Modeling and response time analysis of the Level 2 system for a continuous steel casting process
Yongsoo Kim, Cheeha Kim*
Department of Computer Science and Engineering, Pohang University of Science and Technology (POSTECH), 77 Cheongam-Ro, Nam-Gu, Pohang, Gyeongbuk 790-784, Republic of Korea

A B S T R A C T
When a process control system (PCS) is designed or improved, it is very desirable to check in advance whether it responds within time constraints because correcting a system after implementation requires time and effort. A PCS processes bursty messages from programmable logic controllers. The response time to each individual message is different from one another, so it must be evaluated. This paper presents a modeling method to set up an analytic model of a PCS for the response time analysis using Colored Generalized Stochastic Petri Net. The model is analyzed using transient queueing analysis and simulation. A test bed system is built that is identical to a part of a PCS in a continuous casting plant, and is used to measure response times. The approach is validated by showing that simulations using the analytic model match the measured values well. The approach is applied to the whole PCS in the same plant to estimate the response times, which are presented and discussed in the paper.

1. Introduction
Diverse control systems for factory automation in the manufacturing industry have been configured in a hierarchical manner for various reasons (Kim, Choi, & Park, 2012). The control systems are functionally categorized into five levels according to the ISA-95 standard (ISA, 2010). The hierarchy of control systems in the steel-making industry is described in Fig. 1. Level 2 systems consist of server computers called process control system (PCS). The PCS supervises and controls the factory. The PCS collects field data from Level 1 systems, calculates reference values, and sends them to Level 1 systems, which control field equipment directly. Level 1 systems include programmable logic controllers (PLCs) and distributed control systems (DCSs). The PCS receives production orders from a Manufacturing Executive System (MES) and manages production data. Level 4 systems participate in business-related activities and consist of Enterprise Resource Planning (ERP) and Supply Chain Management (SCM) systems. Level 4 systems are not described in Fig. 1.

To track materials and send reference values punctually, the PCS must respond within time constraints, because a delayed response may result in defective products. So when a new PCS is designed or an already-deployed PCS is modified, one should check in advance that the anticipated system meets the requirements, because correcting the system after it is implemented requires time and effort. In this paper, we propose a method to model PCS and to analyze its response time. To the best of our knowledge, there is no tool to evaluate the response time of PCS during the design stage.

To develop and maintain PCS effectively, middleware is used as the standard development environment of PCS in the steel-making industry. The middleware for PCS provides operating system-dependent application programming interfaces (APIs) such as task service, message service, file service, timer service, shared memory service and monitoring service (Hwang & Kim, 2005; Hwang & Shin, 2008; Kim et al., 2012). The message service has a service handler and a message queue for sending and receiving messages (Fig. 2).

The most commonly used function of the middleware in the steel-making process is to transmit and receive messages between application tasks. Hwang and Shin (2008) measured the response time of the message service between only two application tasks (Fig. 3) but did not evaluate the response time of the whole application software.

Alwakeel and Almansour (2011) analyzed the throughput of message-oriented middleware with priority queuing. Vandal, Gokhale, and Gokhale (2007) used CSIM to present a model-based performance analysis method of an active object based system. Praphamontripong, Gokhale, Gokhale, and Gray (2007) used Stochastic Reward Net (SRN) to present a model of the performance of a Reactor-pattern-based system. Ramani, Trivedi, and

*Corresponding author.
E-mail addresses: nangakim@postech.ac.kr (Y. Kim), chkim@postech.ac.kr (C. Kim).
http://dx.doi.org/10.1016/j.conengprac.2015.08.001 0967-0661/© 2015 Elsevier Ltd. All rights reserved.
Dasarathy (2000) developed SRN models of the Common Object Request Broker Architecture (CORBA) event service. Fernandes et al. (2004) used the Generalized Stochastic Petri Net (GSPN) to model and analyze the performance of IBM’s MQseries. These studies focused on performance analysis of the middleware systems. In contrast, we are concerned with evaluating the response time of the whole PCS.

Kounev (2006) presented a case study of a distributed component-based E-business system, showing how Queueing Petri Net models can be exploited to predict performance. Praphamontripong, Gokhale, Gokhale, and Gray (2007b) used SRN to decompose and implement a queueing model of an asynchronous Web server. However, these studies evaluated overall average response time. Notably, PCS of a steel-making process handles the bursty messages from PLC, the arrival rate of which is higher than the service rate in the short bursty period, so the response time to messages may vary. Because the response times to messages differ, the response time to each individual message needs to be evaluated. Especially, we focus on evaluating the response time for the burst messages. There are many approaches to response time analysis (Alwakeel & Almansour, 2011; Fernandes et al., 2004; Hwang & Shin, 2008; Kounev, 2006; Praphamontripong et al., 2007a, 2007b; Ramani et al., 2000; Vandal et al., 2007), but we hardly find a good method to analyze the response time of a PCS.

In this paper, we present a modeling method to set up an analytical model of a PCS for the response time analysis using Colored Generalized Stochastic Petri Net (CGSPN). Section 2 gives an overview of PCS and CGSPN. Section 3 presents the modeling method and its validation. Section 4 demonstrates an application of the approach to the whole PCS in a continuous casting plant and also to a modified PCS in the same plant to estimate the response times. Section 5 makes concluding remarks with directions for future research.

2. Background

2.1. Overview of PCS for a continuous casting process

Continuous casting is the process in which molten steel is solidified into billets, blooms or slabs (Fig. 4). A ladle is filled with liquid steel in the electric or basic oxygen furnace. The ladle is taken to the continuous casting process and is raised onto a turret. The turret rotates the ladle into the casting position above the tundish. Liquid steel in the ladle is poured into the tundish, then into a mold. Solidification begins in the water-cooled copper mold, and continues with sprayed water and air at the strand. Solidified products are torch-cut, then discharged for intermediate storage. The process has been automated in terms of PLCs and a PCS. In the continuous casting process, Level 1 systems consist of only PLC, not like Fig. 1.

Most PCSs in the steel-making industry have similar software architecture. The software architecture of a PCS in a continuous casting plant is described in Fig. 5. The PCS executes several tasks for material tracking, process and equipment control, data management and communication.

The level 1 receiving task (L1R) receives data from a PLC (Level 1 system) and transmits the messages to the tracking task (TRK). The level 2 receiving task (L2R) receives messages from other process control systems (Level 2 systems), the function of which is to supervise the electric or basic oxygen furnaces and to send the positions of ladles and the information of materials, and transmits the messages to TRK. The level 3 receiving task (L3R) receives messages from MES (Level 3 system) and transmits the messages to the control task (CTL). TRK tracks all important values related to equipment and materials; the values include the attributes and heat data of the slabs produced in the continuous casting plant. TRK collects data and controls equipment. After processing the received messages, TRK sends the associated messages to other...
Data gathering, modification, and storage of casting results are performed. The data-gathering task (DTG) produces data for subsequent processing. The data-gathering task (DTG) and the level 3 sending task (L3S) are sent to external systems by the level 1 sending task (L1S), the level 2 sending task (L2S), and the level 3 sending task (L3S).

Fig. 1 shows the general range of the response time for each layer of control systems in the steel-making industry. In a continuous casting plant, PLCs scan the status of field instruments and control the machines in 2–100 ms periods. L1R in the PCS gets the sensing data from the PLCs in 1-s intervals (Fig. 6). The PCS is required to process each message within 500 ms for the PLC or within 1 s for other PCSs and MES. L1R checks the plant status from the received data and generates messages associated with each equipment and material in 1-ms intervals on average. Therefore, bursty messages are transmitted to TRK and propagated to DTG and CTL. Because the execution time of TRK is shorter than those of CTL and DTG, the messages from TRK to CTL and DTG are bursty. Also, those bursty messages are propagated to L1S, L2S, and L3S, and thereby effect the response time of PCS. Delay of feedback to the PLC may result in defective products, so the response time to each message should be checked.

Sequence diagrams in Fig. 7 show how the PCS handles the messages from the PLC. The parallel dotted vertical lines represent time. There are two boxes on each dotted line; one with the task name and an empty box. An empty box corresponds to the task execution. The horizontal arrows represent the message flows.

When a ladle arrives at the casting position (Fig. 7(a)), the message of LadleCastingPosition is generated and sent to TRK in order to update a tracking data table and collect the ladle weight. CTL calculates cooling patterns of the mold and the strand. Then, the results are sent to the PLC. When a ladle is closed (Fig. 7(b)), the message of LadleClose is generated and sent to TRK which updates a tracking data table and collects the close time. DTG gathers the casting result data and writes them into a data file. Then, the information of the ladle close is sent to other PCSs.

For each incoming message, the PCS processes it and figures out where to send the processing results. For example, at the end of the task L1R in Fig. 7, the destination is determined depending on the input message of LadleCastingPositionOn or LadleCloseOn. According to the destination, there are different timing constraints. For PLC, it is within 500 ms and for other PCSs and MES, within 1 s in the continuous casting process.

2.2. Colored generalized stochastic Petri net and probability distributions

In this section we present an overview of CGSPN, which is a tool for modeling and analyzing system performance. We first examine the Generalized Stochastic Petri Net (GSPN).

A GSNP is defined by a set of places, transitions, input arcs, output arcs, firing weights (probabilities), and initial marking (Marsan, 1990; Zimmermann & Knoke, 2007). Places are used to represent conditions or local system states, and transitions are used to describe events that occur in the system. Arcs specify interconnections between places and transitions. An input arc leads from a place to a transition and indicates that the event can occur in the local state. An output arc leads from a transition to a place, and indicates that the event will induce a local transformation. A token in a place represents the corresponding condition or local state. Tokens move between places according to firing rules imposed by the transitions. When each input place of a transition contains at least the number of tokens equal to the multiplicity of the associated arc, the transition may fire, in which case it removes from each of its input places the number of tokens that is equal to the multiplicity and deposits them in each of its output places.

Two types of transitions are used: immediate transitions, which fire in zero time; and timed transitions, which fire after a certain delay. The numbers of tokens in each place represent the state of the system and are called the marking of the net. The initial arrangement of tokens is called the initial marking.

Colored GSPN allows specific colors (types) to be assigned to a
2.3. Probability distributions and queueing systems

The firing delay of a timed transition specifies the service time of a task. The firing delays are assumed to follow an exponential distribution.

The places of CGSPN represent queues. If \( n \) customers exist in a queue and their service times are \( X_1, X_2, \ldots, X_n \) iid (independent identically distributed) exponential(\( \mu \)) random variables, the distribution of the sum is the Erlang distribution, i.e.

\[
\sum_{i=1}^{n} X_i \sim \text{Erlang}(\mu, n).
\]

Tasks in a PCS can be modeled by queueing systems. Generally, queueing systems with infinite customer population is denoted as A/S/X, where

- A: probability distribution of inter-arrival times,
- S: probability distribution of service times,
- X: the number of servers.

For example, an M/M/1 queue is composed of Poisson arrivals, exponential service time and a single server with unlimited queueing capacity. A D/M/1 queue is used to describe deterministic arrivals with bulk inputs of size \( s \), exponential service time and a single server with unlimited queueing capacity (Chaudhry & Templeton, 1983; Cox & Hinkley, 1970).

For an open M/M/1 queue, which has the unlimited number of customers, the inter-departure times are exponentially distributed and the departure process is a Poisson process (Lipsky, 1992). If the arrival rate \( \lambda < \text{service rate } \mu \), the departure rate is \( \lambda \). If \( \lambda > \mu \), the departure rate is \( \mu \). Inter-arrival times during the bursty and non-bursty period are independently exponentially distributed and so the output of M/M/1 queue also has independently exponentially distributed inter-departure time.

In the network of queues, splitting a Poisson process leads probabilistically to processes that are also Poisson, and merging independent Poisson processes leads to a process that is also Poisson. If the arrival distribution is a Poisson with parameter \( \lambda \), and two departures have probabilities \( P \) and \( (1 - P) \), these departures also have Poisson distribution with mean rates of \( P \lambda \) and \( (1 - P) \lambda \). If two Poisson arrivals with mean rate \( \lambda_1 \) and \( \lambda_2 \) are merged, the resulting arrival is Poisson with a mean rate of \( \lambda_1 + \lambda_2 \).

PLC data arrives periodically at the PCS with bulk input and the inter-departure time of each message is exponentially distributed. This model will be described in the next section.

3. Response time analysis method

3.1. Queueing model of PCS and transient queueing analysis

In this section, we present an analytical model of a PCS for response time analysis, and provide transient queueing analysis of the model. L1R in Fig. 6 can be modeled as a D/M/1 queue with an exponential service time and deterministic bulk inputs, which are of size \( s \). PLC data arrives at the L1R task at discrete time points which are equally spaced at unit time apart. The data is processed and converted to \( s \) messages which are transmitted to TRK. The service time of L1R is independently exponentially distributed with mean \( 1/\mu_{\text{L1R}} \).

In task L1R, let \( N_i \) be the number of messages present immediately following arrivals at time \( i \). We can assume that \( s \) messages arrive at time \( i \). Let \( D_{i,n} \) be the time until the \( n \)th departure after time \( i \) and \( S_{i,n} \) be the service time of the nth message after time \( i \). \( S_{i,n} \) has an independent and identically distributed exponential distribution. Let \( Z_{i,n} \) be the time between the \( (n - 1) \)th and \( n \)th departures, then

\[
Z_{i,n} = D_{i,n} - D_{i,n-1} = S_{i,n}
\]
The mean $\mu_N$ and standard deviation $\sigma_N$ of $N$ are (Cox & Hinkley, 1970)

\[
\mu_N = s/(1 - e^{-k_{1L}}), \quad \sigma_N = s(1 - e^{-k_{1L}})/ (1 - e^{-2k_{1L}}).
\]

Because $k_{1L} = 1000$ in Fig. 6, $\mu_N$ and $\sigma_N$ approximate to $s$ and 0 respectively. Therefore, $s$ consecutive messages depart from L1R, and their inter-departure times have independently exponentially distributed times with mean $1/\lambda_{1L}$.

In the queueing model (Fig. 8) of the PCS, the service times of tasks have exponential distributions with $\mu_{TRK}$, $\mu_{DTG}$, $\mu_{L1S}$, $\mu_{L2S}$ and $\mu_{L3S}$. Messages from L2R and L3R arrive according to a Poisson process with arrival rates $\lambda_{L2R}$ and $\lambda_{L3R}$ respectively. Messages from L1R arrive at an exponential inter-arrival time interval with mean $1/\lambda_{1L}$ during the short bursty period which is the sum of $s$ consecutive inter-departure times. No message from L1R arrives during the non-bursty period. Therefore, the TRK task can be modeled as an M/M/1 queue with two-level arrival rates: $\lambda_{L1R}$ and $\lambda_{L2R}$ during the bursty period and $\lambda_{L2R}$ during the non-bursty period. During the bursty period of TRK, CTL and DTG, the arrival rates are faster than the service rates, i.e. $\lambda_{L1R} > \mu_{TRK} > \mu_{CTL}$, $\mu_{DTG}$. In the queues of L1S, L2S and L3S, the service rates are faster than the arrival rates, i.e. $\mu_{L1S}$, $\mu_{L2S}$, $\mu_{L3S}$.

Tasks (Fig. 8) can be modeled as M/M/1 queues with two-level arrival rates and have similar arrival processes and service mechanisms, so we set up an analytic model for TRK as an M/M/1 queue to study the response time of TRK. We then compare the results with the calculation of transient queueing analysis. Let the size of bulk input be $s$, then L1R receives data from the PLC at 1-s intervals and generates $s$ messages which are transmitted to TRK sequentially at 1-ms intervals. L2R receives messages from other PCS and sends the messages to TRK. In the CGSPN model of TRK (Fig. 9), places represent queues of tasks and transitions represent execution times of the tasks.

The places of the model are as follows:

- $P_{L2R}$ is used to represent a queue which contains messages from other PCS.
- $P_{L1R}$ is used to represent a queue which contains messages from the PLC.
- $P_{TRK}$ is used to represent the queue of TRK.

The following types of timed transitions are used in the model:

- $T_{L2R}$ has an exponentially distributed firing time with mean 64.5 ms.
- $T_{L1R}$ has an exponentially distributed firing time with mean 1 ms.
- $T_{P6}$ is used to generate deterministic bulk messages with 1-s period. Multiplicities of input arc and output arc are both $s$ to move $s$ consecutive messages simultaneously.
- $T_{TRK}$ has an exponentially distributed firing time with mean 10 ms.
All places are assumed to have unlimited capacities and First-In-First-Out scheduling strategy. The colors of tokens are identical in all places and their attributes have generated times, external sources and sequence numbers of messages to evaluate the response time to each message.

3.3. Validation of model

The mean sojourn time (Eq. (5)) and the mean number of messages (Eq. (6)) in TRK were calculated using APPL. Because of computational complexity, we assume that $s=8$. Also the CGSPN model of TRK was analyzed using TimeNet. To validate our approach, we built a test bed system that is identical to parts of the PCS, and measured the system's response times. The implemented software components were L1R, L2R and TRK of a continuous casting plant. An HP Z600 workstation was used: Intel Xeon CPU X5650 at 2.67 GHz, 8 GB RAM. Windows 7 Professional K (64-bit) was installed as an operating system and POSCO Middleware for Open Systems (POSMOS) was installed as middleware for a steel-making industry (Kim et al., 2012).

The numbers of messages in the queue and service were calculated, including the just-arrived message when the 8th message from L1R and the 5th message from L2R arrive (Table 1). When the 8th message arrived, the mean number of messages in the queue and service including the 8th message was 7.63 in CGSPN. Upon the arrival of the 5th message, the mean number of messages in the queue and service including the 5th message was 1.45 in CGSPN. TRK reached equilibrium after the 5th message from L2R, and so the mean number of messages in the queue and service at equilibrium state was 0.45.

When each of eight messages arrives from L1R, the probabilities that the queue and system have $j$ messages including the just-arrived message are $P(n, j)$ for $n=1, 2, \ldots, 8$ because we can approximate that $k=0$ at time $t=0$. The mean response time $E(T_{n,0})$ can be calculated using $P(n, j)$ for $n=1, 2, \ldots, 8$ (Calculation (L1R) in Fig. 10). Upon arrival of the 8th message, the mean number of messages in the queue and service including the 5th message was $N_{8,0}$, which is 7.32 (Table 1). In Table 1, c.i. means a confidence interval.

The 1st message from L2R was assumed to arrive after the 8th message from L1R because the eight messages from L1R are bursty. Upon arrival of the 1st message from L2R, we can approximate $k=7$ because $N_{8,0} = 7.32$. The probability that there are $j$ messages from L2R in the queue and service including the just-arrived messages is $P(n, j)$ for $n \geq 1$ (Calculation (L2R) in Fig. 10). When the 5th message arrives, the mean number of messages in the queue and service including the 5th message was $N_{7,0} = 1.28$. At equilibrium, the mean number of messages in queue and service is 0.28 (in calculation). The mean response time $E(T_{n,0})$ is represented by Calculation (L2R) (Fig. 10).

Analytical results obtained by simulating our analytical model matched the measured values well. The response times of the CGSPN model, the transient analysis and the test bed system were similar (Fig. 10). Also, the response times of the CGSPN model included 99% confidence intervals for comparison. The response times of the 8th message from L1R were 75.67 ms, 73.21 ms and 79.26 ms respectively; the response times of the 5th message from L2R were 12.81 ms, 12.83 ms and 14.49 ms respectively. The response times of messages from L2R during the bursty period were 53.88 ms in CGSPN and 47.05 ms in the test bed. Therefore, the CGSPN model is valid for the response time analysis of TRK, so our approach was applied to the whole PCS in the same plant to estimate the response time in the next section.

4. Application of the model

4.1. Modeling of PCS

The analysis model using CGSPN was set up to evaluate the response time of the whole PCS in a continuous casting plant (Fig. 11).

The places of the model are as follows:

- $P_{L1R}$ and $P_{L3R}$ are used to represent queues for messages from other PCSs (Level 2 systems) and MES (Level 3 system) respectively.
- $P_{L1R}$ is used to represent a queue for messages from PLC.
- $P_6$ and $P_7$ are used to generate 18 consecutive messages. The size of bulk input is $s=18$ in a continuous casting plant.
- $P_{TRK, P_{CTL}, P_{DTG}, P_{L1S}, P_{L2S}$ and $P_{L3S}$ are used to represent queues of the TRK, CTL, DTG, L1S, L2S and L3S tasks respectively.
- $P_3$ and $P_4$ are used to partition messages.

Transitions (Tables 2, 3) represent the arrivals from external systems and the service times of tasks. $T_{L1R}$ is a timed transition which has an exponentially distributed firing time with mean 1 ms. In Fig. 11, $T_{10}$ is a timed transition used to generate the bulk of bursty messages with 1-ms intervals. Multiplicities of the input arc and the output arc in $T_{10}$ are both 18 to move 18 bursty messages simultaneously.

Messages from TRK are partitioned according to local guards (Table 4). The local guard is a firing condition that depends on a boolean function of the input arc variable. An attribute of the message is defined as either data or order. The attribute $x.data$ represents where the message $x$ comes from, and the $x.order$ represents the arrival order of the message $x$. The transition is enabled if the local guard becomes TRUE. The weight in Table 4 is the relative firing probability of the transition with respect to other simultaneously enabled transitions. According to $T_3$ and $T_4$, the first 8 of the 18 messages are transmitted to CTL and the last 10 are transmitted to DTG. $T_0$ and $T_1$ partition messages from L2R with probabilities 0.43 and 0.57 respectively. $T_6$ and $T_9$ partition messages from DTG with probabilities 0.38 and 0.62 respectively.

![Fig. 10. Response times of TRK obtained using three methods.](image-url)
4.2. Simulation results

We measured the elapsed time in TRK of each message since the departure from L1R and L2R (Fig. 12). The elapsed times of messages from L1R increased proportionately with the increase of the arrival order because the arrival rate is larger than the service rate of TRK. The elapsed times of 1st, 8th and 18th message were 8.9 ms, 66.3 ms and 159.1 ms respectively. The elapsed times of messages from L2R decreased drastically because the arrival rate is much smaller than the service rate. Messages from PLC require faster response than those from other PCS and MES, and so the orders of messages are assigned according to required response time; messages that require fast response are placed in the front order.

The first 8 messages among 18 messages are transmitted to the PLC via CTL and L1S. The response time to messages from L1R ranged from 38.1 ms to 169.8 ms (Fig. 13). The response times to the 1st messages from L2R and L3R were 112.7 ms and 77.8 ms respectively. The response times to L2R and L3R messages which arrive during the bursty period were 104.0 ms and 69.4 ms respectively. The maximum processing time for messages from L1R, L2R and L3R to L1S was much less than 200 ms, so the timing constraints might always be satisfied.

The response times to messages up to the departures of L2S and L3S were less than 300 ms (Figs. 14, 15). Because messages to other PCS and MES were required to be processed within 1 s, the timing constraints were satisfied.

4.3. Use of model to predict the response time of a modified PCS

To enhance the control performance in plants, new measuring devices may be installed. The measured values from the field devices are transmitted to a PCS for calculating reference values, and the calculated values are fed back to the field devices. The response time of the new measured values should be guaranteed without degrading the response times to the existing messages. When the PCS is modified to process more messages from a PLC than before, the response time of the PCS must be predicted. The CGSPN model (Fig. 16) of the modified control software includes CTL1 and TRK1 which have separate queues and CPUs.

![Fig. 11. CGSPN model of PCS in a continuous casting plant.](image)

![Fig. 12. Elapsed time of TRK.](image)

![Fig. 13. Elapsed time of messages from L1R, L2R and L3R to L1S.](image)
When a new message is added, messages from L1R to L1S are divided into two groups: one that contains the odd-numbered messages and one that contains the even-numbered messages. The first group is processed by TRK1 and CTL1; the second is processed by TRK and CTL. The maximum response time of TRK was reduced from 159.1 ms (Fig. 12) to 43.1 ms (Fig. 17). The elapsed time of the 8th message up to L1S decreased from 169.8 ms to 101.0 ms. The elapsed times up to TRK and CTL were also reduced. Therefore we can decide that the modified software architecture (Fig. 16) is acceptable.

5. Conclusion

We presented a modeling method to evaluate the response time of the PCS in a continuous casting plant using CGSPN. The L1R task with the bulk data from the PLC was modeled as a D|M/1 queue with deterministic bulk inputs and an exponential service time. Each task of the PCS had a queue and a service time which was represented by a timed transition. The colors of tokens classified the external sources and arrival orders of messages. To validate our proposed method, we showed that the results of the analytic model agreed well with values measured using a test bed. Then the analytic model was applied to the PCS and a modified PCS in the same continuous casting plant. The method described in the paper might have the potential for providing a basis for development of a similar scheme to evaluate the timing of other Level 2 systems.

The PCS is deployed as a fault tolerant system equipped with a standby PCS. There is no specific analysis for the behavior of a failover signal. In the future, we will study on the method to analyze the performance of failover. In near future, PCS and MES are expected to be merged into one system, so we will analyze the characteristics including response times of the integrated system.
References


