한국정보전자통신기술학회논문지(jkiiect)'18-08, Vol.11 No.4
논문 18-11-04-331 http://dx.doi.org/10.17661/jkiiect.2018.11.4.331

네트워크 본딩 기술을 기반한 IEEE 1588의 고장 허용 기술 연구

무스타파 알타하, 이종명*

Fault Tolerance for IEEE 1588 Based on Network Bonding

Mustafa Altaha ,Jong Myung Rhee*

요 약 IEEE 1588은 측정 및 제어 시스템에서 사용되는 네트워크의 정확한 시각 동기 표준(PTP, Precision Time Protocol)이다. Best Master Clock (BMC) 알고리즘은 PTP에서 최적의 마스터-슬레이브 계층을 선택하기 위해 사용한다. 슬레이브가 마스터와의 링크 장애 또는 현재의 시각 동기 에러가 발생하였을 때, BMC는 자동으로 다른 마스터 신호를 수신할 수 있도록 한다. 이때의 슬레이브 클럭은 마스터 신호의 장애 보상 시간 값에 따라 달라진다. 그러나 BMC 알고리즘에서는 마스터 클럭의 장애 발생에 따른 빠른 고장 복구 방안은 전혀 고려하지 않았다. 이에 본 논문에서는 네트워크 본딩 (Bonding) 기술을 적용하여 마스터 클럭의 장애에 따른 빠른 복구 방안을 제시하였다. 본 연구는 리눅스 시스템의 PTP livery 데몬(Ptpd)과 IEEE 1588의 특정 프로파일을 사용하였으며, 본딩 모드를 통해서 제어하도록 하였다. 네트워크 본딩 기술은 둘 이상의 네트워크 인터페이스 신호를 하나의 네트워크 인터페이스에 전송하기 위해 신호를 결합하는 과정에 대한 것으로, 네트워크의 이중화와 성능 향상을 제공한다. 본딩 기술은 만약 하나의 링크에서 장애가 발생하면, 본딩되어 있는 다른 링크를 통해서 즉각적으로 신호 전달이 가능하기에 네트워크의 이중화 또는 부하 분산 등에 사용한다. IEEE 1588만 적용한 것과 대비하여 IEEE 1588 기술과 네트워크 본딩 기술을 결합한 네트워크 복구 기술의 뛰어난 성능을 본 논문을 통하여 증명하였다.

Abstract The IEEE 1588, commonly known as a precision time protocol (PTP), is a standard for precise clock synchronization that maintains networked measurements and control systems. The best master clock (BMC) algorithm is currently used to establish the master-slave hierarchy for PTP. The BMC allows a slave clock to automatically take over the duties of the master when the slave is disconnected due to a link failure and loses its synchronization; the slave clock depends on a timer to compensate for the failure of the master. However, the BMC algorithm does not provide a fast recovery mechanism in the case of a master failure. In this paper, we propose a technique that combines the IEEE 1588 with network bonding to provide a faster recovery mechanism in the case of a master failure. This technique is implemented by utilizing a pre-existing library PTP daemon (Ptpd) in Linux system, with a specific profile of the IEEE 1588 and it's controlled through bonding modes. Network bonding is a process of combining or joining two or more network interfaces together into a single interface. Network bonding offers performance improvements and redundancy. If one link fails, the other link will work immediately. It can be used in situations where fault tolerance, redundancy, or load balancing networks are needed. The results show combining IEEE 1588 with network bonding enables an incredible shorter recovery time than simply just relying on the IEEE 1588 recovery method alone.

Key Words : precision time protocol (PTP); best master clock (BMC) algorithm; PTP-bonding; IEEE 1588; fast recovery; clock synchronization.

1. Introduction

Over the past two decades, precision clock synchronization has become one of the key elements for Ethernet-based real-time systems. Clock time synchronization is required to maintain high precision for distributed systems in many application domains, such as automation, testing and measurement, and These applications commonly use communication networks that link the distributed network nodes rather than requiring the building of a dedicated synchronization infrastructure [1]. At first, existing clock synchronization protocols for computer networks, such as the network time

*Department of Information and communication Myongji University
**Jong Myong Rhee: Department of Information and Communication Technology at Myongji University jmr77@mju.ac.kr

Received May 02, 2018 Revised May 09, 2018 Accepted Jul 17, 2018

protocol (NTP) [2], have been used for this purpose, but their limitation in accuracy and precision has led to the design of customized protocols that better suit the needs for distributed real-time systems. Nowadays, IEEE 1588 precision time protocol (PTP) [3] is the standard for clock synchronization in various application domains such as industrial automation, telecommunication, and entertainment. The PTP provides a mechanism for synchronizing the clocks of participating nodes in a system to a high degree of accuracy and precision. The PTP defines several different kinds of clocks, as follows [4]:

● Ordinary clock (OC): An OC has only one PTP port. The OC can be the grandmaster (GM) in a system, or it can be a slave clock in the master-slave hierarchy.

● Boundary clock (BC): A BC typically has several PTP ports. It can function both as a master and a slave in the master-slave hierarchy.

● Transparent clock (TC): A TC forwards all PTP messages just as it does in a normal switch or router. The PTP is based on a straightforward master-slave synchronization principle.

Synchronization is achieved by exchanging PTP messages between the master port of a clock and the slave port of another clock, as shown in Fig 1. The messages are divided into event messages and general messages. Event messages are timestamped with both transmission and reception times. General messages do not require accurate timestamps and are used for both synchronization and configuration purposes.

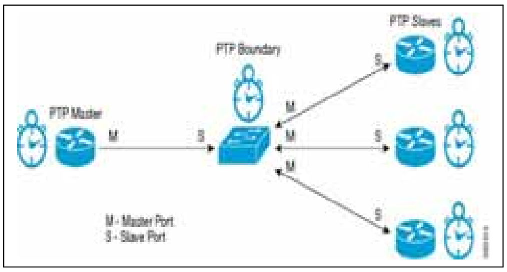

Fig. 1. Example of the PTP master-slave synchronization hierarchy

PTP uses the best master clock (BMC) algorithm to establish the master-slave hierarchy for a network. Only OCs and BCs run the BMC algorithm to build the hierarchy; TCs do not participate in establishing the hierarchy. After an OC runs the BMC algorithm, its port can be in a master state or a slave state. If the port is in a master state, the OC is the GM of the PTP system. Otherwise, the OC will be a slave in the hierarchy. The master-slave hierarchy is mainly established based on BCs in the network. The BMC algorithm divides the clocks in the network into master clocks and slave clocks. Slave clocks synchronize their local clocks with the time of their master clocks. However, the BMC algorithm does not provide a fast recovery mechanism for the master-slave hierarchy in the case of a master failure. The failure of a master (e.g., device failure or link failure) requires the BMC algorithm to re-elect a new master and re-establish the hierarchy [5]. The start of a master election is based on the duration of the election; thus, it requires a specific time span during which the clocks are not synchronized, so they run freely [6]. This drawback causes the loss of reference clock synchronization from the master clock as well as the clock drifting from the clocks during the re-election of the new master, thereby decreasing the network's clock synchronization accuracy.

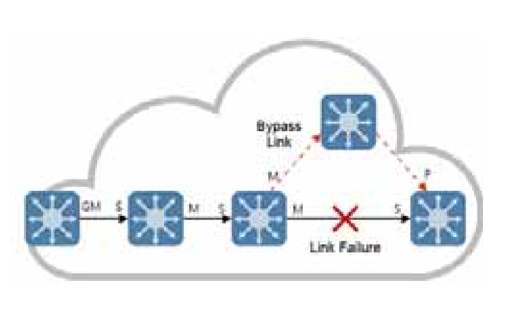

Fig. 2. A sample PTP network

Therefore, we propose a common approach for increasing network resilience and tolerance by combining multiple communication links. In the lower layers, the combination is usually done by interface aggregation, which means grouping multiple physical network interfaces to form a single, logical interface; this is called bonding. Bonding means aggregating several NICs into a group. This group of interfaces appears as a single interface to higher network layers. There are several types of interface aggregations in networks today. On networking devices, this feature is called Ether Channel [7] by Cisco, or Link Aggregation in 802.3ad standard [8] by IEEE.

The rest of this paper is outlined as follows: Section 2 mentions the related works, and Section 3 provides a brief overview on Linux bonding features. In Section 4, after the description of the experiment set-up, we present our results and discussion. Finally, concluding remarks are given in Section 5.

2. Related Works

Several approaches have been proposed to provide fast recovery mechanisms in the event of a link failure for the IEEE1588. The standard PTP can work with various protocols, such as Rapid Spanning Tree Protocol (RSTP), which constructs a synchronization tree instead of PTP and blocks the redundant paths in order to construct a loop-free synchronization topology. Moreover, RSTP detects the failure occurrence in the network after a specific duration, ranging between hundreds of milliseconds and a few seconds [9], and subsequently, RSTP recovers the failure in the network and starts the synchronization mechanism through the standard PTP. Gaderer et al. [10] proposed a democratic approach to enhance the PTP with fault tolerance and overcome the transient deterioration of synchronization accuracy during recovery from a master failure. In this approach, masters are distributed in a group of nodes with sufficiently accurate clocks. The failure of one group member is not a problem because the group can still find a fault-tolerant time value [11]. In the past, network based redundancy architectures such as the High-availability, Seamless Redundancy (HSR) networks have been investigated with respect to transferring time information [12], with the aim of increasing the overall availability of the clock synchronization service.

3. BMC Algorithm

PTP uses the BMC algorithm to establish the master-slave hierarchy for a network, as mentioned above. The BMC algorithm compares datasets describing two clocks to determine which describes the better clock. The BMC algorithm runs locally in OCs and BCs to determine which clock is better. By running the BMC algorithm locally, clocks do not need to negotiate which clock should be the master and which clock should be the slaves. Instead, each clock computes only the state of its own ports

[13]. The BMC algorithm analyzes and compares the contents of the Announce messages received by clocks and datasets associated with the clocks to determine the state of each port of the clocks. Each OC and BC port maintains a separate copy of the PTP state machine. This state machine defines the allowed states of the port and the transition rules between the states. The port states that determine the master-slave hierarchy include the following:

Master: The master port is the source of time on the path served by the port.

Slave: The slave port synchronizes to the clock on the path with the port that is in the master state.

Passive: The passive port is not the master on the path, and it does not synchronize to a master.

The BMC algorithm works based on the data contained in the Announce messages received by a given clock and on the datasets maintained by the local clock. Announce messages provide status and characterization information about the transmitting clock and its GM. The information is used by the receiving node when executing the BMC algorithm. The BMC algorithm selects the BMC by comparing the data contained in the Announce messages received from different ports with the datasets describing the local clock. If the local clock is selected as the BMC, the local clock functions as the GM. If an external clock is selected as the BMC, the local clock traces the master clock. In other words, if the BC sees an Announce message from a better clock, it goes into a slave state, or in the case that it already has a better master, it becomes passive. If the BC does not see an Announce message from a better clock within the Announce Time Out Interval, then it takes over the role of GM. Since the BC has no other port in the slave state, the BMC algorithm updates the port's datasets to the master state configuration and the BC changes to the free-running mode until another master is detected, as shown in Fig. 3.

4. Overview of Network Bonding

Bonding means aggregating several NICs into a group on Linux machines. This group of interfaces appears as a single interface to higher network layers. In the Linux bonding, physical interfaces in the group are called slaves, while the logical interface is called a master. When a packet is sent from a higher layer to the master interface, the bonding driver will deliver this packet to one or more slave interfaces.

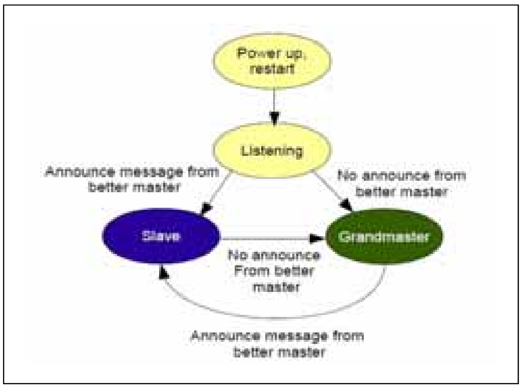
Power up, restart
Listening
Announce message from better master
No announce from better master
Slave
Grandmaster
No announce From better master
Announce message from better master

Fig. 3. BMC algorithm

Packets can be delivered in several ways, depending on which mode the bonding driver is in. The process of receiving packets is similar to the sending processes. When a packet comes to a slave from a remote host, the bonding driver will decide to direct this packet to the master or to drop it, depending on the bonding mode [14]. Bonding modes are as follows:

Round-Robin Mode: In this mode, each packet is sent to one slave interface in turn. For example, in the case of a bonding group with two slave interfaces, the first packet will be sent through slave 1, the second one will be sent through slave 2, and the third one will be sent through slave 1, and so on.

Active-Backup Mode: In this mode, one interface is active at a time. Other slaves are in standby state, and do not send or receive packets. When the active one fails, one backup slave will be chosen as the new active one.

Balance-XOR Mode: This mode generates a simple transmission hash based on the MAC addresses to decide which slave will carry a particular traffic stream. This mode is useful when we want to isolate flows of network packets to clients and distribute them between slave interfaces. This ensures that if a large transmission is initiated by a client, at most, only one slave interface would be occupied in handling it, leaving other interfaces free to handle other flows.

802.3ad Mode: 802.3ad is an IEEE standard for link aggregation [6], which also includes Link Aggregation Control Protocol (LACP) between two networking devices. In this mode, a Linux host can connect to an LACP-enabled switch through a group of aggregated links. The limitation of this mode is that the 802.3ad requires all links running in full duplex mode at the same speed. Outgoing traffic from the Linux host is distributed by the same algorithm as in the Balance-XOR mode.

Broadcast Mode: This mode allows sending of the same data to each bonded interface. This provides a fault tolerance in which all attached switches may receive the same data.

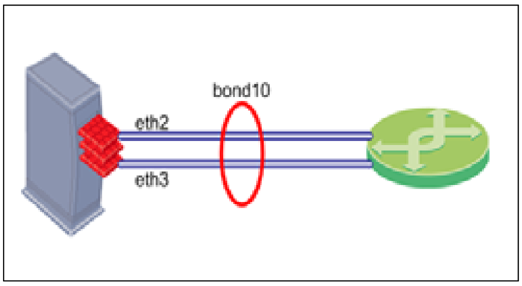

Fig. 4. Link aggregation between a switch and a server

5. Evaluation Method

In this section, we present the evaluation of four bonding modes: round-robin, active-backup, broadcast, and 802.3ad. These modes are chosen because they perform well for two major criteria on which we want to focus: fault tolerance, and load balancing for the IEEE 1588. This implementation will utilize an existing open source module called ""Ptpd"" [15]. A specific profile of the IEEE 1588 V2 will be used, according to our module. A Gns3 [16] real time simulator is utilized to conduct the experiments.

5.1 Experiment Set-Up

Five simulations were conducted, one being the control simulation with an un-bonded connection, and four testing each of the previously mentioned bonding processes. These simulations were conducted ten times each using the Gns3. The experiment utilized two Linux hosts whose system parameters are listed in Table 1. The two hosts have several network cards with the same negotiated speed of 1000 Mbps working as BC: one is connected to the OC-GM and the other connected to OC-slave . The two BC'-s are connected through a TC network. This topology allows us to conveniently monitor how traffic is distributed among the links by using the capture filter, Wireshark [17].

Table 1. Testing Systems

System Parameters	Settings
CPU	Intel Core i7
Memory	1024GB
Operating System	Ubuntu
Kernel Version	3.11.0 - 14
Network Adapter	7.3.21 - k8 - NAPI

The control simulation run the BMC IEEE 1588 as a recovery mechanism. Furthermore, the Sync message rate is assumed to be one message-per-second and the Announce message rate is 0.5 messages-per-second, which was defined in the ITU-T G.8275.2 standard profile [18]. Therefore, the Sync-Interval will be determined as 1 seconds and the Announce-Interval will be 2 seconds. The Announce Time Out Interval, will be 12 seconds according to the default BMC.

The bonded simulations all demonstrate BC-to-BC bonding, as illustrated in Fig. 5a. The control simulation is shown in Fig. 5b, in which the same BC'-s are used without bonding. We focus on evaluating how fast this bonding mode can switch traffic to a backup link when the active link fails. For this purpose, we measure recovery time, which is defined as the duration of PTP flow interruption due to a network failure, to assess fault tolerance capability. Recovery time is determined by monitoring the time between the last PTP message of an old active slave, just before link failure, and the time of the first PTP message arrives on the new active slave.

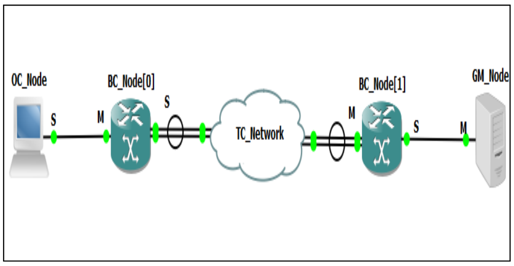

Fig. 5a. Bonded network topology

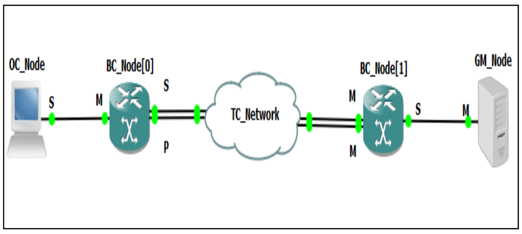

Fig. 5b. Control simulation topology

5.2 Simulation Results

This research focuses on fault tolerance capability. As seen in Fig. 6, the recovery time differences of the five different simulations is expressed in seconds. The control simulation, PTP standard without bonding (PTP-BMC), was significantly slower in its recovery time than the PTP with bonded modes, with an average recovery time of 11.91 seconds. The slowest PTP with bonding mode was the active-backup mode (PTP-ab), which resulted in an average recovery time of 1.05 seconds, 11.91 seconds faster than the standard (PTP-BMC). The round-robin mode (PTP-rr) averaged 0.71 seconds. The 802.3ad mode (PTP-e) and the balance-XOR mode (PTP-XOR) both results around 0.64 seconds. PTP with broadcast bonding mode (PTP-b) resulted in an almost indistinguishable interruption and therefore is represented as an average of 0 seconds.

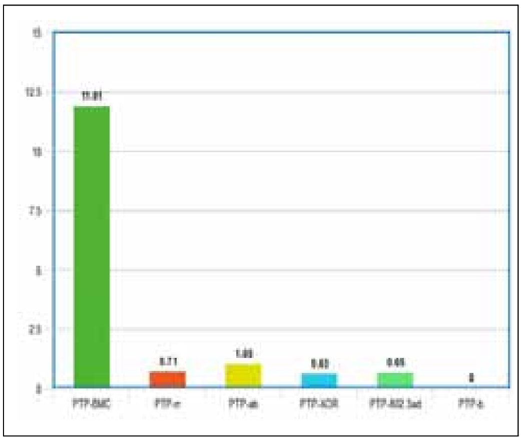

Fig. 6. Recovery time

The results of the simulations show that the total delay duration was reduced by about 91%-100% of the standard PTP, because the standard PTP depends on the BMC as the recovery mechanism.

5.3 Discussion

The PTP standard simulation took a further 11 seconds to recover in comparison with the bonded modes. As a result, the slave clock of the PTP standard drifted from the master clock timing, losing accuracy and efficiency. In contrast, the bonded modes with the PTP (PTP-rr,PTP-ab,PTP-XOR,PTP-802.3ad, and PTP-b), recovered quickly; the slave clock drifted only slightly from the master clock, indicative of better accuracy and efficiency. The simulation results show that the bonding approach significantly reduced the local clock drifting of the slave system due to the 91%-100% decrease in recovery time.

6. Conclusion

In this paper, we have presented the performance evaluation of the five typical Linux bonding modes, round-robin, active-backup, broadcast, balance-XOR, and 802.3ad modes, for fault tolerance with the IEEE 1588. The active-backup mode with two bonded NICs can provide fault tolerance recovery time faster than in the unbonded case that simply relied on the BMC algorithm. As previously mentioned, the IEEE 1588 dictates the use of a timer in the event of a master failure. As demonstrated by this body of research, this leads to a loss of synchronization due to the need for re-establishing BMC hierarchy through election and use of a timer, which wastes time. To prevent an interruption of network synchronization, bonding should be used to lower recovery time. The simulation results showed that the PTP-bonding approach reduced the total delay detection and recovery duration by 91%-100% in comparison with the standard PTP. Therefore, PTP-bonding improves the reliability and the availability of the PTP network; it enhances the network performance and the synchronization accuracy of PTP clocks.

Our future work will involve developing a BMC algorithm that can utilize bonding in the case of having a different master. We hope to create a bonded network that can tolerate a master failure, not just a link failure, by using the proper bonding mode with the developed BMC algorithm.

REFERENCES

[1] IETF Standard: Network Time Protocol Version 4: Protocol and Algorithms Specification. RFC 5905. Available online: https://tools.ietf.org/html/rfc5905 (accessed on 30 May 2017).

[2] D. L. Mills, Computer Network Time Synchronization : The Network Time Protocol on Earth and in Space ,2010.

[3] IEEE, IEEE Standard for a Precision Clock Synchronization Protocol for Networked

Measurement and Control Systems,Sep. 2008.

[4] J. C. Eidson, Measurement, Control, and Communication Using IEEE 1588. New York, NY, USA: Springer, 2006

[5] IEEE Instrument and Measurement Society. IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems. IEEE 1588-2008 Standard. Available online: https://standards.ieee.org/findstds/standard/1588-2008.html.

[6] IEEE Standard for A Precision Clock Synchronization Protocol for Networked Measurement and Control Systems; IEEE Std 1588-2002; IEEE: Piscataway, NJ, USA,pp. i-144, 2002.

[7] Cisco EtherChannel - White Paper (accessed January 18, 2014), http://www.cisco.com/en/US/tech/tk389/tk213/tech white papers list.html.

[8] "IEEE 802.3ad Link Bundling". Cisco Systems. 2007-02-27. Retrieved 2012-03-15

[9] Ferrant, Jean-Loup, et al. Synchronous Ethernet and IEEE 1588 in Telecoms: Next Generation Synchronization Networks . John Wiley &Sons, 2013.

[10] Gaderer, G.; Rinaldi, S.; Kero, N. Master Failures in the Precision Time Protocol. In Proceedings of the International IEEE Synposium on Precision Clock Synchronization for Measurement, Control and Communication (ISPCS), Ann Arbor, MI, USA, 22-26 September 2008.

[11] IEEE Standard for Local and Metropolitan Area Networks: Media Access Control (MAC) Bridges; IEEE Std 802.1D-2004 (Revision of IEEE Std 802.1D-1998); IEEE: Piscataway, NJ, USA, pp. 1-277, 2004.

[12] De Dominicis, C.M., Ferrari, P., Flammini, A., Rinaldi, S. and Quarantelli, M., 2011. On the use of IEEE 1588 in existing IEC 61850-based SASs: Current behavior and future challenges. IEEE transactions on instrumentation and measurement, pp.3070 - 3081, Sep, 2011.

[13] Guijarro, Manuel; Ruben Gaspar; et al. (2008). "Experience and Lessons learnt from running High Availability Databases on Network Attached Storage" (PDF). Journal of Physics: Conference Series. Conference Series .vol .119,No.4, p. 042015,2008.

[14] S, Kim JO, Davis P, Yamaguchi A, Obana S. Evaluation of linux bonding features. In Communication Technology, ICCT' 06. International Conference on IEEE, pp. 1-6, Nov ,2006 .

[15] Correll K, Barendt N, Branicky M. Design considerations for software only implementations of the IEEE 1588 precision time protocol. In Conference on IEEE, pp. 11-15,Oct , 2005 .

[16] Simulator, Graphical Network. "GNS 3." (1989).

[17] Combs,G., et al.:The Wireshark Network Protocol Analyzer,. Ghttp://www.wireshark.org.

[18] Precision Time Protocol Telecom Profile for Time/Phase Synchronization with Partial Timing Support From the Network; ITU-TG.8275.2 Recommendation; International Telecommunication Union/ITU Telcommunication Sector: Geneva, Switzerland, pp. 1-46, 2016.

저자약력

무스타파 알타하(Mustafa Altaha)

- Born on February 7, 1992. Graduated from AI MansourUniversity College in 2014.In 2014-2015 2014 - 2015 EarthlinkCo. for internet service provision ISP as Network operation systems. then start Master program at Myongji University from 2015-2017 at Department of Information and Communication Engineering. From 2017 start my, Ph.D ,myongji unversity too.

이종명(Jong Myung Rhee)

- JongMyung Rhee received his PhD from North Carolina State University, USA, in 1987. After 20 years at the Agency for Defense Development in Korea, where he made noteworthy contributions to C4I and military satellite communications, he joined DACOM and Hanaro Telecom in 1997 and 1999, respectively. At Hanaro Telecom, which was the second largest local carrier in Korea, he served as Chief Technology Officer (CTO), with a senior executive vice-president position. His main duty at Hanaro Telecom was a combination of management and new technology development for high-speed Internet, VoIP, and IPTV. In 2006, he joined Myongji University and is currently a full professor in the Information and Communication Engineering Department. His recent research interests are centered on military communications and smart grid, including ad-hoc and fault-tolerant networks. He is a member of IEEE and IEICE.