Software Design of Packet Analyzer based on Byte-Filtered Packet Inspection Mechanism for UW-ASN

Sardorbek Muminov[†], Nam-Yeol Yun[††], Soo-Hyun Park[†††]

ABSTRACT

The rapid growth of UnderWater Acoustic Sensor Networks (UW-ASNs) has led researchers to enhance underwater MAC protocols against limitations existing in underwater environment. We propose the customized robust real-time packet inspection mechanism with addressing the problem of the search for the data packet loss and network performance quality analysis in UW-ASNs, and describe our experiences using this approach. The goal of this work is to provide a framework to assess the network real-time performance quality. We propose a customized and adaptive mechanism to detect, monitor and analyze the data packets according to the MAC protocol standards in UW-ASNs. The packet analyzing method and software we propose is easy to implement, maintain, update and enhance. We take input stream as real data packets from sniffer node in capture mode and perform fully analysis. We were interested in developing software and hardware designed tool with the same capabilities which almost all terrestrial network packet sniffers have. Experimental results confirm that the best way to achieve maximum performance requires the most adaptive algorithm. In this paper, we present and offer the proposed packet analyzer, which can be effectively used for implementing underwater MAC protocols.

Key words: Underwater Acoustic Sensor Networks (UW-ASNs), Packet Analyzer, Packet Sniffer, packet detection, Medium Access Control (MAC)

1. INTRODUCTION

UnderWater Acoustic Sensor Networks (UW-ASNs) are envisioned to enable applications for oceanographic data collection, pollution monitoring, offshore exploration, disaster prevention, assisted navigation and tactical surveillance applications. Multiple unmanned or autonomous underwater vehicles (UUVs, AUVs), equipped with underwater sensors, will also find application in exploration of natural undersea resources and gathering of scientific data in collaborative monitoring missions [1,2]. UW-ASNs consist of a variable number of sensors and vehicles that are deployed to perform collaborative monitoring tasks over a given area. To achieve this objective, sensors and vehicles self-organize in an autonomous network which can adapt to the characteristics of the ocean environment[3]. Its fact that, underwater sensor networks shares many common properties with terrestrial sensor networks, such as the large number of nodes and energy issues, but yet UW-ASNs are different in many ways from the conventional terrestrial sensor technology. First, radio communications can

※ Corresponding Author : Soo-Hyun Park, Address : (136-702) C-303, Business Administration Building, Kookmin University, Jeongneung-ro 77, Seongbuk-gu, Seoul , TEL : +82-2-910-5085, FAX : +82-2- 910-4017, E-mail : shpark21@kookmin.ac.kr

Receipt date : Nov. 1, 2011, Revision date : Dec. 16, 2011
Approval date : Dec. 28, 2011

[†] Ubiquitous System Lab., Graduate School of BIT, Kookmin University
(E-mail: smuminov@kookmin.ac.kr)
[††] Ubiquitous System Lab., Graduate School of BIT, Kookmin University
(E-mail: anuice@kookmin.ac.kr)
[†††] School of Management Information Systems, Kookmin University

※ This research was supported by the MKE (The Ministry of Knowledge Economy), Korea, under the ITRC (Information Technology Research Center) support program supervised by the NIPA (National IT Industry Promotion Agency) (NIPA-2011-C1090-1121-0001).

not propagate well in deep water, so have to replace this with the acoustic communications as sound propagate well in water as compare to radio signal. Due to this replacement, propagation speed is five orders of magnitude less than the radio frequency, as the characteristics of communication changes from the speed of light to the speed of sound. Second, while most terrestrial sensors nodes are static, underwater sensor nodes can move due to different underwater activities, as during normal conditions a node can move 2-3 m/sec with water currents [3-5]. Ian F. Akyildiz and Dario Pompili, et al. [6], overviewed the main challenges for efficient communications in underwater acoustic sensor networks. They described the peculiarities of the underwater channel with particular reference to networking solutions for monitoring applications of the ocean environment. As we know, mobile underwater acoustic networks can be envisioned in two forms: centralized, which rely on an infrastructure of base stations to relay data, and distributed, where nodes communicate to each other directly or by establishing paths through other nodes. As an quick introduction to UW-ASN, in this paper [7] authors focused on the latter type of network, and in particular on an ad-hoc network in which vehicles establish communication links autonomously upon deployment. There have been studied many researches for finding optimal packet size for Data-Link layer protocols in UW-ASN [8] and recently implemented in various applications [2] based on UW-ASN.

Network packet[9] contains information about network activity that is the detailed description of the general network behavior. Network packet analyzers [10] became a useful tool for network administrators to capture such kind of network information. Packet sniffing is a special technique to intercept data which flow across the network from one host to another ones. Most of terrestrial network sniffers support all existing protocols for fully analysis. Packet sniffers help users to debug the network to ensure what kind of information is going to which host from where. In terrestrial networks, packet sniffers are able to operate fully in systems where network adapter hardware supports promiscuous mode. Promiscuous mode [9,10] is a mode for network interface controller which enables to pass all traffic to receive passing frames through the connected network. Some packet sniffers can also generates fake traffic and acts as a real device. Those are also called protocol testers.

The remainder of the paper will concentrate the existing packet detection mechanisms and our proposed mechanism in Section 2. Then in Section 3, we present our approach and system architecture design. Also the results which we have received during implementation, all characteristics and its implementation in Section 4. Finally, we finish our paper concluding in Section 5.

2. PACKET INSPECTION MECHANISM

Packet analyzers are often used as debugging tool for monitoring network performance. Most of packet analyzers [11] have Graphic User Interface (GUI) with having graphic components to illustrate current network state, e.g. amount of data collisions, number of broken packets and so on. There are many novell researches have been done in last decades related to packet analysis methods. Some of them are extracting information from binary data and some adaptive methods helped to extend this area. We describe the main issue to develop packet analyzer for UW-ASN is to precise the network performance and estimate problems all it has. But in UW-ASN, there is no researches have been done especially for the network debugging field. Underwater applications meet many restrictions in underwater environment. If we suppose there are three nodes need to be communicated each other according to the existing routing protocols and when the node sends message to the second node, at that time the third node also receives the same

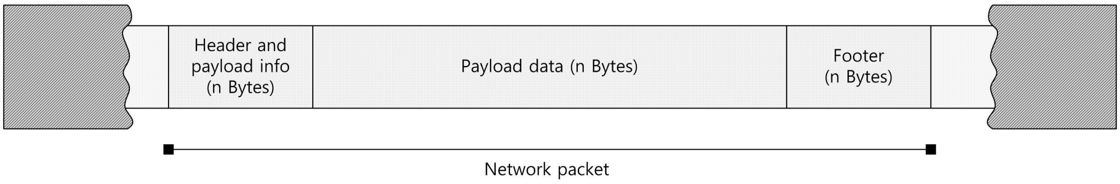


Fig. 1. Network packet in data stream

packet which second one should. It is often called underwater multipath propagation. In other case, if first node sends message to other one, the receiving node gets packet having continuously trailing of last byte. The other important issues can be reasonable against making robust network connectivity. Unlike in terrestrial networks, there is no any similar issues have been met. In UW-ASN there are some of main problems exist. They are propagation delay between transmissions, limited bandwidth and latency.

Therefore, main considerable opportunity for packet analyzer is UW-ASN's having low throughput. It means we can examine and analyze the data which we have captured before. But it doesn't meet real-time packet analyzer requirements. Assume we have data stream flows across the network and when parsing stream into packets there one unique network packet became as in shown in Figure 1. Packet header, payload data and footer information are different. For improving the detection of packet header, payload and footer information, we proposed packet extraction block diagram in Figure 2. Flowing data stream is monitored by capture monitor and saved in buffer then directed to three type of extraction mechanisms. We describe their operations in detailed figures and algorithms. The first mechanism which we have is token-byte mechanism. It extracts packet by filtering bytes which are chosen as token bytes. For example, if we know that our network packets have sequence numbering bytes in contents, then we set our token bytes to extract the packets. The packet can be extracted from the beginning of its first token byte to the last first-met token-byte in stream. That we called it token-byte mechanism. The second one is separator-byte mechanism. It detects and extracts packets by filtering the constant separator-byte.

If we have network packets which are consisted of 0xF8 separator-byte between every packets, then we detect, extract by estimating the packet length by filtering only the given separator-byte. The last one we named is pattern-match mechanism. Terrestrial network sniffers also have such kind of functions, for example if we give some packet pattern then we can extract packets by comparing the given packet pattern. We have been interested in software based packet analyzer which can help us to capture and analyze flowing network

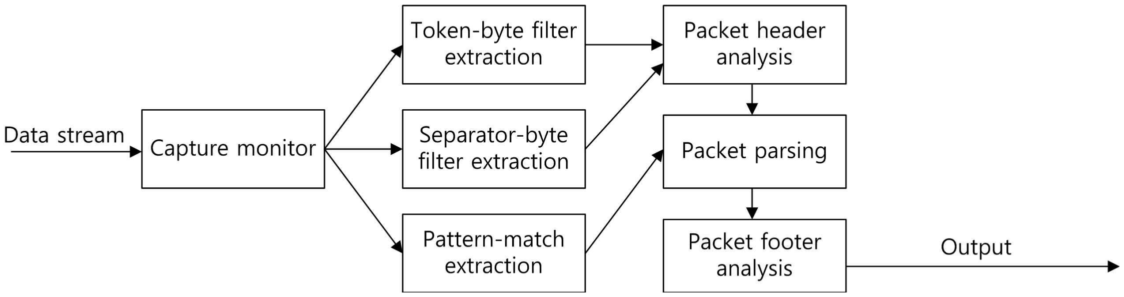


Fig. 2. Packet extraction block diagram.

data packets in real-time. In the following subsection, we consider packet analyzing methods existing in terrestrial networks and compare them according to the UW-ASN characteristics.

Many researches have been done on packet inspection methods for terrestrial networks. Most of them considered and developed automatic extraction methods, such as using dynamic binary array analysis [12], scenario-based packet extraction language for advanced protocol analysis [13], application for parsing protocols [14], also packet scheduling techniques for multi-core architectures [15], packet encoding and decoding methods for packet combining in sensor networks [16] and string-matching for improving DPI [17-19]. We should monitor what kind of packet is being broken every transmission time. And the main purpose of analysis is what kind of data collisions we meet in our deployed network. Packet analysis has sparked renewed research interest due to its usefulness for deep packet inspection in applications such as intrusion detection, virus scanning, and content filtering. Matching expressive pattern specifications with a scalable and efficient design, accelerating the entire packet flow, and string matching with high-level semantics are promising topics for further researches, despite existing research, the study of string matching for DPI still has a way to go in the near future.

In addition to the growing of packet inspection techniques that make scalability, matching with semantically contextual information also complicates the traditional model of byte-filtered packet inspection mechanism. Dealing with this complication is particularly significant because many existing efforts in terrestrial networks still use the traditional model to develop their solutions.

3. APPROACH AND SYSTEM ARCHITECTURE

Our design provides adaptive framework for hardware and software implementations, from memory minimized to performance maximized. In

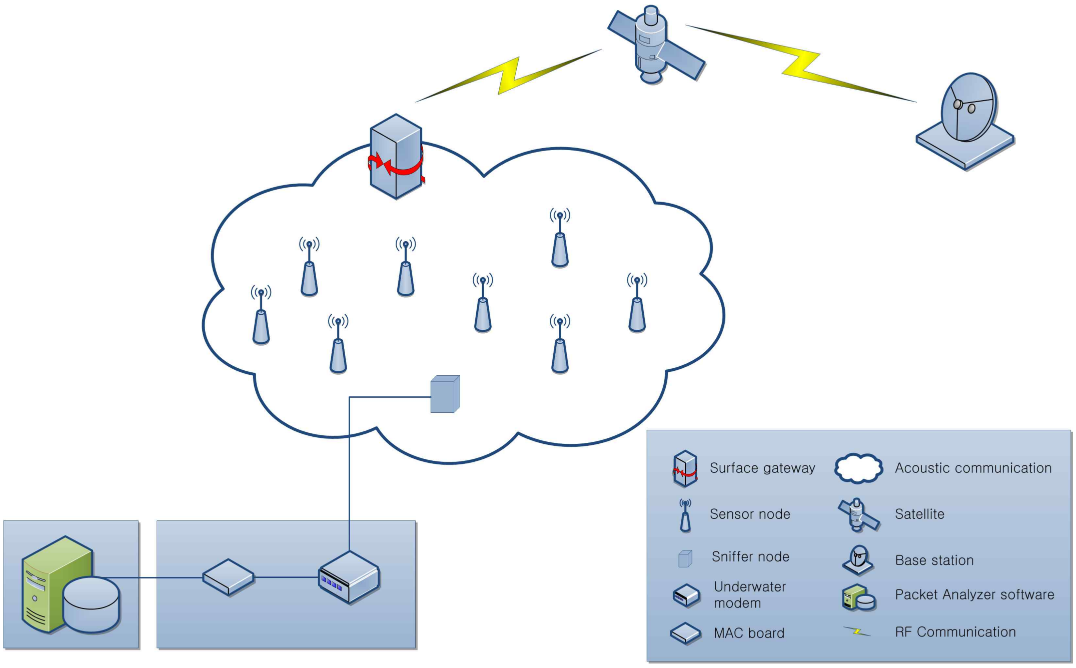


Fig. 3. Basic system architecture

this section, we will explore the details of the system architecture. The basic system architecture of our packet analyzer is shown in Figure 3.

We named one node as sniffer node for making the architecture more explainable. Sniffer node is the regular node having it's own transducer and it everytime stands on Rx mode for receiving all packets. As we know, in UWASNwe use surface gateway to send the information via RF communication towards the base station. Our packet analyzer has to have a database support due to save all information of captured data. Sniffer node has its transduce connected to the underwater modem and MAC base-board connected to PC on which packet analyzer runs. This architecture is totally same with terrestrial network architecture, but UW-ASN may differ of it's network topology.

3.1 Token-byte mechanism

The most appropriate data structure for implementing packet inspection is filtering the bytes which were given already. And it is very efficient for further operations.

By employing this mechanism each for a captured data, we can get a set of filtered packets at all. According to the match results, we can ignore broken packets or save them to database for further analysis. In figure 4, bytes which will be filtered are called leading and closing token-bytes. Fortunately, for analyzing captured data we need such mechanism which can function upon overlapped packets.

It indicates that we can get the results of all analyzed packets including not-matched and broken packets. To make an approach to this mechanism we proposed our designed algorithm in Figure 5.

3.2 Separator-byte mechanism

A possible yet prevent data collision may occur when the data stream has different type of packets randomly.

In case of this, a configurable separator-byte can be accessed at earlier stage. For example, if our data stream cannot be analyzed through token-byte filter we can set separator-byte to parse data stream into several packets. Separator-byte means the unique or fixed byte that appears repeatedly in data stream for separating the packet header, main payload part and footer. Receiver knows which byte is separator byte and it receives full packet in Figure 6 and designed algorithm is shown in Figure 7.

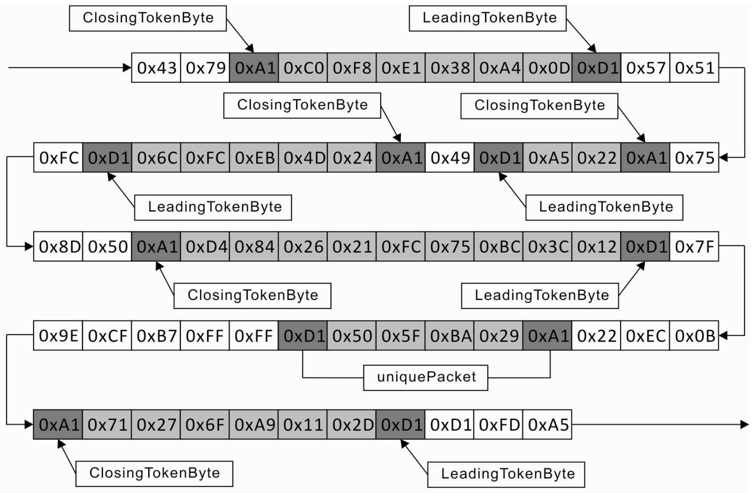
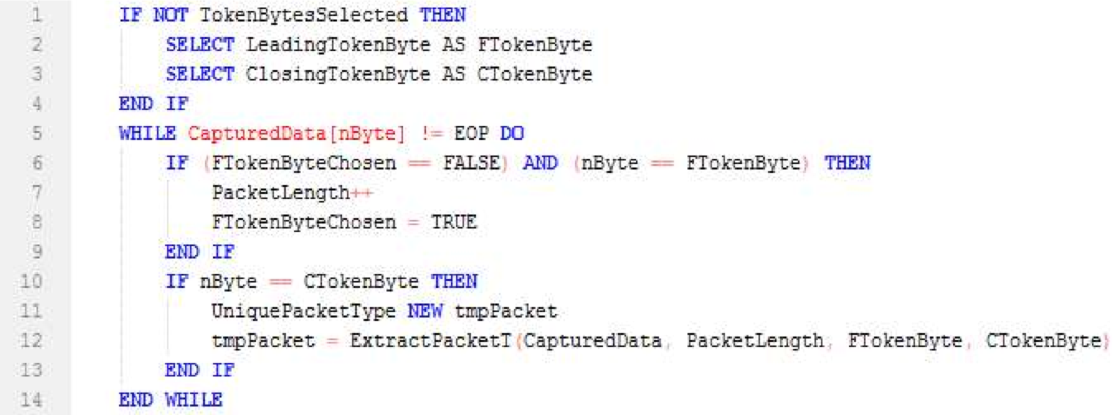

ClosingTokenByte
LeadingTokenByte
0x43 0x79 0xA1 0xC0 0xF8 0xE1 0x38 0xA4 0x0D 0xD1 0x57 0x51
ClosingTokenByte
ClosingTokenByte
0xFC 0xD1 0x6C 0xFC 0xEB 0x4D 0x24 0xA1 0x49 0xD1 0xA5 0x22 0xA1 0x75
LeadingTokenByte
LeadingTokenByte
0x8D 0x50 0xA1 0xD4 0x84 0x26 0x21 0xFC 0x75 0xBC 0x3C 0x12 0xD1 0x7F
ClosingTokenByte
LeadingTokenByte
0x9E 0xCF 0xB7 0xFF 0xFF 0xD1 0x50 0x5F 0xBA 0x29 0xA1 0x22 0xEC 0x0B
uniquePacket
0xA1 0x71 0x27 0x6F 0xA9 0x11 0x2D 0xD1 0xD1 0xFD 0xA5
ClosingTokenByte
LeadingTokenByte

Fig. 4. Token-byte mechanism.



```
IF NOT TokenBytesSelected THEN
    SELECT LeadingTokenByte AS FTokenByte
    SELECT ClosingTokenByte AS CTokenByte
END IF
WHILE CapturedData[nByte] != EOP DO
    IF (FTokenByteChosen == FALSE) AND (nByte == FTokenByte) THEN
        PacketLength++
        FTokenByteChosen = TRUE
    END IF
    IF nByte == CTokenByte THEN
        UniquePacketType NEW tmpPacket
        tmpPacket = ExtractPacketT(CapturedData, PacketLength, FTokenByte, CTokenByte)
    END IF
END WHILE
```

Fig. 5. Token-byte mechanism algorithm.

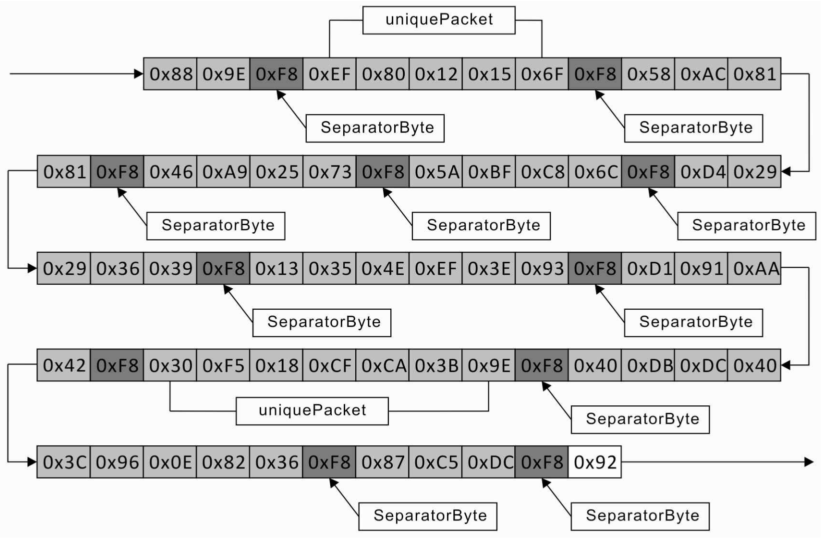
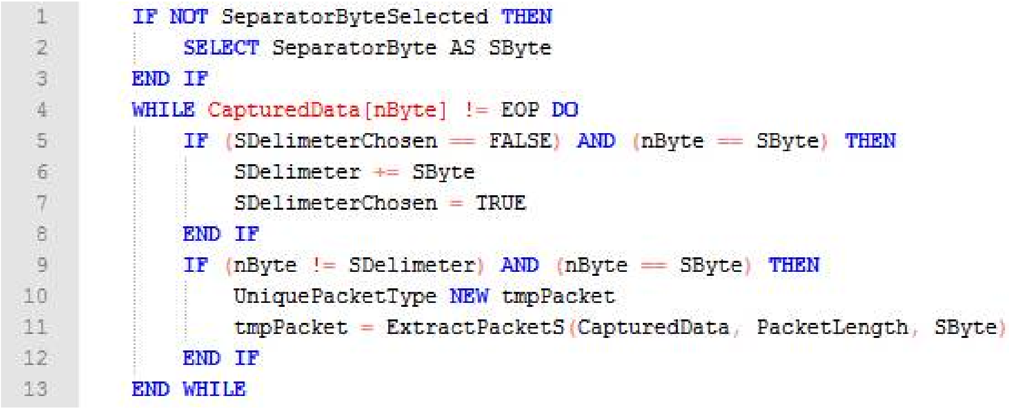

uniquePacket
0x88 0x9E 0xF8 0xEF 0x80 0x12 0x15 0x6F 0xF8 0x58 0xAC 0x81
SeparatorByte
SeparatorByte
0x81 0xF8 0x46 0xA9 0x25 0x73 0xF8 0x5A 0xBF 0xC8 0x6C 0xF8 0xD4 0x29
SeparatorByte
SeparatorByte
SeparatorByte
0x29 0x36 0x39 0xF8 0x13 0x35 0x4E 0xEF 0x3E 0x93 0xF8 0xD1 0x91 0xAA
SeparatorByte
SeparatorByte
0x42 0xF8 0x30 0xF5 0x18 0xCF 0xCA 0x3B 0x9E 0xF8 0x40 0xDB 0xDC 0x40
uniquePacket
SeparatorByte
0x3C 0x96 0x0E 0x82 0x36 0xF8 0x87 0xC5 0xDC 0xF8 0x92
SeparatorByte
SeparatorByte

Fig. 6. Separator-byte mechanism.



```
IF NOT SeparatorByteSelected THEN
    SELECT SeparatorByte AS SByte
END IF
WHILE CapturedData[nByte] != EOP DO
    IF (SDelimeterChosen == FALSE) AND (nByte == SByte) THEN
        SDelimeter += SByte
        SDelimeterChosen = TRUE
    END IF
    IF (nByte != SDelimeter) AND (nByte == SByte) THEN
        UniquePacketType NEW tmpPacket
        tmpPacket = ExtractPacketS(CapturedData, PacketLength, SByte)
    END IF
END WHILE
```

Fig. 7. Separator-byte mechanism algorithm

3.3 Pattern-match mechanism

If we have a network data stream that cannot be analyzed neither with token-byte or separator bye, then we can easily set some packet pattern.

In figure 8, if we have 7 bytes consisted packet and the packet has 2 bytes header, 4 bytes data payload and 1 byte footer, then we should make appropriate packet-pattern including header and footer. In our designed algorithm which is shown in figure 9, mechanism sets the total packet length by estimating the given pattern length.

Pattern-match mechanism is often called as extraction by static filtering.

4. IMPLEMENTATION

Especially, proposed packet analyzer needs to be implemented using with suitable underwater acoustic modem. After the connection between the hardware board and the PC software is established, developers can choose the packet extraction mechanism to detect the packets. After clicking the con-

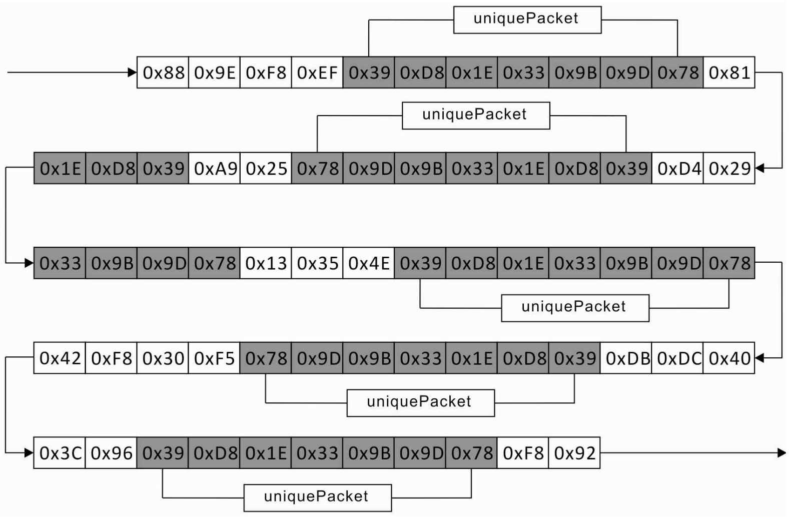
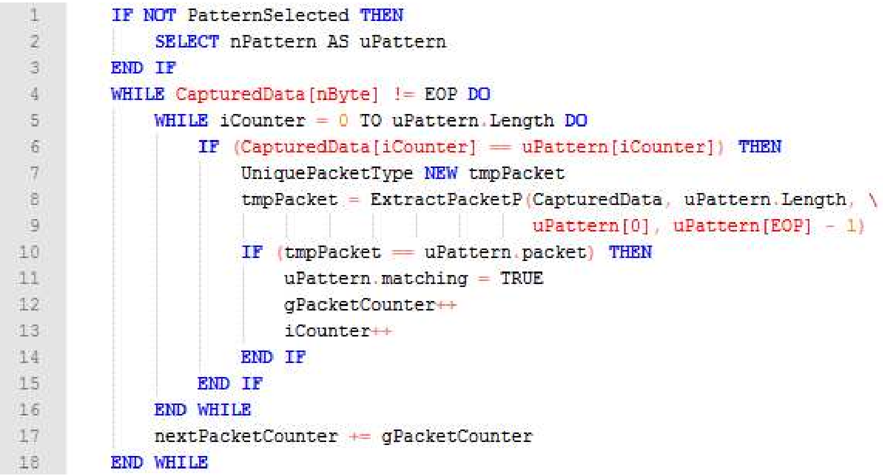
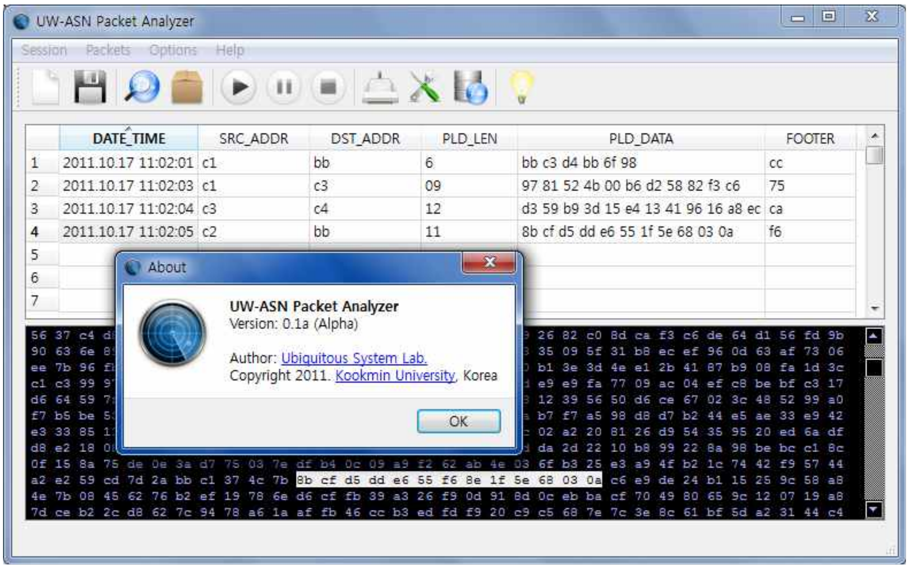

uniquePacket
0x88 0x9E 0xF8 0xEF 0x39 0xD8 0x1E 0x33 0x9B 0x9D 0x78 0x81
uniquePacket
0x1E 0xD8 0x39 0xA9 0x25 0x78 0x9D 0x9B 0x33 0x1E 0xD8 0x39 0xD4 0x29
0x33 0x9B 0x9D 0x78 0x13 0x35 0x4E 0x39 0xD8 0x1E 0x33 0x9B 0x9D 0x78
uniquePacket
0x42 0xF8 0x30 0xF5 0x78 0x9D 0x9B 0x33 0x1E 0xD8 0x39 0xDB 0xDC 0x40
uniquePacket
0x3C 0x96 0x39 0xD8 0x1E 0x33 0x9B 0x9D 0x78 0xF8 0x92
uniquePacket

Fig. 8. Patter-match mechanism.



```
IF NOT PatternSelected THEN
    SELECT nPattern AS uPattern
END IF
WHILE CapturedData[nByte] != EOP DO
    WHILE iCounter = 0 TO uPattern.Length DO
        IF (CapturedData[iCounter] == uPattern[iCounter]) THEN
            UniquePacketType NEW tmpPacket
            tmpPacket = ExtractPacketP(CapturedData, uPattern.Length, \
                                    uPattern[0], uPattern[EOP] - 1)
            IF (tmpPacket == uPattern.packet) THEN
                uPattern.matching = TRUE
                gPacketCounter++
                iCounter++
            END IF
        END IF
    END WHILE
    nextPacketCounter += gPacketCounter
END WHILE
```

Fig. 9. Pattern-match mechanism algorithm



	DATE_TIME	SRC_ADDR	DST_ADDR	PLD_LEN	PLD_DATA	FOOTER
1	2011.10.17 11:02:01	c1	bb	6	bb c3 d4 bb 6f 98	cc
2	2011.10.17 11:02:03	c1	c3	09	97 81 52 4b 00 b6 d2 58 82 f3 c6	75
3	2011.10.17 11:02:04	c3	c4	12	d3 59 b9 3d 15 e4 13 41 96 16 a8 ec	ca
4	2011.10.17 11:02:05	c2	bb	11	8b cf d5 dd e6 55 1f 5e 68 03 0a	f6
5						
6						
7						

About

UW-ASN Packet Analyzer
Version: 0.1a (Alpha)

Author: Ubiquitous System Lab.
Copyright 2011. Kookmin University, Korea

```
56 37 c4                                                 26 82 c0 8d ca f3 c6 de 64 d1 56 fd 9b
90 63 6e                                                 35 09 5f 31 b8 ec ef 96 0d 63 af 73 06
ee 7b 96                                                 b1 3e 3d 4e e1 2b 41 87 b9 08 fa 1d 3c
c1 c3 99                                                 e9 e9 fa 77 09 ac 04 ef c8 be bf c3 17
d6 64 59                                                 12 39 56 50 d6 ce 67 02 3c 48 52 99 a0
f7 b5 be                                                 b7 f7 a5 98 d8 d7 b2 44 e5 ae 33 e9 42
e3 33 85                                                 02 a2 20 81 26 d9 54 35 95 20 ed 6a df
d8 e2 18                                                 da 2d 22 10 b8 99 22 8a 98 be bc c1 8c
0f 15 8a 75 de 0e 3a d7 75 03 7e df b4 0c 09 a9 f2 62 ab 4e 03 6f b3 25 e3 a9 4f b2 1c 74 42 f9 57 44
a2 e2 59 cd 7d 2a bb c1 37 4c 7b 8b cf d5 dd e6 55 f6 8e 1f 5e 68 03 0a c6 e9 de 24 b1 15 25 9c 58 a8
4e 7b 08 45 62 76 b2 ef 19 78 6e d6 cf fb 39 a3 26 f9 0d 91 8d 0c eb ba cf 70 49 80 65 9c 12 07 19 a8
7d ce b2 2c d8 62 7c 94 78 a6 1a af fb 46 cc b3 ed fd f9 20 c9 c5 68 7e 7c 3e 8c 61 bf 5d a2 31 44 c4
```

Fig. 10. Software design of packet analyzer

nection icon to begin the capture, packet analyzer sends the start capture command to hardware. When captured data comes from hardware side then all data should be defined by their contents with choosing what kind of mechanism needed to extract the packet. If there any collision occurs during analysis it can also be displayed in terminal window. Main conceptual idea is to analyze the protocol packets and we need all data packets also including broken packets. As shown in Figure 10, our proposed packet analyzer has GUI components. We have three functional buttons, such as, START, PAUSE and STOP buttons to control the packet analysis process. Also we designed character-based terminal window and packet details window to display all information. User can control the connection and packet buffer size in connections panel. Analysis process shows us which packet will be analyzed with selecting it defined color in selected sections.

Three considerations drive performance for packet payload searching: size, matching complexity and packet footer analysis.

a) Size: Both number of bytes and packet size can vary substantially and greatly impact search performance.

b) Matching Complexity: The performance of filter processor may vary greatly in response to regular expressions which require significant backtracking or result in state explosion.

c) Packet Footer Analysis: Naturally, calculating the payload checksum or CRC values given in packet footer and comparing them can also be consideration point for improving the built-in error detection mechanism.

5. CONCLUSION AND FUTURE WORKS

In this paper, we proposed new approach on finding packet detection mechanisms and software design of packet analyzer in UW-ASN. The performance gains shown are promising, in light of the simple design choices made. Integrating our packet analysis algorithm leads to improve capturing and analyzing network packets realtime. We named packet analysis as a function of debugging network performance. Our analysis explicitly considers the probability of missed packets, which means received packets may be same type or they became because of underwater limitations. In this proposed packet analyzer we tried to get more debug information of network, but we didn't consider illustrating their performance in graphical appearance. Illustrating realtime packets and their source/destination nodes helps developers to monitor protocol performances. Our future researches may consist developing such kind of tool [20], with which its possible to illustrate realtime UW-ASN.

This paper proposed the packet inspection and extraction mechanism with dealing problem of extracting the protocol packet. The protocol packet includes multiple message formats. Our problem is then, given a number of captured data and extract the packet of each of those data streams according to the protocol formats. The main challenge in extracting the packet is to find the field boundaries. In our future researches we try to improve our proposed packet analyzer's mechanism addressing the problem RSSI, LQI and error detection analysis.

ACKNOWLEDGEMENT

This research was supported by the MKE (The Ministry of Knowledge Economy), Korea, under the ITRC (Information Technology Research Center) support program supervised by the NIPA (National IT Industry Promotion Agency) (NIPA-2011-C1090-1121-0001)

REFERENCES

[1] Ian F. Akyildiz, Dario Pompili, and Tommaso Melodia, "Underwater Acoustic Sensor Networks: Research Challenges," *Ad Hoc Networks*, Vol.3, No.3, 2005.

[2] John Heidemann, Wei Ye, Jack Wills, Affan Syed, and Yuan Li, "Research Challenges and Applications for Underwater Sensor Networking," *Wireless Communications and Networking Conference*, pp. 228-235, 2006.

[3] Muhammad Ayaz and Azween Abdullah, "Underwater Wireless Sensor Networks: Routing Issues and Future Challenges," *In Proceedings of the 7th International Conference on Advances in Mobile Computing and Multimedia (MoMM '09)*. ACM, New York, USA, pp. 370-375, 2009.

[4] Soo-Young Shin, Seung-Joo Lee, and Soo-Hyun Park, "MA : Multiple Acknowledgement Mechanism for UWSN (Under-Water Sensor Network)," *Journal of Korea Multimedia Society*, Vol.12, No.12, pp. 1769-1777, 2009.

[5] Jim Partan, Jim Kurose, and Brian Neil Levine, "A Survey of Practical Issues in Underwater Networks," *In Proceedings of the 1st ACM international workshop on Underwater networks (WUWNet '06),* ACM, New York, USA, pp. 17-24, 2006.

[6] Ian F. Akyildiz, Dario Pompili, and Tommaso Melodia, "Challenges for Efficient Communication in Underwater Acoustic Sensor Networks," *SIGBED Rev.,* Vol.1, No.2, 2004.

[7] Francisco Salva-Garau and Milica Stojanovic, "Multi-Cluster Protocol for Ad Hoc Mobile Underwater acoustic networks," *OCEANS 2003*. Proceedings, pp. 91-98 2003.

[8] Milica Stojanovic, "Optimization of a Data Link Protocol for An Underwater Acoustic Channel," *Oceans 2005-Europe,* Vol.1, pp. 68-73, 2005.

[9] Andrew S. Tanenbaum. "Computer Networks (5th Edition)." *Prentice-Hall, Inc., Upper S-addle River*, NJ, USA. October 7, 2010

[10] Sabeel Ansari, Rajeev S.G., and Chandrashekar H.S., "Packet Sniffing: a Brief Introduction," *Potentials, IEEE*, Vol.21, No.5, pp. 17-19, Dec 2002/Jan 2003.

[11] Susan Mengel and Salman Ali, "A Network Protocol Analyzer With Tutorial." *In Proceedings of the ACM symposium on Applied Computing (SAC '96)*, K. M. George, Janice H. Carroll, Dave Oppenheim, and Jim Hightower (Eds.). ACM, New York, NY, USA,

pp. 115-119, 1996.

[12] Juan Caballero, Heng Yin, Zhenkai Liang, and Dawn Song. “Polyglot: Atomatic Extraction of Protocol Message Format Using Dynamic Binary Analysis.” *In Proceedings of the 14th ACM conference on Computer and communications security (CCS '07).* ACM, New York, NY, USA, pp. 317-329. 2007.

[13] Tim Reichert, Edmund Klaus, Wolfgang Schoch, Ansgar Meroth, and Dominikus Herzberg. “A Language for Advanced Protocol Analysis in Automotive Networks.” *In Proceedings of the 30th international conference on Software engineering (ICSE '08).* ACM, New York, NY, USA, pp. 593-602. 2008.

[14] Ruoming Pang, Vern Paxson, Robin Sommer, and Larry Peterson. “Binpac: a Yacc for Writing Application Protocol Parsers.” *In Proceedings of the 6th ACM SIGCOMM conference on Internet measurement (IMC '06).* ACM, New York, NY, USA, pp. 289-300. 2006.

[15] Terry Nelms and Mustaque Ahamad. “Packet Scheduling for Deep Packet Inspection on Multi-Core Architectures.” *In Proceedings of the 6th ACM/IEEE Symposium on Architectures for Networking and Communications Systems (ANCS '10).* ACM, New York, NY, USA, Article 21, 11 pages. 2010.

[16] Henri Dubois-Ferriere, Deborah Estrin, and Martin Vetterli. “Packet Combining in Sensor Networks.” *In Proceedings of the 3rd international conference on Embedded networked sensor systems (SenSys '05).* ACM, New York, NY, USA, pp. 102-115. 2005.

[17] Kun Huang and Dafang Zhang, “A Byte-Filtered String Matching Algorithm for Fast Deep Packet Inspection,” *ICYCS 2008*, pp. 2073-2078, 2008.

[18] Po-Ching Lin, Ying-Dar Lin, Tsern-Huei Lee, and Yuan-Cheng Lai, “Using String Matching for Deep Packet Inspection,” *Computer*, Vol.41, No.4, pp. 23-28, 2008.

[19] Lei Guo, Yadi Wang, Qing Yao, Shuqiao Chen, and Yuxian Jian, “A Fast Regular Expression Matching algorithm for Deep Packet Inspection,” *Information Theory and Information Security (ICITIS)*, pp. 620-624, 2010.

[20] Sung Jun Ban, Hyeonwoo Cho, ChangWoo Lee, and Sang Woo Kim, “Implementation of IEEE 802.15.4 Packet Analyzer,” *World Academy of Science, Engineering and Technology 35 2007.*

Sardorbek Muminov

He has received the B.S. degree in Electrical Engineering from Ferghana Polytechnic University, Uzbekistan in 2008. He is now Master student of Graduate School of Business IT at Kookmin University, Seoul, Korea. His research interests are embedded systems, routing protocols and underwater MAC protocol.

Nam-Yeol Yun

He has received the B.S. degree in Information & Communication Engineering from Andong National University, Korea, in 2003 and M.S. degree from Kookmin University in 2009. Now he is Ph.D student of Graduate School of Business IT at Kookmin University, Seoul, Korea. His current research interests are embedded systems, USN and underwater MAC protocol.

Soo-Hyun Park

He has 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 had worked as a senior research engineer of the Central R&D Complex, LG Electronics, Ltd from 1990 to 1999 in Anyang, Korea. Now, He is a professor in School of Business IT, Kookmin University, Seoul, Korea. His current research interests include next generation IP-based routing, sensor network and MAC in Ubiquitous Network.