Dynamic Polling Scheme Based on Time Variation of Network Management Information Values

K. Yoshihara, K. Sugiyama, H. Horiuchi, S. Obana
KDD R&D Laboratories Inc.
2 – 1 – 15 Ohara
Kamifukuoka-Shi, Saitama
JAPAN
{yoshi, sugiyama, hiroki, obana}@csg.lab.kdd.co.jp

Abstract

Network monitoring is one of the most significant functions in network management to understand the state of a network in real-time. In network management, such as SNMP (Simple Network Management Protocol), polling is used for this purpose. If the time interval for two consecutive polling requests is too long, then we cannot understand the state of the network in real-time. Conversely, if it is too short, then the polling message traffic increases and imposes a heavy load on the network although we can understand the state of the network in real-time.

Many schemes have been proposed for controlling the overheads of dynamic polling by throttling polling rates. Unfortunately, by only considering overheads, these schemes fail to take into account the rate at which information must be obtained in order to achieve management tasks. Examples of these tasks include checking for threshold violations and determining if additional capacity should be allocated dynamically.

This paper proposes a new scheme for dynamic polling that considers both the overhead of polling traffic and the message rates required for management tasks. Our scheme examines polling traffic, applying the Discrete Fourier Transformation to extract the desired polling rate. We demonstrate the availability of this approach through simulations in which polling requests are varied dynamically. Our scheme adjusts polling rates to make them more consistent and to reduce resource overheads.

Keywords

Monitoring, polling, SNMP, Discrete Fourier Transformation

1. Introduction

Network monitoring is one of the most significant functions in network management to understand the state of a network in real-time. In network management, such as SNMP (Simple Network Management Protocol)[1], polling is used for this purpose. If the time interval between two consecutive polling requests is too long, then we cannot understand the state of the network in real-time. Conversely, if it is too short, then the polling message traffic itself increases and imposes a heavy load on the network although we can understand the state of network in real-time.

Many dynamic polling schemes have been proposed for controlling the increase in the polling message traffic[2, 3, 4, 5, 6, 7]. However, they cannot comply with the time variations of management information values, since their main objectives are to check whether or not a network node is active and the next polling interval is determined based on the round trip time or processing load of a network node, being independent of the time variations of the values. The existing polling schemes are thereby not applicable to such as the following cases where monitoring the time variation of management information values is critical; 1) the case where thresholds are set to management information in order to detect some abnormal network phenomena in real-time, such as abnormal increase in a CPU load and degradation of throughput, 2) the case where service providers monitor their customers' peak utilization and should advise their customers to increase the contract bandwidth when the peak utilization often exceeds the contract bandwidth.

The values which cross the upper and lower threshold or the peak value may often be missed by the existing polling schemes. This may cause us to miss a symptom of a network failure and lose an opportunity to advise. Then, we require a new dynamic polling scheme which can monitor the values much closer to the maximum and minimum values than by the existing polling schemes.

In this paper, we propose a new dynamic polling scheme which enables, by making use of Discrete Fourier Transformation, not only to control the increase in the polling message traffic but also to comply with the time variations of management information values.

In the next section, we describe network management based on polling. Section 3 gives brief reviews of existing schemes and points out their drawbacks and Section 4 introduces the proposed scheme. We evaluate it through simulations in Section 5 and offer some discussions in Section 6.

2. Network Management based on Polling

As shown in Figure 1, a network management system, referred to as a **manager** hereafter, which monitors and controls a network, for example a LAN, polls a network node, referred to as an **agent** hereafter, such as a router, a bridge, a workstation, a PC, or a printer in the network for getting the most recent performance of the agents, for example, the response time, throughput, or pro-

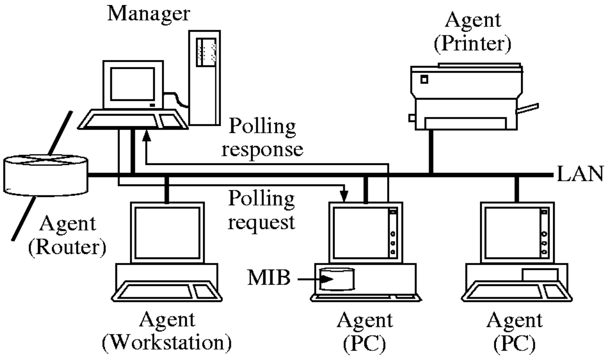
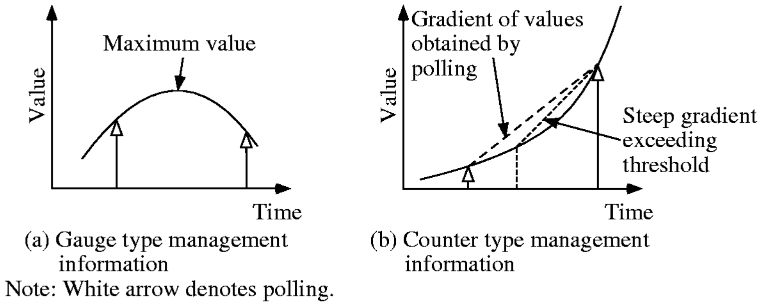


Figure 1: Network management based on polling.



(a) Gauge type management information

(b) Counter type management information

Note: White arrow denotes polling.

Figure 2: Time variation of management information values.

cessing load. The agent polled by a manager obtains a management information value, which corresponds to an object instance in SNMP, from a Management Information Base (MIB) at each agent. Then, the agent responds to the polling request with the result of the polling, referred to as a **polling response** hereafter, which is sent to the manager.

A management information value necessary for polling in compliance with its time variation is a value such as a **Gauge type value** which is a non-negative integer increasing or decreasing, and a **Counter type value** which is a non-negative integer that only increases as shown in Figure 2. The value obtained by a polling response and the gradient of values obtained by consecutive polling responses are important in network management. As such value types, there are the total number of transmitted and received packets at an interface (Counter type values), CPU utilization (a Gauge type value), and buffer occupancy percentage (a Gauge type value) concerning a device provided in MIB-II[8] and vendor specific MIBs[9, 10].

Furthermore, as the importance of application and system management has been growing more and more recently, there are also CPU and memory utilization (Gauge type values) for each individual application such as "FTP" or "WWW", which are sensitive to users' demands in real-time, and which are provided in some MIB[11].

3. Existing Schemes and Their Drawbacks

Since the bandwidth in a network, for example a LAN, is shared by user communication traffic and polling message traffic for network management, it is generally desirable that the polling message traffic is restricted to at most 5% of the minimum bandwidth of a network[1], referred to as **network management bandwidth** hereafter. Some dynamic polling schemes have been proposed for controlling the polling message traffic as described below.

3.1 Existing Schemes

Schemes based on Round Trip Time of Previous Polling[2, 3, 4, 5]
In these schemes, a manager determines the next polling interval based on the round trip time of previous polling, which is associated with network congestion. If the manager does not receive a polling response within a specified period of time, then it predicts that there may be network congestion and expands the next polling interval to control the polling message traffic.

Schemes based on Processing Load of Agent[6, 7]
In these schemes, a manager polls management information values which represent the processing load of an agent such as CPU and memory utilization and then classifies the state of the agent into some levels. For example, if the CPU utilization of the agent is less than 25%, then the state of the agent is classified as first level, that is, an idle state. If the CPU utilization is 75% or more, then the state of the agent is classified as fourth level, that is, a busy state. As the level is higher, the schemes expand the next polling interval to control the polling message traffic.

3.2 Drawbacks

All the above existing schemes based on the round trip time of a previous polling or the processing load of an agent can certainly control the increase in the polling message traffic. However, a problem arises in that managers using the existing schemes cannot comply with the maximum and minimum values of the Gauge type shown in Figure 2 (a) or the steep gradient of the Counter type value shown in Figure 2 (b) exceeding a threshold. This is because their main objectives are to check whether or not a network node is active and the next polling interval is determined based on the round trip time and processing load of a network node, which are independent of the time variations of the values.

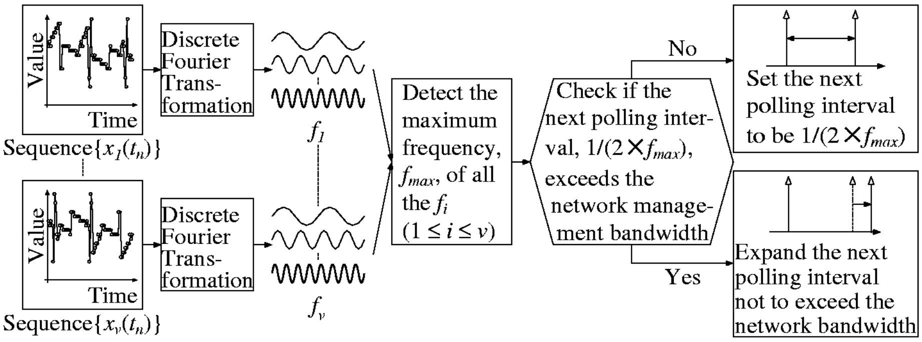


Figure 3: Principle of proposed scheme.

For a solution to the problem, we propose a new dynamic polling scheme which enables us not only to control the increase in the polling message traffic but also to comply with the time variation of management information values by making use of Discrete Fourier Transformation.

4. Proposed Scheme

4.1 Principle

Figure 3 shows the principle of the proposed scheme. Let $\{x_i(t_n)\}$ $(1 \leq i \leq v)$ denote a sequence of values obtained by polling a management information value $\{x_i(t)\}$ at time t, where v is the number of management information values to be polled at an agent and n is the length of the sequence, that is, the number of values in the sequence.

In the proposed scheme, at first, a manager decomposes each sequence, $\{x_i(t_n)\}$, into the sum of sinusoids with different frequencies by means of Discrete Fourier Transformation and finds the maximum frequency f_i for each sequence. Then, the manager detects the maximum frequency f_{max} of all the v maximum frequencies f_i $(1 \leq i \leq v)$, and sets the next polling interval to be $1/(2 \times f_{max})$. If the next polling interval exceeds the network management bandwidth, then the manager expands it so as not to exceed the network management bandwidth.

According to the principle above, the manager can determine the next polling interval in order not only to control the increase in the polling traffic message but also to comply with the time variations of management information values.

Note that Discrete Fourier Transformation is directly applied in the proposed scheme assuming that each interval of the sequence is identical, although each interval of the sequence $\{x_i(t_n)\}$ is not necessarily identical since the manager

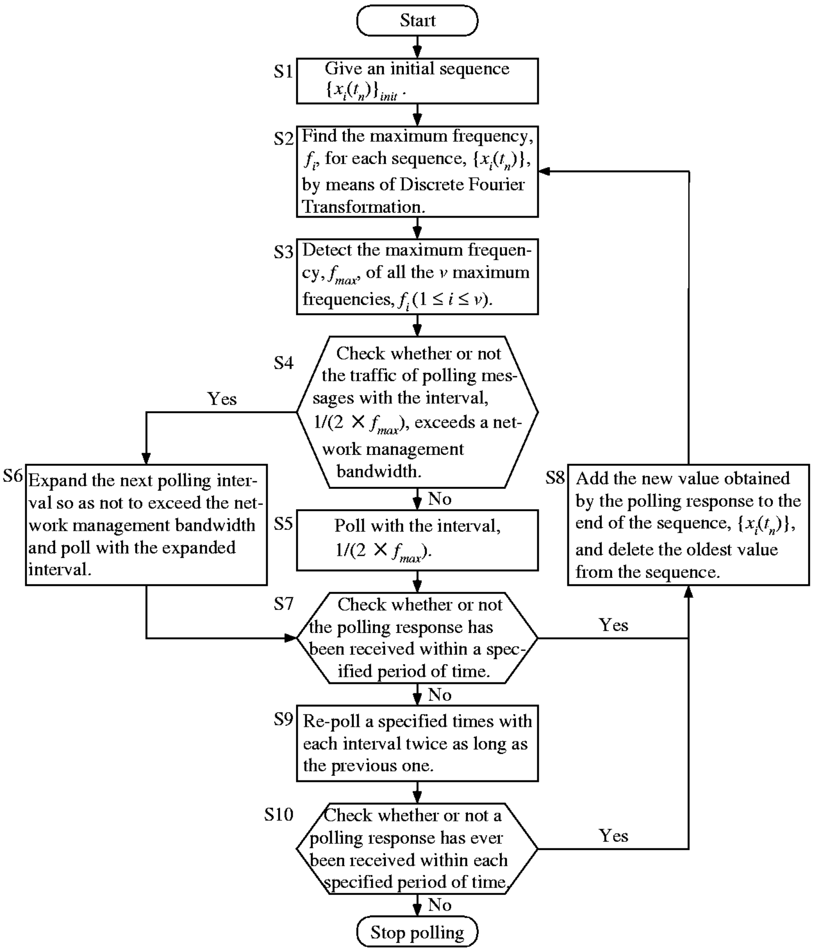


Figure 4: Flow chart of proposed scheme.

dynamically determines the polling intervals. This assumption may cause the manager to detect a higher maximum frequency than the actual one. However, it does not degrade the capability to poll in compliance with the time variations of the values since the polling interval may only be shortened and never be expanded.

4.2 **Algorithm**

We show the flow chart of the proposed scheme in Figure 4. We give an initial sequence, $\{x_i(t_n)\}_{init}$, to the scheme in order to determine the first polling interval(S1). See Section 4.4 for more detail on deriving the polling interval for obtaining the initial sequence. Note that the lengths of each sequence n are all equal to 2^m, where m is a non-negative integer.

The scheme applies Discrete Fourier Transformation to each sequence $\{x_i(t_n)\}$ of management information values, finds the maximum frequency f_i (S2) and then detects the maximum frequency f_{max} of all the v maximum frequencies f_i $(1 \leq i \leq v)$(S3). If $f_i = 0$ for all sequences and $f_{max} = 0$, then the previous f_{max} $(\neq 0)$ is applied.

The scheme derives the bandwidth necessary for polling with the interval $1/(2 \times f_{max})$ from the quotient of the sum of the size of the polling request and its response protocol data units (PDUs) divided by $1/(2 \times f_{max})$, and then the scheme checks whether or not the traffic of polling messages with the interval $1/(2 \times f_{max})$ exceeds the network management bandwidth(S4). See Section 4.3 for more detail on deriving the network management bandwidth. In SNMP, the size of the polling response PDU p (bits) can be treated approximately as a linear function of the number of management information values v to be polled and is derived by equation (1). Note that since the size of the polling request PDU is almost the same as that of the polling response PDU, the total size of the polling request and its response PDUs is summed up to $2 \times p$.

$$p = 144 \times v + 720 \text{ (bits)}. \quad (1)$$

If the traffic of polling messages with the interval $1/(2 \times f_{max})$ does not exceed the network management bandwidth, then the manager polls with the interval(S5).

If it does exceed the network management bandwidth, then the manager expands the next polling interval T (sec) so as not to exceed the network management bandwidth b (bps) by applying the equation (2), and polls with the expanded polling interval(S6).

$$T = \frac{2 \times p}{b} \text{ (sec)}. \quad (2)$$

In the scheme, the manager checks whether or not the polling response has been received within a specified period of time, for example 10 sec(S7). If received, then the manager adds the new value obtained by the polling response to the end of the sequence, $\{x_i(t_n)\}$, and deletes the oldest value from the sequence. The scheme moves to S2 and continues the process(S8). If not received, then it re-polls a specified number of times, for example at most 4 times, with each interval twice as long as the previous one(S9).

The manager finally checks whether or not any polling responses have ever been received within each specified period of time. If the manager has received a polling response to the re-polling, then it moves to S8(S10). If never received, then the manager anticipates that there may be a link or an agent failure in the network and stops polling.

4.3 Network Management Bandwidth

Let B (bps) and N denote the minimum bandwidth in a network and the number of agents to be polled, respectively. Then the network management bandwidth is equally assigned to N agents and is derived by the equation (3), where we assume that 5% (= 0.05) of the minimum bandwidth in the network can be contributed to the network management bandwidth.

$$b = \frac{B \times 0.05}{N} \text{ (bps)}. \quad (3)$$

For example, when $B = 2 \times 10^6$ (bps) (= 2 Mbps) and $N = 20$, the network management bandwidth for each agent b is 5×10^3 (bps) (= 5 Kbps).

4.4 Polling Interval for Obtaining Initial Sequence

We derive the polling interval for obtaining an initial sequence from the sum of the size of the polling request and its response PDUs $2 \times p$ in equation (1) and the network management bandwidth b (bps) in equation (3). This enables us to obtain an initial sequence such that the polling message traffic never exceeds the network management bandwidth.

For example, if the number of management information values to be polled v is 10, that is, each size of the polling request and its response PDU p is 2160 (bits) from equation (1), and the network management bandwidth b is 5×10^3 (bps) (= 5 Kbps), then the polling interval for obtaining the initial sequence is 8.64×10^{-1} (sec) from equation (2).

5. Evaluation of Proposed Scheme through Simulations

We implement the proposed scheme in C language and show the capability to detect a polling interval in compliance with the time variations of management information values through simulations performed on a SUN Ultra1 with a single CPU.

At first, in Section 5.1, we evaluate precision of the maximum frequency f_{max} detected by the proposed scheme, by comparing with an actual maximum frequency. Next, in Section 5.2, we evaluate the capability to poll complying with the time variations of the values, by comparing with a simple polling scheme with a constant polling interval.

We show the conditions in the simulations in Table 1. We make use of two functions $g_1(t)$ and $g_2(t)$ as shown in Figure 5, for the time variations of management information values in the simulations. The maximum frequency of each function is 0.2 and 0.4, respectively. The time variation of the values is defined to be $g_1(t)$ in $0 \leq t < 2000$ (sec) and $g_2(t)$ in $2000 \leq t < 4000$ (sec).

5.1 Precision of Maximum Frequency Detected

Figure 6 shows the maximum frequency f_{max} detected by the proposed scheme

Table 1: Conditions in simulations

Item	*Value*
Number of agents (N)	1
Number of management information values (v)	1 (Gauge type)
Minimum bandwidth of network (B)	1.0×10^5 bps
Network management bandwidth (b)	5.0×10^3 bps
Length of sequence (n)	1024
Polling interval for initial sequence	1.0 sec
Time variation of management information values	See Figure 5.
Processing time for a polling request PDU at manager	2.0×10^{-3} sec
Polling response time*	5.0×10^{-3} sec
Processing time for a polling response PDU at manager	3.0×10^{-3} sec
Processing time for updating sequence	1.0×10^{-3} sec

$*$ Round trip time including processing time in agent.

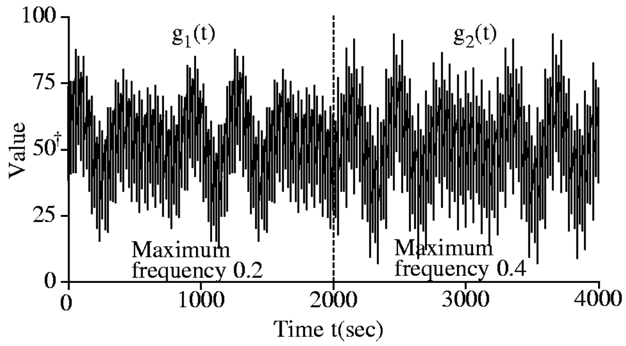
when it is applied to the time variations of management information values shown in Figure 5. Note that the maximum frequency detected by the proposed scheme is less than or equal to 0.5 due to Discrete Fourier Transformation.

The average value of f_{max} in $0 \leq t < 2000$ (sec) is 0.25 (1/sec) and the manager polls 1000 times since the average polling interval $1/(2 \times f_{max})$ is 1/0.5 sec (= 2 sec). The average value of f_{max} in $2000 \leq t < 4000$ (sec) is 0.43 (1/sec) and the manager polls 1720 times since the average polling interval $1/(2 \times f_{max})$ is 1/0.86 sec ($\approx$ 1.16 sec). Each average value is close to the actual maximum frequency 0.2 of $g_1(t)$ and 0.4 of $g_2(t)$. This shows that the proposed scheme practically achieves enough precision.

5.2 Capability to Poll Complying with Time Variation of Management Information Values

We evaluate the capability to poll complying with the maximum and minimum, referred to as **extremum** hereafter, of management information values such as shown in Figure 2 (a) by comparing with that of the simple polling scheme with a constant polling interval. We introduce an evaluation function defined by equation (4). The numerator of equation (4) is the sum of the polled values. Note that we subtract 50 from each polled value in evaluating the equation since 50 has been added to the value to make all the values polled non-negative (see Figure 5). The denominator P of equation (4) is the number of pollings in 4000 sec.

$$\frac{\sum_{i=1}^{P} |\text{Value by the } i\text{-th polling} - 50|}{P}. \quad (4)$$



$$g_1(t)=8\{\sin(0.4\pi t)+\sin(0.2\pi t)+\sin((2/15)\pi t)+\sin(0.1\pi t)+\sin((1/150)\pi t)+\sin((1/200)\pi t)\}+50 \quad (0\leq t<2000)$$

$$g_2(t)=8\{\sin(0.8\pi t)+\sin(0.2\pi t)+\sin((2/15)\pi t)+\sin(0.1\pi t)+\sin((1/150)\pi t)+\sin((1/200)\pi t)\}+50 \quad (2000\leq t<4000)$$

†50 is added to the value to make all the values polled non-negative.

††The maximum angular frequency of $g_1(t)$, $\omega_1=0.4\pi=2\pi f$. ($f=0.2$)
The maximum angular frequency of $g_2(t)$, $\omega_2=0.8\pi=2\pi f$. ($f=0.4$)

Figure 5: Time variation of management information values used in simulations.

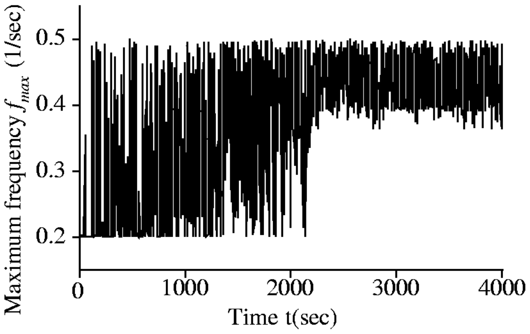


Figure 6: Maximum frequency detected by proposed scheme.

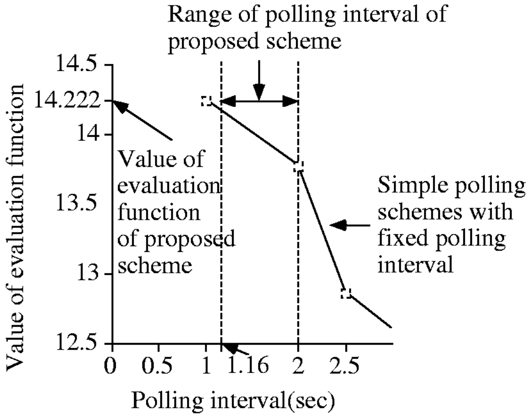


Figure 7: Capability to poll complying with time variation of management information values.

The higher the value of the evaluation function of equation (4) is, the higher the capability to poll complying with the extremum of management information values is. In other words, if we fix the number of pollings P, then the greater the sum of the polled values is, the closer to the extremum is the value obtained by every polling. Or, if we fix the sum of the polled values, then the less the number of pollings P the closer to the extremum is the value obtained by every polling.

Figure 7 shows the results, which are values of the evaluation function relative to the polling interval(sec). In the proposed scheme, the manager polls 2720 times in 4000 sec and the value of the evaluation function is 14.222. This is almost the same as the value 14.235 in the simple polling scheme with a fixed 1 sec polling interval. In this example, the proposed scheme can comply with the time variation of management information values with a smaller number of pollings, that is, 68% (= (2720/4000) × 100) of that of the simple polling scheme.

6. Discussion

6.1 Theoretical Time Complexity

The theoretical time complexity of the proposed scheme is $O(Nvn \log n)$, which is dominated by that of Discrete Fourier Transformation, $(n \log n)$, where N is the number of agents, v is the number of management information values, and n is the length of sequence.

6.2 Actual Processing Time

The actual processing time (overhead) of the proposed scheme is 13.1×10^{-3} sec (= 13.1 msec) on the average even on a low grade machine under the conditions in Table 1, where the processing time for Discrete Fourier Transformation, 7.1×10^{-3} sec (= 7.1 msec), is dominant. This is acceptable if we assume a network such as a LAN where a manager and agents are closely connected physically and end-to-end delay between them is negligible. For example, the theoretical minimum polling interval such that the polling message traffic never exceeds the network management bandwidth is 17.3 msec, where the minimum bandwidth in a network B is 2×10^6 (bps) (= 2 Mbps, effect bandwidth in Ethernet), the number of agents N is 1, and the number of management information values v is 1.

6.3 Scalability of Proposed Scheme in the Number of Agents Polled

We derive the maximum number of agents polled by the proposed scheme simultaneously when the following two conditions hold. We assume that the minimum bandwidth in a network B is 5% of 2.0×10^6 (bps) (= 2 Mbps), and the minimum polling interval T_{min} of the polling whose traffic never exceeds the network management bandwidth, is 1 sec.

condition 1; The polling message traffic does not exceed the network management bandwidth.
condition 2; The processing time of the proposed scheme does not exceed the next polling interval.

We obtain the maximum number of agents in two cases, i.e. one, the number of management information values v is 1 and two, v is 32.

In the case where v is 1, from equations (1), (2), and (3), the maximum number of agents N satisfying condition 1 is 57. Also, from the theoretical time complexity in Section 6.1 and the processing time in Section 6.2, if the processing time for Discrete Fourier Transformation is 7.1×10^{-3} sec (= 7.1 msec) and that for updating the sequence is 1.0×10^{-3} sec (= 1.0 msec) when the length of the sequence n is $1,024$, then the value of N satisfying condition 2 is 139 from equation (5). Therefore, the value of N satisfying both conditions is 57. Note that the second term of equation (5) is the processing time for updating the sequence and the third term is the sum of the processing time for a polling request PDU, its response PDU and the polling response time in Table 1.

$N \times 1 \times 7.1 + 1.0 + 11.0 < 1000$ (msec). (5)

In the case where v is 32, the value of N satisfying condition 1 is 9 and the value of N satisfying condition 2 is 4. Therefore, the value of N satisfying both conditions is 4.

When we apply the proposed scheme to a large-scale network, we make use of a general technique to provide multiple managers in a geographically or organizationally distributed manner and to introduce a hierarchical relationship among these managers. By this technique, the number of agents to be polled per manager can be reduced so that the scalability of the proposed scheme is acceptable. If the number of devices to be polled per manager is above the scalability of the proposed scheme in a domain, then we divide the domain and provide a manager in each divided sub-domain.

7. Conclusions

This paper proposes a new scheme for dynamic polling that considers both the overhead of polling message traffic and the message rates required for network management tasks.

The proposed scheme decomposes sequences of values obtained by past polling responses into the sum of sinusoids with different frequencies by making use of Discrete Fourier Transformation and determines the next polling interval based on the maximum frequency of all of the sinusoids.

Through the simulations, we show that the scheme is highly applicable, from the view points of precision in detecting the maximum frequency of the time variation of management information values and capability to poll complying with the time variation of the values.

The theoretical time complexity of the proposed scheme is $O(Nvn \log n)$, where N, v, and n denote the number of agents, the number of management information values, and the length of the sequence, respectively. The actual processing time is 13.1 msec on a widely available workstation with rather low processing capability, where $N = 1$, $v = 1$, $n = 1,024$. Evaluation of the proposed scheme in actual environment is for future study.

ACKNOWLEDGMENT

The authors wish to thank to Dr. Takuro Muratani, President & CEO of KDD R&D Laboratories Inc. for his continuous encouragement and Dr. Kenji Suzuki, Executive Vice President for his useful suggestions in this research.

References

[1]. IETF RFC 1157. *A Simple Network Management Protocol (SNMP)*, May 1990.

[2]. K. Ohta, N. Sun, G. Mansfield, and and Y. Nemoto. Effective polling control for network management using SNMP. *IEICE Technical Reports on Information Network*, IN94-135:91–96, Nov. 1994.

[3]. T. Ika, K. Ohta, N. Kato, G. Mansfield, and and Y. Nemoto. Management information gathering on wide-area network in consideration of information accuracy. *Proc. of the* 1997 *IEICE General Conference*, B-7-46:175, Mar. 1997.

[4]. P. Moghe and M. Evangelista. Rap - rate adaptive polling for network

management applications. In *Proc. of IEEE NOMS '98*, pp.395–399, 1998.
[5]. B. Bondi. A nonblocking mechanism for regulating the transmission of network management polls. In *Proc. of IFIP/IEEE IM '97*, pp.565–580, 1997.
[6]. P. Dini and R. Boutaba. Deriving variable polling frequency policies for pro-active management in networks and distributed systems. In *Proc. of IFIP/IEEE IM '97*, pp.541–552, 1997.
[7]. P. Dini, G. Bochmann, T. Koch, and B. Kramer. Agent based management of distributed systems with variable polling frequency policies. In *Proc. of IFIP/IEEE IM '97*, pp.553–564, 1997.
[8]. IETF, RFC 1213. *Management Information Base for Network Management of TCP/IP-based internets: MIB-II*, Mar. 1991.
[9]. Fore Systems, Inc. *Fore-switch.mib*, Mar. 1997.
[10]. Cisco Systems, Inc. *Cisco-IP-stat-mib*, Aug. 1997.
[11]. IETF, RFC 2287. *Definitions of System-Level Managed Objects for Applications*, Feb. 1998.

Biography

Kiyohito Yoshihara is a member of the Network Management System Laboratory, KDD R&D Laboratories inc... Since joining the Labs. in 1995, he has been engaged in research on network management. He received the B.E. and M.E. from Tokyo Institute of Technology in 1993 and 1995, respectively.

Keizo Sugiyama is a senior research engineer of the Network Management System Laboratory, KDD R&D Laboratories Inc.. Since joining the Labs. in 1987, he has been engaged in research on OSI Protocol, EDI (Electronic Data Interchange) and network management. He received the B.E. and M.E. from Kyoto University in 1985 and 1987, respectively.

Hiroki Horiuchi is a senior research engineer of the Network Management System Laboratory, KDD R&D Laboratories Inc.. Since joining the Labs. in 1985, he has been engaged in research on formal description techniques for communication protocols, OSI protocols and network management. He received the B.E. and M.E. from Nagoya University in 1983 and 1985, respectively.

Sadao Obana is a senior manager of the Network Management System Laboratory, KDD R&D Laboratories Inc.. Since joining the Labs. in 1978, he has been engaged in research on computer communication, database and network management systems. He received the B.E., M.E. and Dr. of Eng. Degree of electrical engineering from Keio University in 1976, 1978 and 1993, respectively.