Fault Detection and Diagnosis in IP-Based Mission Critical Industrial Process Control Networks

Young J. Won, Mi-Jung Choi, and James Won-Ki Hong, POSTECH
Myung-Sup Kim, Korea University
Hwawon Hwang, Jun-Hyub Lee, and Sung-Gyoo Lee, POSCO

ABSTRACT

Mission-critical industrial process control networks support secure and reliable communications of devices in a controlling or manufacturing environment. They used to mostly use proprietary protocols and networks. Recently, however, many of them are being migrated to IP-based networks to consolidate many different types of networks into a single common network to simplify network operation, administration, and maintenance, and reduce operational expenses and capital expenditures. Despite their wide deployment, most operators have very little knowledge on how to operate them reliably and securely. This is mainly due to the operators' unfamiliarity with various faults that occur on IP-based process control networks. The current process of detecting and diagnosing faults in process control networks is mostly manual and thus the operators detect the problems only after noticeable process malfunctions. This article presents an overview of industrial process control networks and discusses the issues of introducing IP technologies into them. We then propose a fault detection and diagnosis method which is suitable for IP-based process control networks. We also present the system architecture and implementation of fault detection and diagnosis system as well as its deployment at POSCO. Finally, based on operational experience, we have generated a failure prediction model that can be used to predict potential alarms.

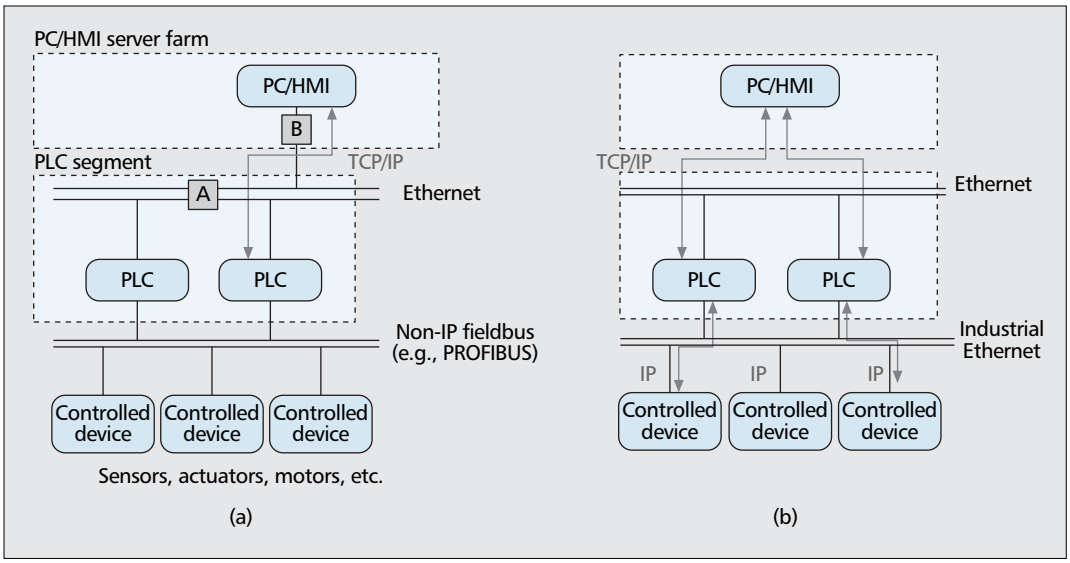
INTRODUCTION

The communication in a complex automated manufacturing system, typically conveying belt lines in a factory, has been heavily dependent on proprietary protocols from various vendor-specific technologies. Integration of the devices and programming tools from multiple vendors in a single network was almost impossible prior to the introduction of IP technologies; in fact, the total solution from one single vendor was preferable for network designers and maintenance personnel to handle the process control network requirements. Traditional process control network technologies (e.g., FOUNDATION Fieldbus [3], PROFIBUS [4], MODBUS [5], BACnet [6]) were developed separately from the relatively recent emergence of Ethernet and IP-based network technologies. Recently, newer versions of these technologies have adopted Ethernet (e.g., Industrial Ethernet) and IP for low cost, high scalability, interoperability, and easy maintenance. Their place in the market is still limited because the decision to move toward an all-IP environment in a manufacturing environment is beyond the technical superiority of Ethernet and IP. Replacing existing devices, including numerous low-end controlled machineries (sensors, actuators, motors, etc.), requires a huge investment.

Some researchers have discussed important issues in moving toward IP-based control networks [1, 2]. The roles and environments of process control networks are distinguishable depending on the final products; however, they are all deployed in mission-critical operations. Thus, minimum delay and guaranteed transmission are particularly important quality of service (QoS) attributes for every existing communication session in process control networks [7]. Second, while we are in the process of moving slowly toward IP-based control networks, we encounter problems reflecting vulnerabilities of IP networks in general. The majority of current process control networks are mixtures of IP-based and non-IP-based control technologies, so different monitoring and maintenance techniques should be used [2].

This article investigates operational and maintenance needs for real-world process control networks in POSCO, the world's third largest iron and steel manufacturer. Their single operational site consists of more than 40 manufacturing plants, and each plant has its own process control network. We have selected a few process control plants where they are particularly vulnerable to network outage and have previ-

0163-6804/08/$25.00 © 2008 IEEE



The process control network components are organized in a hierarchical fashion where the controller at the top triggers corresponding actions in one or more controlled devices.

Figure 1. *Simplified view of a) typical; b) all-IP-based process control networks.*

ously experienced unstable network conditions. By identifying the network trouble cases in the real-world process control networks, we summarize operation, administration, and maintenance (OA&M) considerations for possible network failures when running process control networks, and propose a remote fault detection and diagnosis system to replace the conventional IP network monitoring tools.

The organization of this article is as follows. An overview of process control networks and our motivation are given. We identify the communication fault cases and present a control network diagnosis system for failure cause analysis. We provide an alarm pattern analysis based on the real-world deployments and a failure prediction model that can be used to predict potential alarms. Finally, concluding remarks are given and possible future work is discussed.

Overview of Process Control Networks

In this section we describe process control network components and some issues of adopting IP technologies. The motivation for fault analysis in IP-based process control networks is also given.

Toward IP-Based Process Control Networks

Typical process control networks are illustrated in Fig. 1. The process control network components are organized in a hierarchical fashion where the controller at the top triggers corresponding actions in one or more controlled devices. The following are brief descriptions of each element and its role:

- Process controller (PC): This is a part of the software and hardware package provided by programmable logic controller (PLC) vendors. It is the process control software on a computer running UNIX or Windows that can remotely access PLCs. Custom-built or vendor-provided server applications are placed in a PC to communicate with PLCs. It also communicates with the machines running human-machine interface (HMI) solutions that provide graphical presentation of real-time process status monitoring.
- Programmable logic controller (PLC): It is a microprocessor computer for process control attached to a process control network. Complex sequence control of machinery (or low-end controlled devices on factory assembly lines) is handled by custom-built software programs running in the PLC.
- Controlled devices: These are sensors, actuators, motors, and so on. They receive command signals from PLCs via the embedded interface and perform various tasks.

Since the late 1980s, the communications between these three entities have been handled by Fieldbus technology [3], which was suitable for control in a distributed environment. It was more of a vendor-driven technology and relied on proprietary protocols. The widely known PLC manufacturers (e.g., Siemens, ABB Ltd., Mitsubishi Electric) have been actively involved in deploying their choice of Fieldbus technologies in their solutions, which left little room for actual users like POSCO to customize their purchase for their own operational and maintenance needs. A few problems have been observed when maintaining vendor-specific solutions: high hardware cost (e.g., EIA-485 network interface), PC and PLC programming difficulty, and a lack of experienced engineers. These problems occur because the details of PLC technology were undisclosed to users, and its underlying mechanisms for communication are difficult to understand.

The difficulty arises when merging multivendor devices into a single network. Communication between different vendors' products was unreliable; consequently, the Ethernet and IP technologies were considered as an alternative to offer more flexibility, scalability, and lower cost in designing process control networks. Communication over IP networks was more convenient in terms of understanding the communication

It is very crucial for the engineers to detect early symptoms for unstable communications to controlled devices, analyze the cause, and take an appropriate countermeasure (such as replacing problematic cables or parts).

mechanisms and network programming for the engineers.

Figure 1a shows the current status of process control networks, which is a combination of IP-based and non-IP-based control technologies. Figure 1b illustrates the future of an all-IP-based process control network environment where the communications between the PLC and controlled devices are established over Industrial Ethernet and IP. Adapting to such a scheme requires more IP addresses for every controlled device and reprogramming of existing PLCs. Yet most companies are reluctant to pursue this newer architectural model due to uncertainty about the stability and security of IP communication in a harsh manufacturing environment. This article focuses on a wide deployment of choice in industrial process control networks as illustrated in Fig. 1a.

Motivation for Remote Fault Detection and Analysis

Network communication failures at any stage of process control networks can be fatal. Since all machineries operate in a synchronous manner, a single communication failure to a device may delay or even force a shutdown of the entire plant. Thus, it is much less fault-tolerant than typical IP networks. For instance, a steel manufacturing plant involves a series of processes that are dependent on one another for the final product. It is very crucial for the engineers to detect early symptoms of unstable communications to controlled devices, analyze the cause, and take an appropriate countermeasure (e.g., replacing problematic cables or parts).

Two monitoring domains exist in process control networks: PC to PLC network and PLC to controlled device network. We focus on identifying the problems in the first half of control networks, which is an Ethernet and IP-based network. Although many IP network diagnosis tools (e.g., Sniffer, Wireshark) are available, they often cannot detect control network failure cases due to a significant difference of traffic nature as well as distinct communication failure characteristics. Because of the obscure failure patterns, network administrators apply these tools after communication failures are reported by end users and then wait for the same problem to occur. Another issue is that these diagnosis tools require on-site monitoring of links. It is more convenient to have remote access to a monitoring system that can monitor multiple control networks. However, no IP network diagnosis tools have satisfyingly supported process control network specifics. Thus, we investigate what needs to be monitored in the traffic and the possible IP communication failure symptoms, and develop a process control network monitoring system that overcomes the deficiencies of existing IP monitoring tools as a solution.

Fault Detection and Diagnosis

In this section we present real-world fault cases in process control networks. By analyzing previous communication failure cases, we categorize specific process control network failures. We also provide the design and implementation details of a fault detection and diagnosis system for process control networks.

Real-World Fault Cases

We have analyzed the log history of process control network fault cases reported by network administrators at POSCO over the past several years. Note that these cases have been identified intuitively by the network administrators after the communication failures occurred. The existing IP network diagnosis tools (e.g., Sniffer, Wireshark) do not yet have the capability to understand the failures in process control networks and provide the causes of such failures. The possible fault cases are as follows.

Ethernet duplex mismatch: In the auto-configuration enabled environment, two end Ethernet devices may disagree about their duplex (half or full) settings after negotiation. Mismatch can increase the frame loss rate and add extra delays due to collision frames. This problem is not much of an issue in IP data networks, which consist of high-end computers and switching devices. While some embedded interface modules in PLCs and controlled devices are incomplete against the up-to-date Ethernet standard, the duplex negotiation in control networks often assigns an incorrect setting. A manual setting for every link is a time consuming procedure, but has been the only solution available to administrators thus far.

PLC programming bugs: Poor PLC programming causes communication failures due to the lack of understanding and experience in socket programming. There have been a number of reports that might infer PLC programming bugs, such as unexpected packet occurrences (e.g., irregular keep-alive packets), disordered packet sequences, and unusual TCP window size. Reprogramming at either or both of the server and client applications is needed to fix the problem.

Device driver bugs: Device drivers in PLCs and controlled devices face an interoperability problem between various devices from different vendors. The robustness of a device is also crucial for continuous operation. For example, if out-of-range signals in the physical link are received, it may force shutdown of the devices. It is difficult to manipulate vendor-specific drivers in embedded hardware, so replacement with a newer part for the device is needed.

Link corruptions: These refer to physical damage to cables, such as a cable cut or dust on the fiber interface. There are more chances of these incidents because process control networks are typically located in a hostile environment such as a factory. Detecting the damage is also challenging. We can only speculate cable damage from measuring a few network metrics: invalid frame size, frame collision, cyclic redundancy check (CRC) error, inconsistent throughput, and so on.

Protocol unawareness: If a particular device encounters unrecognizable protocols, it malfunctions and most likely causes a stoppage. It is important to prevent unsupported protocol traffic, which is indicated in the vendor's manual, from floating in the control networks. For example, when Simple Network Management Proto-

col (SNMP) traffic was received in an old PLC, the system went down and had to reboot for recovery.

Packet overflooding: The bandwidth is occupied by unwanted traffic, such as that caused by Internet worms. For example, an excessive amount of ARP packets, a so called ARP storm, was observed in process control networks. The required bandwidth for PLC communication could not be provided by the network.

Electrical noise: Unstable transmission occurs due to signal interference. Network links, especially coaxial cables, and devices (nearby high voltage machinery) can be interfered with by the noise.

Power outage: Power supply of devices is malfunctioning. This problem is related to harsh conditions in process control networks, such as heat and moisture conditions.

Misconfiguration: Traffic is routed in nonoptimal paths or even stays in a loop due to incorrect routing table entries. This results in packet delays and loss.

Damage to router/switch interface: This is a typical hardware failure. It is somewhat difficult to detect because the entire connection to a certain area of a network can be lost simultaneously.

Based on the above real-world communication failure cases, we have categorized control network failure types as communication errors, network misconfiguration, physical defects, and software defects. Although all these failure types can also occur in ordinary IP networks, particularly the last type (i.e., software defects), they are more common in process control networks. In fact, failures of the last type are due to the fundamental difference of network components (i.e., PLC vs. general computer).

Measurement Metrics and Alarm Conditions

It is important for network administrators to recognize early symptoms of process network communication failures. We need to revisit some IP network metrics that have been overlooked for some time because we rarely notice them anymore in today's IP data communication networks. Thus, we have selected several IP network oriented metrics and identified the alarm conditions (thresholds) that best reflect any irregularity of communication in process control networks: collision/jumbo/runt/error/drop/fragment frames, checksum errors, retransmission, out-of-sequence, throughput variation, window size error, and more (Table 1). The metrics themselves are not very unique, but they have not been properly analyzed in many IP network diagnosis tools. A new set of monitoring categories and conditions is necessary because even a popular tool, like NG Sniffer [8], does not fully detect control network specific failures but generate false network alerts. The selected metrics can be measured using passive monitoring techniques that do not interfere with network operations.

System Architecture and Implementation

We provide a design of a ubiquitous fault detection and diagnosis system for process control networks. The proposed system supports network diagnosis capability and remote accessibility. Figure 2a illustrates the design of process control network traffic monitoring system. It consists of one or more monitoring probes, collectors (DB servers), and a presenter (Web user interface). As illustrated in Fig. 2b, its distributed architecture allows us to design a flexible monitoring system where multiple links can be monitored with a single point of data representation [9].

The monitoring probe is attached to the target link and controlled remotely from the Web server running in the presenter. Packet traces are forwarded to the probe via port mirroring or passive signal tapping. The packet collector captures packets from the interface and passes them on to the flow generator and packet log. The fault analyzer records the list of IP network oriented metrics mentioned earlier and runs the validation check against the thresholds predefined by the network administrators. Note that the method of choosing the appropriate threshold values and the evaluation of the selected values are presented in the upcoming section. If any of the metrics violates the defined alarm conditions, the alarm generator sends an email or SMS message. The metrics should be monitored for each flow. *Flow* refers to a bidirectional packet stream that shares the same header information: a pair of MAC addresses, IP addresses, source/destination ports, and protocol.

Binary packet log files are stored for post in-depth analysis if necessary. In fact, the I/O overhead for storing full packet traces is acceptable in a process control network environment because of very low bandwidth [10]. The packet cecoder converts binary packets into Packet Details Markup Language (PDML) upon user request. At last, the network administrator can attach the explanation to the alarm log. It is more of an intuitive opinion about cause analysis. This information is later referenced to find a mapping relationship between network metrics and previously known fault cases.

For now, all the components are implemented in a single machine. Our implemented system satisfies the following functional requirements: traffic summary statistics analysis, packet decoding display, and alarm statistics analysis. Granting remote accessibility via the Web browser allows one or more network administrators to be ubiquitously aware of the network status faster and more conveniently. In the main user interface, it displays the bandwidth, alarm occurring flows, alarm occurring types, protocol distribution, and so on, which reflect the overall status of the monitoring control networks. This information is updated every 20 s with a delay of about 40 s from the current time. Flow records show the cumulative sum of traffic summary from the flow's start until its termination. The administrators are able to access each flow record even after its termination due to an active flow timeout of 30 s. Note that the reported alarms are not always related to the actual failure of the network; rather, they are early symptoms of possible network failures. Table 2 illustrates the functionality comparison of the proposed system and Sniffer.

There is hardly a case where a single metric infers the cause of failure. They occur in a chain

A new set of monitoring categories and conditions is necessary because even a popular tool, like NG Sniffer, does not fully detect the control network specific failures but generate false network alerts.

Index	Network metrics	Alarm conditions
1	Collision frames	First appearance, or threshold-based
2	Jumbo (≥ 1514 bytes) frames	First appearance
3	Runts (≤ 64 bytes) frames	First appearance
4	CRC error frames	First appearance
5	IP/TCP checksum errors	First appearance
6	Fragment packets	Threshold-based
7	Retransmission packets	First appearance, or threshold-based
8	Packet interarrival time (ms)	Increase to the previous value
9	Throughput (b/s)	Decrease, drop to 0, or pattern analysis over monitoring period
10	Packets per second (or packet burst)	Increase, decrease, drop to 0, or pattern analysis over monitoring period
11	Min/max/diff packet size (bytes)	Change in difference of max and min sizes over monitoring period
12	Min/max/diff TCP window size	Drop to 0, change in difference of max and min sizes over monitoring period
13	Out-of-order sequence packets	First appearance
14	Broadcast packets	Threshold-based
15	Unsupported protocol packets	Threshold-based

Table 1. *Control network specific measurement metrics and alarm conditions.*

of reaction; for example, if error frames or out-of-order packets arrive, there is a high chance of seeing retransmission trials soon after. Consequently, the whole sequence of transmissions might have been triggered from the cable cut or software bugs. The correlation coefficients are calculated for every combination of the proposed network metrics to determine their closeness in each fault case. If a certain group of metrics with strong association is repeatedly observed a reasonable number of times, we can speculate the cause (from the list of fault cases) of a similar traffic pattern for future events. Finally, the user interpretation component represents the level of confidence of the network administrators in reasoning the fault cause. The network administrator can rate each alarm for its accuracy based on their intuition.

Each fault case is represented by the group of metrics and their aggregated values from the previous analysis results. This set of data is referred to as the knowledge base for cause analysis. Since the alarm condition is also based on the values of the proposed IP metrics, we cross-match a stream of alarms with the available set of failure cases that have previously been observed in the network. The result of the cross-match can be reviewed by the network administrator to reduce any false positive in assigning the possible cause. Finally, the identified alarms are fed back to the knowledge base for case updates to be consistent with the nature of the traffic.

We designed the cause analysis procedure for offline use because we needed a sufficient number of network failure reports. The alarms relying on the measurement results cannot suggest appropriate countermeasures to fix the problem, but identify the troubled links or devices (e.g., IP address) that are suffering from the communication failures. faster and accurate cause identification is more important than just alarm generation in real-world network management and brings us one step closer to fault handling automation.

Prediction of Alarm Patterns

In this section we present alarm pattern and prediction analysis results based on real-world deployments. Our fault detection and diagnosis system has been deployed at two process control networks to monitor and detect faults at POSCO since June 2007. Our system also collects and logs traffic traces for close examination of packets in the event of fault detection. A complete isolation of the PLC network from Internet or any other enterprise networks is guaranteed by assigning private IP addresses and physically detached networks to PLC segments. We analyzed a week-long trace at one of the edge segments and a typical work-hour trace at the plant backbone traffic traces (segments A and B in Fig. 1a, respectively). Segment A refers to the top of a local PLC network, and segment B is the collection point where the multiple instances of segment A traffic

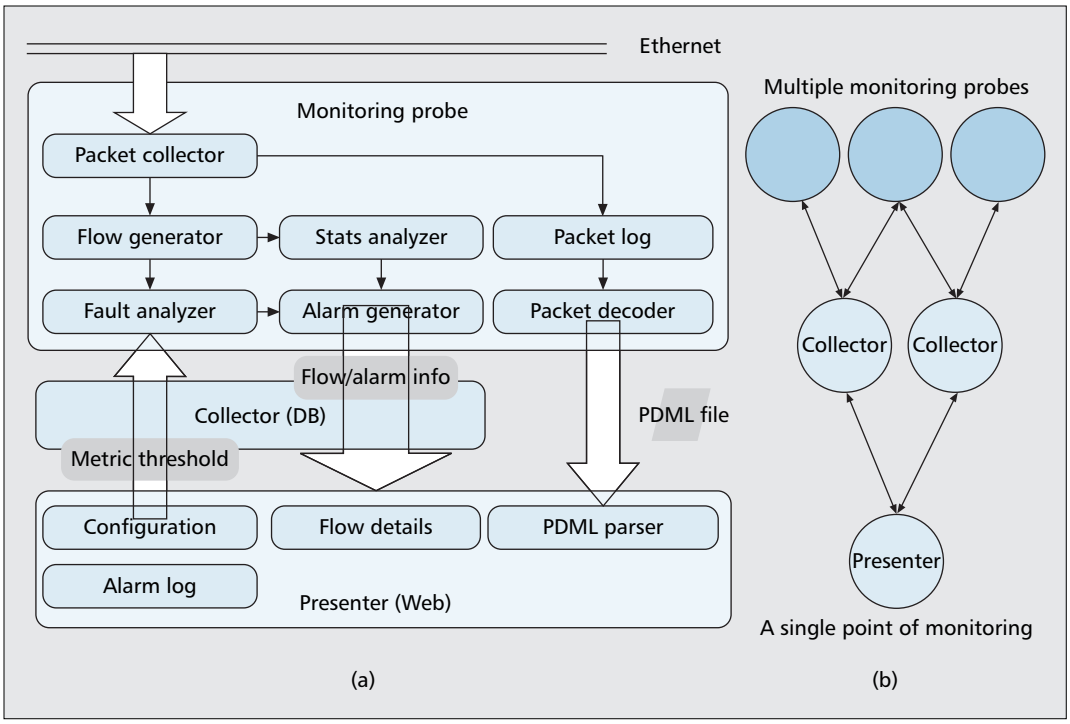


Figure 2. *Design of the control network diagnosis system: a) system architecture; b) abstraction.*

Our fault detection and diagnosis system has been deployed at two process control networks to monitor and detect faults at POSCO since June 2007. Our system also collects and logs traffic traces for close examination of packets in the event of fault detection.

	Proposed system	Sniffer
Accessibility	Remote (Web-based)	On-site only
Applicability	Multiple link monitoring	A single link monitoring
Hardware flexibility	A single or multiple hardware platform for distributed architecture	A single hardware platform
Alarm definition	A new category of alarms can be added	Fixed
Online packet logging	Lossless packet capture	Heavy overhead, occasional packet drops
Online analysis	Yes	Yes
Packet decoding capability	Yes	Yes

Table 2. *Comparison of the proposed system and Sniffer [11].*

are directed. The actual name of each process segment is undisclosed due to security considerations.

The system has reported a few categories of alarms since its deployment. In particular, we have observed that the following three alarm categories mainly appeared: window size error, out-of-sequence packet counts, and retransmission packet counts. We use the two-parameter Weibull distribution [11] to determine appropriate threshold values and analyze the alarm cycle of our system over the continuous monitoring period. It is a probability measure on each alarm type over time based on the empirical data set of the alarm log.

Alarm ratios can be manipulated by user specified threshold values; thus, it is important for the proposed system to determine appropriate threshold values while preserving the alarm credibility. Table 3 illustrates the Weibull distribution fit values for each alarm case and its choice of threshold values. The fit to the empirical cumulative distribution function (CDF) distribution is given by a Weibull CDF distribution and reliability:

$$F(t) = 1 - e^{-\left(\frac{t}{\eta}\right)^{\beta}} \quad (1)$$

$$R(t) = 1 - F(t) \quad (2)$$

where t = time, η = scale parameter, and β = shape parameter (slope).

We focus on the segment A traces for the alarm analysis of the proposed system. A longer trace is preferable to handle the process control network specifics in general. Figures 3a and 3b

illustrate the probability measure of the window size error alarms. We are able to forecast the expected probability of window size error and provide a unique cycle of alarm. The time granularity of measurement is 1 min. The threshold value for a window size alarm report is equal to 1 because a single occurrence of such an abnormality has great potential to cause communication failures.

Figures 3c and 3d illustrate the out-of-sequence alarm probability model for thresholds 10 and 50, respectively. The alarm probability model refers to the probability of the alarm occurrences with the given threshold values over time. Since the probability model for threshold 100 shows an almost prefect fit to Fig. 3d, we have omitted the corresponding graph. Table 3 shows that these two sets of parameter values are very close to each other (1.6234 → 1.6244, 6957.8985 → 6959.1457). The curve in Fig. 3c is slightly steeper than that in Fig. 3d, meaning a bit higher alarm ratio. However, the difference here is minimal. We can conclude that after the certain threshold value, 50, the alarm ratio remains steady. Figures 3e and 3f illustrate the retransmission alarm probability models for thresholds 100 and 500, respectively. The last threshold value of 1000 also shows an almost perfect fit to Fig. 3f, in which a significant reduction of alarm generation from that in Fig. 3e is observed.

Alarm category	Threshold value	Segment A		Segment B	
		β	η	β	η
Window size error 1	1	1.4107	6627.8192	2.1505	372.6533
Out of sequence 10	10	1.5938	6941.4623	2.9220	313.0396
	50	1.6234	6957.8985	2.9220	313.0396
	100	1.6244	6959.1457	2.9220	313.0396
Retransmission 100	100	1.1852	6228.7397	2.9836	319.2062
	500	3.1364	276.5284	255.3608	264.4731
	1000	3.1567	277.3797	255.3608	264.4731

Table 3. *Parameters to fit for each alarm probability.*

Segment A	Window error	Out of sequence	Retransmission
Minutes	Probability of alarm (%)	Probability of alarm (%)	Probability of alarm (%)
1440 (1 day)	10.95	7.45	100
2880 (2 days)	26.54	21.24	100
4320 (3 days)	42.11	36.95	100
5760 (4 days)	55.97	52.09	100
7200 (5 days)	67.49	65.25	100
8640 (6 days)	76.62	75.85	100
10080 (7 days)	83.58	83.88	100

Table 4. *Probability of alarm report.*

Based on the analysis results shown above, we have derived a failure probability model that can be bound to threshold values of 50 and 500, out-of-sequence and retransmission, respectively. These values are set to generate the steady alarm ratios for segment A. When applying these threshold values for segment A, the networks are likely to experience the alarm occurrence trails in Table 4, which illustrates the probability measure of alarm reports. The first two categories show that the probability of occurrence of window error and out-of-sequence alarms increases as the monitoring period advances. Retransmission occurs continuously with 100 percent probability regardless of time because it was continuously observed throughout the monitoring period

We have determined appropriate threshold values for alarm generation in this particular instance. In a similar fashion we can validate the effectiveness of determined threshold values for different networks and build a trajectory model of alarm generations.

Concluding Remarks

The area of monitoring and analyzing industrial process control IP networks has been blinded in the research community thus far. Process control networks are much more vulnerable in terms of network outages because the consequences can be very costly. An active diagnosis of potential communication failures is crucial to maintain a fault-tolerant network operating environment. Our contributions presented in this article are to:

- Introduce IP-based process control networks and their future needs
- Identify process control network specific failure types
- Design and implement a remote fault detection and diagnosis system for process control networks
- Present the analysis results of alarm pattern prediction

For future work, we plan to establish the prediction model for actual network failures by matching the generated alarms with real network outage cases. A comparison study between process control networks and ordinary IP networks will be also helpful to identify and formulate the systematic differences of process control networks.

Acknowledgments

This research was supported in part by POSCO, by the Ministry of Knowledge Economy, Korea, under the Information Technology Research Center support program supervised by the Institute for Information Technology Advancement (IITA-2008-C1090-0801-0045), and the Electrical and Computer Engineering Division at POSTECH under the BK21 program of the Ministry of Education, Korea.

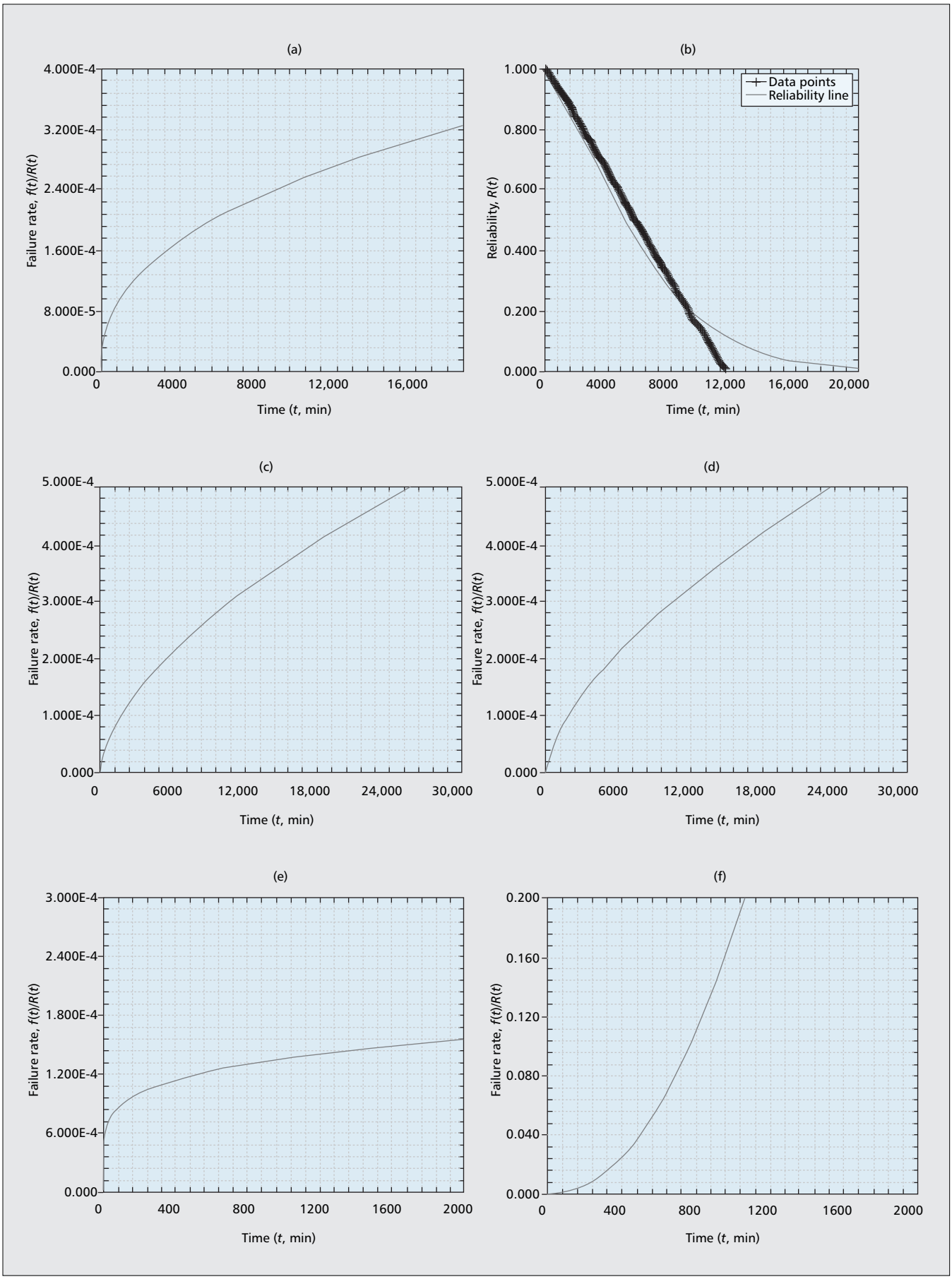


Figure 3. *Segment A, window size error: a) failure (alarm) probability; b) reliability fit to the empirical complementary CDF; out of sequence: failure probability for threshold c) 10 –; d) 50; retransmission: failure probability for threshold e) 100 –; f) 500 –.*

REFERENCES

[1] N. Okabe, "Issues of Control Networks When Introducing IP," *Proc. Symp. Applications and the Internet Wksp.*, vol. 00, 2005, pp. 80–83.
[2] F.-L. Lian, J. R. Moyne, and D. M. Tilbury, "Performance Evaluation of Control Networks: Ethernet, ControlNet, and DeviceNet," *IEEE Control Sys.*., vol. 117, no. 6, 2001, pp. 641–47.
[3] Fieldbus Foundation, FF-581-1.3, "FOUNDATION Specification: System Architecture," 2003.
[4] PROFIBUS International, IEC 61158, "Digital Data communication for Measurement and Control — Fieldbus for Use in Industrial Control Systems," 1999.
[5] MODBUS.ORG, "Modbus Application Protocol V1.0," 2002.
[6] ASHRAE, ANSI/ASHRAE Standard 135-1995, "BACnet A Data Communication Protocol for Building Automation and Control Networks," 1995.
[11] Weibull Distribution, http://www.weibull.com/LifeDataWeb/weibull_probability_density_function.htm/
[7] S. Soucek and T. Sauter, "Quality of Service Concerns in IP-Based Control Systems," *IEEE Trans. Industrial Elect.*, vol. 51, no. 6, 2004, pp. 1249–58.
[10] Y. J. Won *et al.*, "Measurement Analysis of IP-Based Process Control Networks," *APNOMS '07*, LNCS 4773, Sapporo, Hokkaido, Japan, Oct. 10–12, 2007, pp. 385–94.
[9] S.-H. Han *et al.*, "The Architecture of NG-MON: A Passive Network Monitoring System," *13th IFIP/IEEE Int'l. Wksp. Distrib. Sys.: Ops. and Mgmt.*, LNCS 2506, Montreal, Canada, Oct. 2002, pp. 16–27.
[8] Network General, http://www.networkgeneral.com/

BIOGRAPHIES

YOUNG J. WON [S] (yjwon@postech.ac.kr) received his B.Math degree in computer science from the University of Waterloo, Canada, in 2003. He is a Ph.D. student in the Department of Computer Science and Engineering, POSTECH, Korea. His research interests include Internet traffic measurement and analysis, application traffic classification, and traffic modeling.

MI-JUNG Choi [M] (mjchoi@postech.ac.kr) received her B.S. degree in computer science from Ewha Womans University in 1998, and her M.S. and Ph.D. degrees from the Department of Computer Science and Engineering, POSTECH, in 2000 and 2004, respectively. She was a postdoctoral fellow at INRIA, France, from 2004 to 2005 and at the School of Computer Science, University of Waterloo, Canada, from 2005 to 2006. Currently, she works at POSTECH as a research professor. Her research interests include XML-based network management, and NGN and future Internet management. She is a member of KNOM.

MYUNG-SUP KIM [M] (tmskim@korea.ac.kr) received a Ph.D. degree in computer science and engineering from POSTECH, Korea, in 2004. From September 2004 to August 2006 he was a postdoctoral fellow in the Department of Electrical and Computer Engineering, University of Toronto, Canada. He has been a professor with the Department of Computer and Information Science, Korea University, since September 2006. His research interests include service and network management, and Internet traffic monitoring and analysis.

JAMES WON-KI HONG [SM] (jwkhong@postech.ac.kr) is a professor with the Department of Computer Science and Engineering, POSTECH. He received a Ph.D. degree from the University of Waterloo, Canada, in 1991 and an M.S. degree from the University of Western Ontario in 1985. His research interests include network and systems management, distributed computing, network monitoring and analysis, and network planning. He has served as Technical Chair (1998–2000), Vice Chair (2003–2005), and Chair (2005–present) of the IEEE ComSoc Committee on Network Operations and Management (CNOM). He also served as Director of Online Content for IEEE ComSoc (2004–2005). He is a NOMS/IM Steering Committee Member and a Steering Committee Member of APNOMS. He was technical co-chair of NOMS 2000 and APNOMS '99. He was Finance Chair for NOMS 2006 and IM 2005, and Finance Chair and Chair of the Local Planning Committee for NOMS 2004. He was General Chair for APNOMS 2006. He is an editorial advisory board member of *JNSM*, *IJNM*, *JTM*, and *TNSM*. He is also Editor-in-Chief of *KNOM Review Journal*. He is a member of KICS, KNOM, and KISS.

HWAWON HWANG (hwawon@posco.co.kr) received B.S. and M.S. degrees in computer science and engineering from POSTECH in 1993 and 1995, respectively. Currently she is a senior researcher in research laboratories of Pohang Iron & Steel Corporation (POSCO), Korea. Her research interests include network traffic monitoring and analysis, system and network security, and system software development.

JUN-HYUB LEE (mujigae@posco.co.kr) is a network engineer at POSCO, Korea. He received a B.S degree in electrical engineering from Konkuk University, Korea, in 2000. He has been working on troubleshooting, technical support, and consulting networks in the steel-making field since 2001. His research interest is traffic monitoring and analysis in control networks including TCP/IP and fieldbus networks.

SUNG-GYOO LEE (sunggyoo@posco.co.kr) is a general manager of the Electric and Control Maintenance Department at Pohang Works, POSCO. He received a B.S. degree in electrical engineering from Busan National University in 1983 and graduated from the techno M.B.A. program at POSTECH in 2004. He has worked in the field of maintenance of electrical facilities and introduction of the Computerized Maintenance Management System. Currently, he is in charge of the maintenance of electric, instrument, and computer systems at POSCO.

BOOK REVIEWS *(Continued from page 35)*

encompassing IP over ATM, MPLS, GMPLS, the NGN architecture, DWDM, QoS provisioning, and VoIP are discussed.

Part 2 of the book is devoted to queuing theory and its applications. Chapter 4 serves as an introduction to queuing theory. In Chapter 5 the author describes Markov chains. Starting from basic notions such as the Poisson arrival process, birth-death process, and Kendal's notation for queuing systems, the author focuses on analysis of several Markovian queues. The chapter is completed with Erlang-B generalization for non-Poissonian arrivals. Various types of queues with generalized service time distributions are dealt with in Chapter 6. Namely, M/G/1 queuing theory as well as some of its applications are provided in the chapter. Chapter 7 is devoted to application of the queuing theory to analysis of local area networks. Networks of queues are scrutinized in Chapter 8.

The bibliography is very rich. For the reader's convenience, it is provided separately for each chapter as its last section. The references guide the reader to materials on background knowledge discussed briefly or just mentioned in the book as well as additional information on the subject discussed.

The book *Queuing Theory and Telecommunications: Networks and Applications* can be unhesitatingly recommended for students as an essential textbook in studying telecommunications systems and queuing theory as well as for teachers as valuable material for advanced undergraduate and graduate courses. Also, engineers involved in teletraffic and communications systems analysis will find the book valuable and knowledge broadening.