linux (Ubuntu) in IoUT system. For calculation value Table 1 and Figure 4.

Table 1. Analysis of DMD Data Storage Statistics

TIME	Date (min)	Nodes (Number)	Size(byte)	KB	MB	GB
minute	1	1	25	0	0	0
minute	10	4	1000	0.9765625	0.0009537	0
hour	60	10	15000	14.6484375	0.0143051	1.39698E-05
day	1440	100	3600000	3515.625	3.4332275	0.003352761
month	43200	500	540000000	527343.75	514.98413	0.50291419
year	518400	1000	12960000000	12656250	12359.619	12.06994057

That's storing on extravagant to the DMD, however we should make a little different scenario for data aggregating into the DMD.



	1	10	60	1440	43200	518400
nodes (Num)	1	4	10	100	500	1000
MB	0	0.000953674	0.014305115	3.433227539	514.9841309	12359.61914

Figure 4. Analysis of Storage Statistics

We may predict those statistical facts:

1. Gathering data every minute is difficult and excessive. We should regulate the manager and the agent 10 or 15 minutes should do. Certainly, 7 or 10 GB will store in a year.
2. Number of 1000 nodes are not big network for management system, it is like normal.

However, redundant data also stored the DMD. Instead of nodes we can collect data from interface that 5 times less than nodes data.

3. High potentially performance for manager is nice, at least a SSD 250 GB memory.
4. Deploying agent software into Beagle Bone Black device specific requirement.

In that case, we should define more details information about agents and their structures.

Agent:

a. Small piece of code that runs on every SNMP managed device and gather and sends data about managed resource in response to a request from the manager.
b. Collect information from network device, on which it resides and stores in MIB.
c. Can initiate communication with SNMP manager using traps.

Proxy-agent:

d. A Proxy agent is an SNMP agent that maintains information of one or more non-SNMP devices
e. Proxy Agent does the conversion of control messages.
f. The management system can support Management Object (MO), the legacy of the system elements rely on manager and agents which are interconneting to the network.
g. Managed Object may support addressing that can identify each Management Devices.

Master-agent is an entity on adjusting sensor nodes that exchanges SNMP messages with management application, such WEBNMS [16]. That is initial interface between NMS and sub-agent or agent. Interesting how to create, all the SNMP agents using agent

complier can act as both Master-agent and sub-agent. Additionally, proxy-agent can integrated constrained environment devices and interconnecting IoT/IoUT sensor nodes. The right side of Figure 3 has shown this IoT sensor nodes. As before mentioned, limitation devices can integration service and devices discovery that part, especially agent part process. Most important thing is all data structure and localization information in MIB. We should analysis and design our private-MIB. Moreover, we should discuss about communication methods. In this example: GET, SET, and GET-NEXT, TRAP. How agent knows about NMS getting request especially that sensor node. How to sensor devices recognize and generated this request. They are using variable binding (**VarBind**). The Figure 5 shown structure of Variable binding and all devices has own variable-binding consists of two elements: ***Object identifier*** *(OID)* and ***Object value***.

VarBind = OID + Object value

This variable binding is attaching GET method. GET request the value of one or more managed objects within the request the manager specifies the identifier of the objects the manager is interested in.

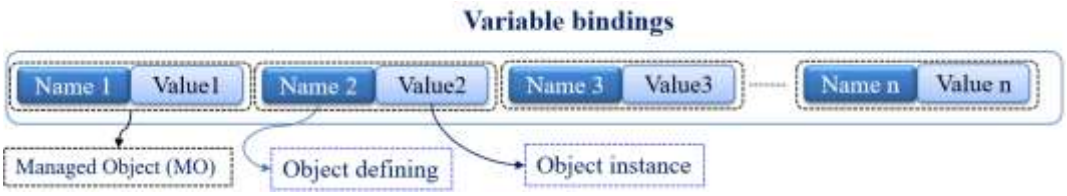


Figure 5. Variable Binding

After GET request the agent determines the value of the tying object and send this value together. OID can back to the NMS (manager) the combination of object identifier an object value is called a variable binding. The GET method only using read only value. In that method cannot modify any value of the agent. Manager only create GET request regularly. After that getting acknowledgment of get-response. More widely valuable method is **GET-NEXT**. The NMS (Manager) wants to inspect or browse which map object are supported by a certain agent. Those value included table, this entries can be indexed by identifiers that change in hunting such IP addresses. In case, routing tables the NMS (manager) cannot such indexes in it forms get PDU is not usable again get next handle these cases. Protocol Data Unit (PDU) is used SNMP manager and agent communication between. NMS (Manager) wants to modify the value of the triangle object since communication manager and agent is based on UDP. Unreliable the agent will always send back a response far bind included signal the request was successful or field. SET method cannot only be used to modify objects value but may also trigger certain actions like disabling interface or rebooting the device instead of modifying. Those static objects the set can also be used to instantiate new table the addition of new role. The table is relatively complex operation which involves usage of whole status note low status textual convention explained the SMI. The SET operation method is atomic either all objects are modified unknown although this sounds like a nice property implementing atomic operations is very hard especially if extensible agent technology is used an important addition that was introduced with SNMPv.2 is the descriptions of the set operation with two phase commit

1. Check – if all objects specified in the request can indeed be modified
2. Enter and at your modifications are performed if check within phase 1 are passed roll back.

Here presented Trap method. The Agent sends the request to NMS (manager) about emergency status (system security, power value, storage usability). It should be noted that reception is not confirmed by the manager the trap mechanism is therefore unreliable and

the manager necessary should continue polling the agents to check if special events are killed. Agents may be configured such that no traps will be transmitted or the traps will only be transmitted to specific managers since traps were quite limited the SNMP v.2 standard completely redefined the operation of the trap you to make it more flexible possible events were no longer defined as part of the protocol operations but could now be defined as part of but modules the SMIv2 notification type macro. Trap usage is more flexible. SNMP v2 also encoding of the trap has changed is now the same as the PDU's. This means that software can be simplified since manager does not acknowledge reception of the trap PDU and SNMP uses the unreliable UDP transport to sense its PDU, the agent cannot be sure if the trap was received by the manager. However, improving quality of security and other facility SNMP protocol. Updating regularly version and adding more detail information about methods. There is GET- Bulk method. The Get Bulk is using SNMP v.3, it allows us to retrieve more than one next object in a single interaction. In this example, we show a get requesting the next three objects the request identifier. The first object and specifies three for the value of max repetitions parameter after reception of the request the agent identifies. Those the values and identifiers of the next three objects and sends this back in the response to the manager. That means the max repetition parameter geld book cannot only behave as a series of consecutive. The GET-NEXT within additional parameters called non-repeaters can also make a book to act as a single get next operation the value of no repeaters indicates how many far binds in the request PDU will be treated as a get next value. Finally, Inform method is using SNMP v.2 threat PDU is able to send to notifications to manager however opposed to the trap video after the reception.

Table 2. SNMP Methods

Methods	Definition
Get-Req	Static OIDs are known
Get-Next-Req	Tables and MIB discovery
Get-Bulk-Req	SNMPv2 and very efficient
Set-Req	To change objects or add table elements
Trap –Req	Notify the manager of special events
Inform-Req	SNMPv2 and confirmed trap
Report – Req	Some type of error between managers.
U-SNMP methods	
U-Get-Req	Getting a data from an agent, and using static OIDs
U-Set-Req	Modifying value of Object Instance & using different value
U-Trap-Req	Status of error or emergency case of the agent.

Method of inform enabler can support manager sends an acknowledgment back to the agent in forms of response PDU. That mains difference of a trap and inform method. Additionally, Table 2 solution of SNMP methods. U-SNMP protocol method relies on get, set and trap methods for constrained IoT environment. Moreover, it can be usefully extended methods possible nodes interconnection.

3.2. Underwater - Management Information Base (U-MIB)

MIB is the gathering of network data and contains the real values of Managed Objects (MO) in the agent in the form of variables, tables of variables. MIB – represent characteristics of a managed device and changes are done in agent MIB.

3.2.1. Structure of Management Information (SMI)

The SMI makes the definition of new-MIB easier. The SMI helps to MIB designers and defines the syntax. This also allows a tool to build MIB structure. Essentially, most of the important things are understand MIB, read and learn how to read MIBs. SMI requires that each MO have the unique name. In this case, we can choose the number of ID source and destination address in Table 2 for interconnection.

Table 3. Scalar and Table

Source ID	Destination ID
19.12.22.3	201.22.23.23
32.3.3.36	699.6.3.56

Management information within managed systems must be represented as:

a. SCALARS is the number of packet received like current time. This compatible such kind of a variable type like Integer, Character, and Octet string as so on (you can see Table-3).

b. TABLES are two-dimensional arrays of scalars. This table creates a structured of scalars. That structure only limited definition to the table, they cannot define C programming language (you can see Table-3).

In that paper, we may use several terminologies which is concerning an object, object ID and Object Instance. This is defining follow steps:

a. Object is the definition and becoming domination element in Management system.
b. Object instance has a value.
c. Object ID is the uniqueness of the system elements.

As before mentioned Table-3 defined several parameters, for instance Object instance is a value of 19.12.22.3. It is dynamically size be change. Question arise how should the manager retrieve elements within the table? How should manger do that? It is not possible to transfer the entire table as an entity. The manager should transfer table elements one by one.

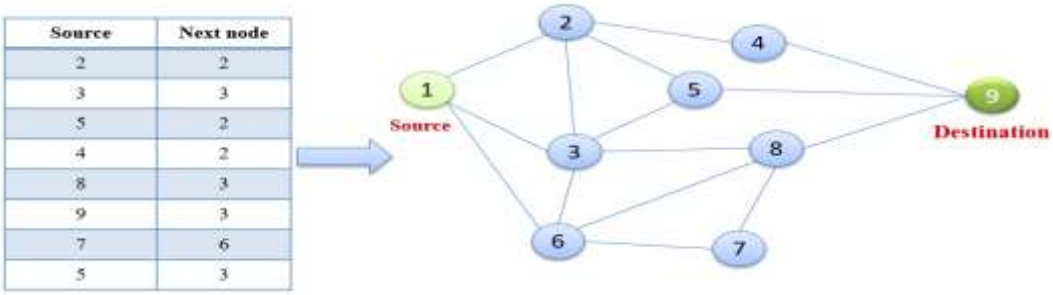


Figure 6. IoUT Nodes Routing Table

Figure 5 is routing table and it has two columns (destination and next node). We can define the process of manager and agent individual working elements. This is row 4 and next node is 2. This can find next node based on short way method or finding fix element way. That depends on the design of system developer. To identify which instance is needed, put the instance number behind the object. Caching is not possible since Intermediate rows may be deleted or added. Moreover, deeply declare IoUT and MIB tree structure. In that case, exactly heterogeneous network example (terrestrial and

underwater). The Figure 6 depicted underwater network part has Sensor node if you can see only a node but three managed object.

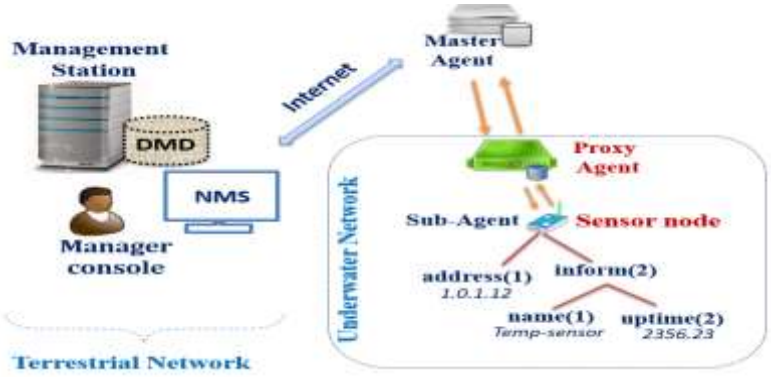


Figure 7. IoUT and MIB Structure Example

NMS and Master-Agent system can the manager identify like the address (1), name (1), and uptime (2). Those variables called Object Instances. The Object is the definition and object instance has a value. How to the recognize address (1), this is 1.1. If we want to the real value of the instance, that can use GET the value of the instance use 1.1.0. Use the Get 1.1 give to error. The Important fact is adding 0 ends of tree structure elements and using value of MO. Get 1.1.0 getting 1.0.1.12. One more example, GET 1.2.0 there is no value error. There is no instance. Instead of 1.2.2.0 one may use 1 info uptime 0. Additionally, this Figure 6 used to Proxy-Agent for sending SNMP request and integrating U-SNMP retrieve response to the NMS.

3.2.2. Management Information Base

MIB has new types SMIv1 and is not manager status and the Figure 8 shown MIB tree structure. Furthermore, three categories MIB allocated, they consist of Standard and Enterprise MIB already defined MIB library like OiDVIEW [21]. Currently, most of the registered Unique number of private MIB already uploaded there, but now no all. That means most of organization need to upload own private MIB resource. In our case, U-MIB will be upload to standard MIB. When we want to use SNMP management protocol in that time we should integrate standard MIB library.

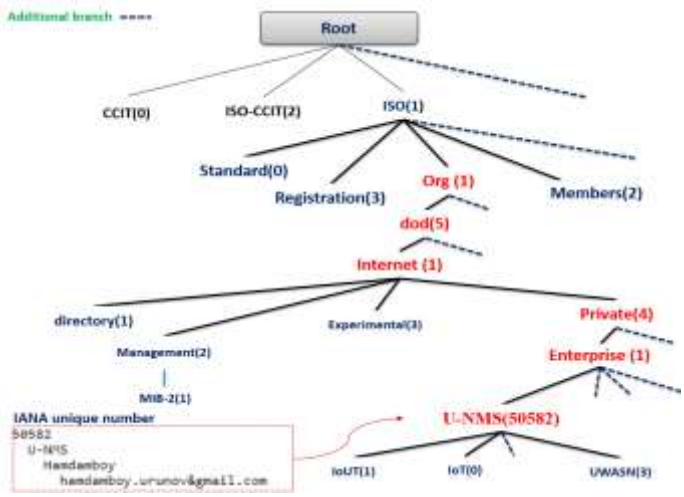


Figure 8. MIB Hierarchy Tree (U-NMS Registered)

However, if we would like to join to our private MIB that is different case of the MIB usability. Moreover, our main focus is U-NMS (**50582**) number exactly on it. In our research center already register IANA [20] registration. We have getting unique number for private MIB. Every organization or institute strive to attach own private MIB. It is normal case but little complex. We want to retrieve value of IoUT (1), it should be follow this way 1.1.5.1.4.1.50582.1 or human readable status (Private (4). Enterprise (1) as so on). Interestingly we are using name object, it is managed object instances.

- Define the data type allowed in MIB
- Define naming structure for each managed objects
- Language syntax is ASN.1 standard and notation elements in SMI.
- ASN.1 functionality is defining name objects and data storage in the object.

The Internet's SMI is defined in term of ASN.1 constructs. That is inherited from OSI roots of SNMP. ASN.1 was probably not the best choice. Simple types: capital letters type group SIMPLE TYPES, this is INTEGER, OCTET STRING, OBJECT INDETIFIER. This already define ASN.1standard. APPLICATION-WIDE TYPES: based on the Simple type and defined by SMI. This is initial capital case. PSEUDO TYPES: is related to BITS. We should define type elements. INTEGER and Integer32. This types totally same 32 bits. Unsigned32 also 32 bits. There are also **Gauge32** counts from 0 to max integer.

Example of U-MIB defining: Object type definition

```
---- Underwater sensor node U-MIB defining
Route-Table OBJECT-TYPE
SYNTAX IpAddress
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"This would be inspected by an agent on receipt of a U-MIB value of trap and inform."
 ::= {U-MIB}

Route-Entry OBJECT-TYPE
SYNTAX Object ID
MAX-ACCESS read-only
STATUS current
DESCRIPTION "A route to a particular destination"
INDEX {destination, next node}
 ::= {route Table }
```

Figure 9. Defining of U-MIB

We defined above mentioned example Figure 9 concerning Figure 8 and Figure 7 related information, SYNTAX value is using addressing of unique, like IP address and Object ID. This means, there may be multiple rows of type Route-Table and Route Entry. The Route Entry is Object-type and there needed for syntactical reasons for defines a table row. Entry of the one row in the table. We have to operate on the individual row elements. INDEX value is new one two elements. MAX-ACCESS value is not-accessible, because of read-only. That mean using GET method. We cannot operate on table directly as a whole. Read and write should be on individual elements.

3.3. Management Lightweight U-SNMP Protocol

Message format and PDU value major element of any protocols. So our main goal modification way to find SNMP protocol and developing U-SNMP protocol. In this part of the paper, we should discuss SNMP message and developing U-SNMP exchanging message structure. This protocol does not send only a PDU value, this wrapped PDU, message, and version of the protocol. Most of SNMP protocol message consists of several parts (variable binding, Error process, Request ID, type of message). Our suggestion U-

SNMP protocol also based on this message format. However, variable binding parameters limitation value case. Every time using the different value of message length, so usually used Basic Encoding Rules (BER). Encoding and decoding Object ID value including BER. More defining BER is functionality of OID using format Integer 32, Counter64 and so on. U-NMS (50582) hierarchic tree (1.1.5.1.4.1.50582) in Figure 8 is using encoding process that we should encode the first 2 integers in the OID. Indeed, they are using different versions (2/3) also. Variable bindings consist of 2 different value Name and Value. The Figure 10 Message format combination structure. Growing number of Managed object and value of length PDU are not good to constrained network. PDU header composed of message ID, size of the message, flag of message, such as security. Sending and receiving message has the unique number on it. It is called Request ID and that will be unique. Size of the message is so important for limitation resource always adjusting message size value and keep to algorithmic way storage reply without redundant data (different point of u-SNMP). Message Flag also wrapped type of message like octet string, and security and authentication. SNMP v.3 struggle to the more secure management system. If possible u-SNMP also becoming secure and lightweight. All possibility of encryption and decryption are following BER for PDU.

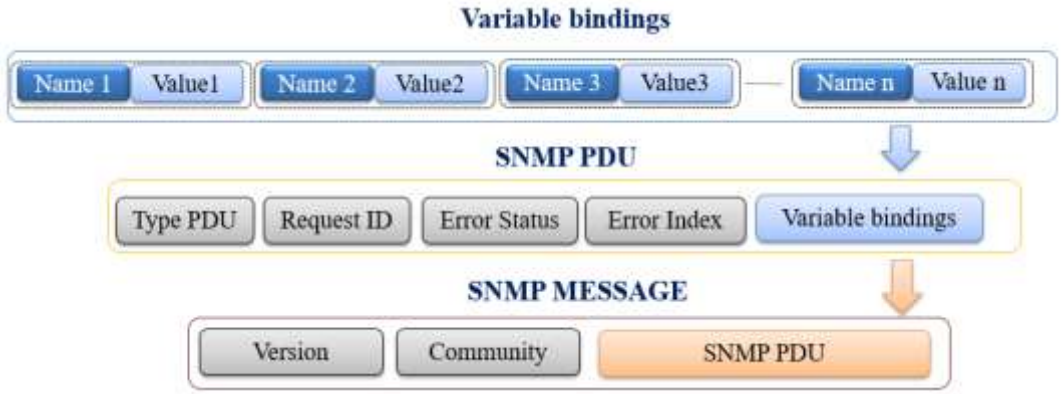


Figure 10. VarBind + SNMP PDU = SNMP Message Format

Figure 10 depicted SNMP PDU plus Variable binding and result are Message format of the SNMP. As mentioned VarBind is MO elements for managed system. Likewise, more information of PDU type value and definition include (Figure 11). PDU status value is important (0 until 5) if the value is 0 submission and retrieving successfully. When we get PDU value is 5, which is any kind of error (transmission problem, signal weakness, or receiver capturing issue). Additionally, PDU value is 3, which means Bad-Value accepted. In that case, the process of DMD storage is invalid.

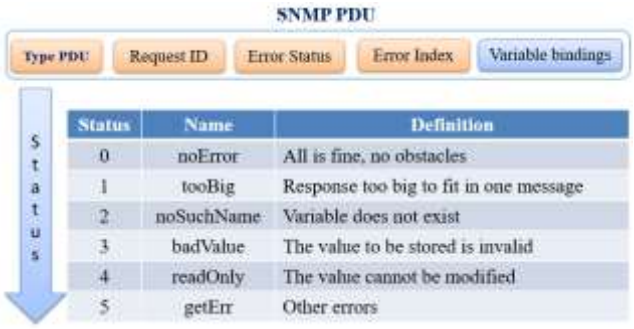


Figure 11. SNMP Issues Status

Integrated of MIB value and figure out accessing and storing problem also return value is PDU 3. SNMP package size is different from the underwater network then terrestrial network. Furthermore, the terrestrial network is consisting of a number of package size bigger than constrained network. Packet size will depend on how much information is sending. There is a UDP header and if you want to see the packet you need to perform a capture. The Figure 12 U- SNMP Message Formats. The PDU Header should use a minimum size of the data sending. There is three part of the message format, they are message preamble, PDU Header and PDU Body. Basically Message formats depend on version of SNMP also, each version of SNMP significantly different on it. The Message Preamble version – INTEGER value = 0, OCTET STRING for Community string, PDU is SEQUENCE of fields. Moreover, PDU header as before mentioned, we should be minimizing maximum value of header. Manager to send request of based on request id and agent should analysis of that message components. Most of error-index value is number on it, it can be Integer and OID value depends on ASN.1. Ver.1 is like abbreviation of version of PDU. The trap PDU consists of the following fields like enterprise – Object Identifiers, so address of Agent is using IP-address, trap types also using Integer. Indeed, trap relies on generic and specific trap.

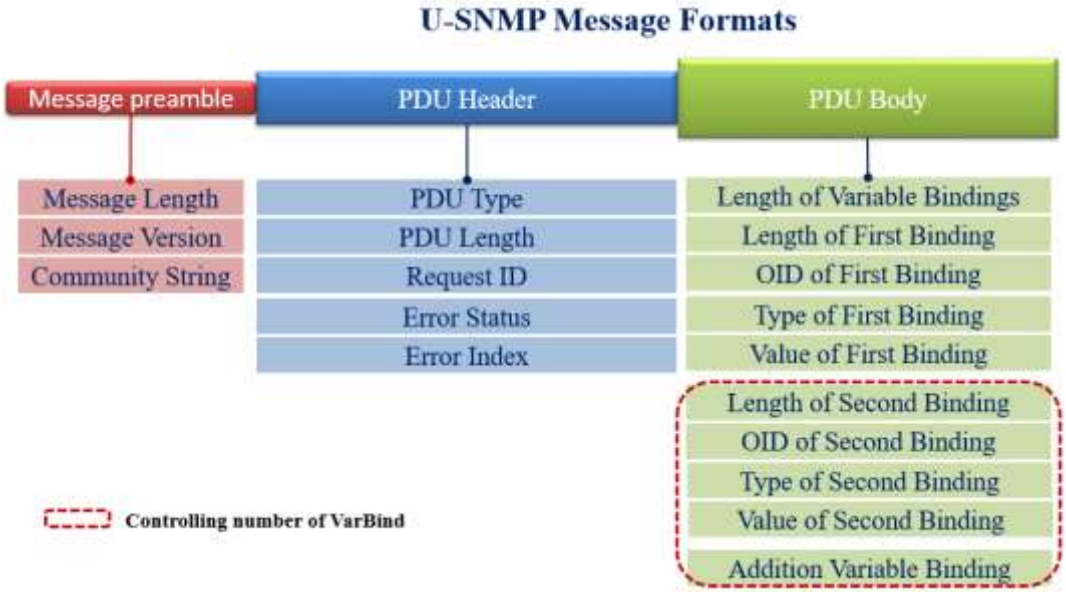


Figure 12. U- SNMP Message Formats

Configuration manager and agent procedure is setting on time-stamp and this function is time-ticks. SEQUENCE is one of vat-Bind type, they are all attached in SNMPv1 [18]. However, SNMPv2 message [17] composed of 3 field's value. This encoded in ASN.1 format (Version is Integer with value = 1, Community string is OCTET STRING, PDU value is SQEUNC of fields). Finally, SNMPv3 Messages [19] and PDU Formats wide explain security also msgID, msgMaxSize, msgFlags, msgSecurityModel – INTEGER. There are msgSecurityParameters and ScopedPDU value. PDU Body value has different VarBind in our underwater case we just using not much variable, red dotted cycle for extra case. If more variable binds and value, need to hardware performance.

4. u-SNMP Protocol Model Integrate of the IoUT

Application layer protocol U-SNMP is using UDP protocol not TCP. Manager and agent software are deploying all domination devices as well, so each device has different ability. That means a category of devices like the Proxy-Agent needs to deploy manager software and other sensor underwater devices need to install agent software. In this Figure 15 shown by MO elements setting configuration, and this point concerning getting OID from the manager. Sensors only need to trap method and variable binding for analyzing other methods requests. After that retrieve the response to the manager. The U-MIB part case is a generation of Object ID source code included there. The daunting task is a simulation of the U-IoT identification system. We have to attach near the feature our U-MIB. It should be helpful and complete all of the processes.

4.1. Implementation Constrained Network Protocol U-SNMP

Our suggestion of new protocol U-SNMP is becoming lightweight and easy integrated to the IoUT system. Before explanation emulation and implementation process we should explain more information U-SNMP methods (U-GET, U-SET, and U-Trap). The Figure 13 U-SNMP method and ID constructer illustrated three important things. The first one is Get and Set methods, second one is trap message construct, finally last one is 1-byte data Identification value constructed BER. Our prediction is concerning manager send request to agent maximum message value 30 bytes. Initial case of our prediction 30 byte, after improving hardware and software usable possibility more data size will be use as well. U-SNMP header includes version = 0 or 1 and then password choose number and text combination on it, afterwards PDU type data encoding and decoding structure. Identification relies on header part of message format and OID need to each MO.

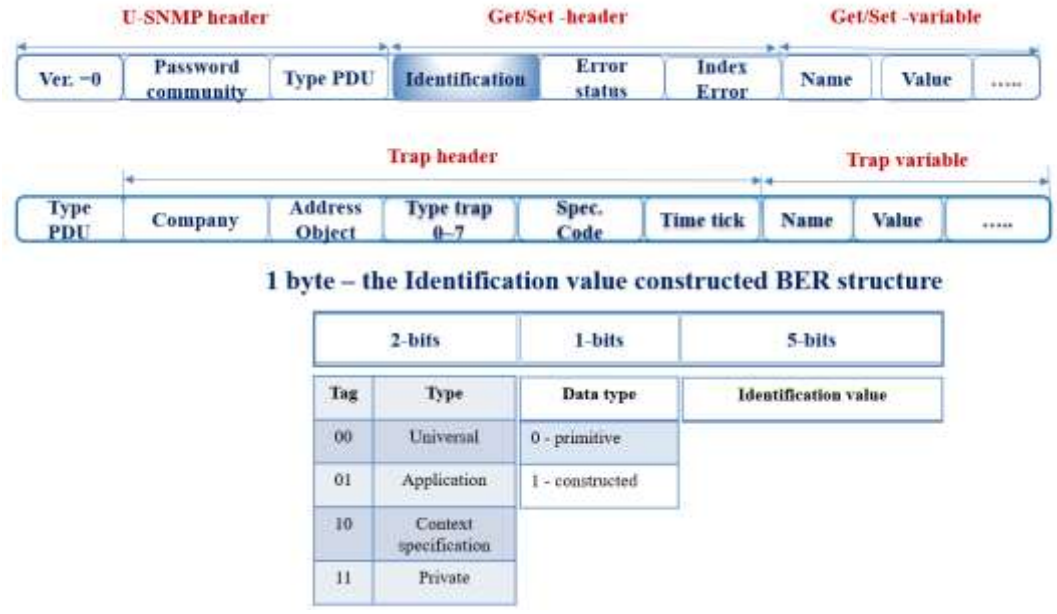


Figure 13. U-SNMP Method and ID Constructer

This case OID is getting 1-byte data. Error status and index are representing error status. Those error position are using Integer 32 type. Finally, the variable binding value is Name and Value as before mentioned. Data size description is concerning PDU type is 4 bytes. The Request ID and Error Status 3 bytes (legacy SNMP v.2 is 4 bytes) based on Integer 32. Error index is 3 bytes based on Integer 32. Indeed, ID constructor 1-byte

structure (2- bits category value) and 1- bit data type and 5 bits Identification value. Additionally, Figure 14 depicted emulation process model. The database containing the information about the network is populated by respective daemons executing on the sites. The management station archives the required information from the database (DMD) or from the agents located at the managed systems and provide the information to the managers as well when requested by them. Moreover, the manager has two different hardware devices. Initial devices for the manager is a computer which will install net-snmp software and it is possible to integrate the embedded system. The second manager system is Gateway or Proxy-Agent and that hardware enable to support net-snmp manager configuration. Additionally, Manager-console can adjust all of the structural possibility (Monitoring, uploading U-MIB, and Troubling shooting). Indeed, DMD all store data as well and manager console dominate this huge of data. There are manager installing on Ubuntu OS version 16.04, in this Figure 14 illustrated installing and updating snmp, and configure **snmpd** file. However, this process is ongoing laptop or server computer and then embedded Linux or RTOS for Gateway/ Proxy-Agent. We should explain the right side of Agent part. There is also Embedded Linux OS and using lightweight U-SNMP protocol. Hardware system relies on Bangle bone or Raspberry PI devices. Hence, U-MIB added there with Standard MIB.

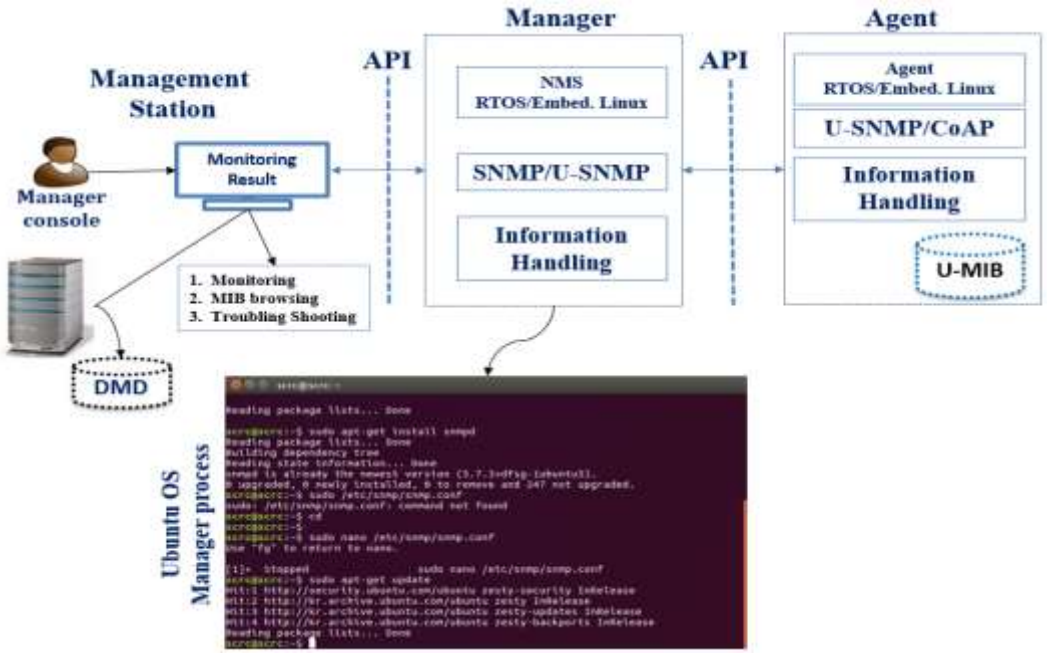


Figure 14. Emulation Process Model

We should configure and install API between manager and agent. Moreover, the managers are Java applications which have the functionally of network management and control. The managers are served by servers. The servers run on one or more sites containing the databases depending on the size of the network. At any instant, a server can serve more than one manager. In contrast with a typical centralized manager comprises of the Monitoring Result, the management applications and NMS RTOS along with the DMD. There can be multiple managers querying one or more servers (depending on the size of the network). Furthermore, we should describe this emulation architectural model to the real field IoUT field. As well this is a little complex but we should be depicted Figure 15. Architectural model of IoUT. More focus on underwater devices, like sensor nodes and relay node as so on. In this example, a sensor node has different variable bindings and Object instances. There is a diagram Managed Object it is called MO, so

shown compatible value (OID, Setting parameter, Data). The OID always related to addressing like unique identity value. Manager console strives to generated Object ID and adjusted algorithmic way to uniqueness. There is 8 bit or more value of OID. The next parameters composed of data and setting. The Data value is after sensing parameters of sensors. The Setting is gathering a different type of sensors. In this paper we primary focus on acoustic signal and time synchronization. Generally, the speed of the sound is 3 km/s in underwater like normal water. The sound speed also dynamically change shallow or deep water with pollution level.

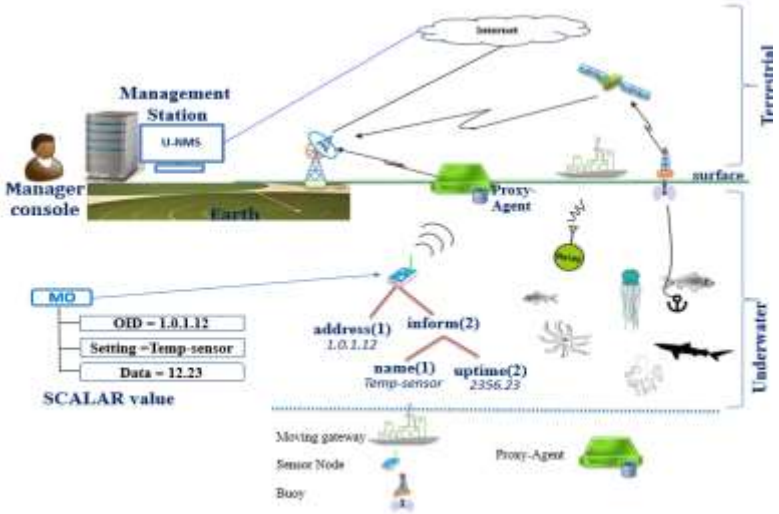


Figure 15. Architectural Model of IoUT

In our Figure 15 shown 10 or 15 minutes generally checking each underwater node status and gather data all of them. One of the main issues for the constrained environment is limitation memory and energy consumption. Improvement device possibility near the future we will deploy optical communication and interconnection of aquatics network.

5. Conclusion

The Internet is a big market and using a lot of technological purposes. In this paper also describe and defined management technology IoT and IoUT technology. However, the quantity of daunting task and challenges which we strive to solve those problems. We analysis several factors. At first, legacy NMS system elements and working technology. The second is using technology and protocol possibility SNMP. We already investigated MIB tree structure and SMI writing and reading technology. Additionally, how can get IANA unique number for our private MIB (U-MIB) and ASN.1 defining methods? We also example announced SCALAR and TABLE data capturing. Finally, in this paper, we endeavor to message format with PDU value. Last part of this paper we showed the architecture of u-SNMP interconnection to the IoUT technology and implementation process. Mainly, the implementation process is ongoing right now. Our main goal installs and deploys real test bed using the beagle-bone board (agent) or Raspberry Pi and manager. Likewise, we are developing our U-MIB. As before mentioned, we will attach to the standard MIB our private our U-MIB. Fortunately, we already have our U-NMS unique number from the IANA. In our future plan concerning of using real testbed of Underwater communication applying to u-SNMP. Moreover, the IoUT enabler should obtain methodic function to gathering and reveal information about current underwater sensor devices availability like power, memory storage, computing power, queue buffer approximately remaining available resource near the future IoUT system for Underwater.

In our future plan, make our underwater management protocol (like u-SNMP) and using this OID system for IoUT. Improve to the emulation and implementation process for integrate constrained network protocol as much as possible.

Acknowledgments

This research was a part of the project titled 'Development of Distributed Underwater Monitoring & Control Networks', funded by the Ministry of Oceans and Fisheries, Korea.

"U_CoAP: Well-deserved Constrained Environment Protocol for Underwater Internet of Things". International Conference on Green and Human Information Technology (ICGHIT-2017). Feb. 15 ~ Feb. 17, 2017 in Hangzhou, China.

References

[1] D. Mauro and K. Schmidt, "Essential SNMP", Help for System and Network Administrators. O'Reilly Media, Inc., **(2009)**.
[2] J. Jiang, X. L. Xu and N. Cao, "Research on Improved Physical Topology Discovery Based on SNMP", Computational Science and Engineering (CSE) and Embedded and Ubiquitous Computing (EUC), IEEE International Conference on. – IEEE– T. 2. – C. 219-222, **(2017)**.
[3] B. Appelman and M. M. Hussain, "Systems and methods for notification management and delivery, U.S. Patent No. 9,729,489. **(2017)**.
[4] C. C. Kao, "A Comprehensive Study on the Internet of Underwater Things: Applications, Challenges and Channel Models", Sensors, T. 17. – №. 7. – C. 1477, **(2017)**.
[5] R. R. Yager and J. P. Espada. "New Advances in the Internet of Things", **(2017)**.
[6] J. Schonwalder and V. Marinov, "On the Impact of Security Protocols on the Performance of SNMP", IEEE Transactions on Network and Service Management, vol. 8, no. 1, **(2011)**.
[7] A. Sehgal and P. Vladislav, "Management of resource constrained devices in the internet of things", IEEE Communications Magazine, vol. 50, no. 12, **(2012)**, pp. 144-149.
[8] C.-R. Dow, Y.-H. Lee and H. R. Yu, "Adaptive SWE and SNMP-based sensor management for environmental monitoring", International Journal of Communication Networks and Distributed Systems, vol. 13, (**2014)**, pp. 314-334.
[9] H.-P.Huang, S.-D. Xiao and X.-Y. Meng, "Applying SNMP Technology to Manage the Sensors in Internet of Things", Open Cybernetics & Systemic Journal, vol. 9, **(2015)**, pp. 1019-1024.
[10] S. Jana and S. Borkar, "Autonomous Object Detection and Tracking using Raspberry Pi", International Journal of Engineering Science. – T. 14145, **(2017)**.
[11] R. Hillbrecht and L. C. E. de Bona, "A SNMP-based virtual machines management interface", Utility and Cloud Computing (UCC), IEEE Fifth International Conference on. – IEEE, **(2012)**, pp. 279-286.d.
[12] H. Deng, G. Liu and L. Zhang, "Analysis and implementation of embedded SNMP agent", International Conference on Computer and Computing Technologies in Agriculture. – Springer, Berlin, Heidelberg, **(2010)**, pp. 96-102.
[13] R. S. Moreira, "A behavioral reflective architecture for managing the integration of personal ubicomp systems: automatic SNMP-based discovery and management of behavior context in smart-spaces", Personal and Ubiquitous Computing, vol. 20, no. 2, **(2016)**, pp. 229-243.
[14] M. Yang and C. Liu, "Research of SNMP-based network topology discovery algorithm in IoT". **(2016)**.
[15] "Create SNMP Client in JAVA Using SNMP4j" [online], Available: http://www.jitendrazaa.com/blog/java/snmp/create-snmp-client-in-java-using-snmp4j/, **(2017)**.
[16] "Implementing SNMP Proxy" [online], Available: https://www.webnms.com/cagent/help/snmp/c_snmp_proxy.html, **(2017)**.
[17] D. Harrington, B. Wijnen and R. Presuhn, "An architecture for describing simple network management protocol (SNMP) management frameworks", (**2002)**.
[18] J. D. Case, "Simple network management protocol (SNMP)", №. RFC 1157, **(1990)**.
[19] B. Wijnen and U. Blumenthal, "User-based security model (USM) for version 3 of the simple network management protocol", (SNMPv3), (**2002).**
[20] "Private enterprise numbers" [online], Available: https://www.iana.org/assignments/enterprise-numbers/enterprise-numbers, **(2017)**.
[21] "OiDViEW SNMP MIB Browser and Network Management Toolset" [online], Available: http://www.oidview.com/oidview.html, **(2017)**.

Authors

Khamdamboy Urunov, he has received his B.S. degree in Information Technologies department at Tashkent University of Information Technology of Urgench branch, 2009. He graduated master degree department of Applied Informatics at Tashkent University of Information Technologies, Tashkent, Uzbekistan, 2011. He is studying the Ph.D. degree in Financial Information Security department of Kookmin University, Seoul, Korea. His current research interests includes Internet of Underwater Things (IoUT) and Network Management System (NMS). He is developing u-SNMP protocol as well.

Shin, Soo Young, she received her M.S. degree in Information and Communication from Duksung Women's University in 2003 and Ph.D. degree in Department of Information System from Kookmin University, Seoul, Korea, in 2007. Now, she is a vice-director in Special Communication Research Center, Kookmin University, and Seoul, Korea. Her research interests include Underwater Sensor Network and Wireless MAC Protocol.

Soo-Hyun Park, he received his B.S., M.S. and Ph.D. degrees in computer science & engineering from Korea University, Seoul, Korea, in 1988, 1990 and 1998, respectively. He worked as a senior research engineer of the Central R&D Complex, LG Information & Communications, Ltd from 1990 to 1999 in Anyang, Korea. Now, He is a professor in Department of Information System, Kookmin University, and Seoul, Korea. His current research interests include Ubiquitous Network and Underwater Sensor Network.

Dr. Kwan Yi, he is an assistant professor at the Department of Curriculum and Instruction, Library Program. He holds a Ph.D. in Library and Information Science from McGill University, Master degrees in Computer Science and Applied Mathematics from McGill University and the University of Illinois, Urbana-Champaign, respectively, and BS in Computer Science and BA in Mass Communications, both from Korea University. At EKU, he is primarily teaching courses in the areas of information classification and information technology. His broad research interests lies in the representation, organization and classification, and access of digital information resources. For the last six years, he published about 20 articles at scholarly journals and conference proceedings in library and information science field. Prior to joining the faculty at EKU, he taught in the library and information science at the University of Kentucky. He had also worked in a number of different information technology fields as computer programmer and research staff, at the Information Research Center at Montreal (CRIM - Centre de Recherché Informatique de Montréal), Montreal in Canada, the U.S. Army Construction Engineering Research Laboratory, Illinois, and Korea Securities Dealers Association, Seoul, Korea.