韓國港灣學會 第13卷 第2號, pp. 1~14. 1999.

論 文

A Study On Management Protocol For Distributed Systems Management

*S. S. Park**

분산시스템관리를 위한 관리 프로토콜에 관한 연구

박 승 섭

Key Words : Distributed Systems Management(DSM: 분산시스템관리), Manager-manager Interaction(매너저간 상호동작), CMIP(공통관리정보 프로토콜), Distributed Network Management System(DNMS: 분산망관리시스템), Negotiation Process (교섭 프로세스), Cooperation(협력)

Abstract

As the size and complexity of network increase, Distributed Systems Management (DSM) will be significant issue within information network in order to increase the high reliability and to improve the flexibility of network management. The OSI management model has several problems. The key problems are that it does not fully address the problem of how to develop communication protocol in support of DSM, and how to classify the management connection criteria.

In this paper, to solve first problem described above, this paper propose the connectionless CMIP to accomplish for effectively managing the distributed management system, and indicate its efficiency; this protocol is available to negotiate among the managing systems, to handle the dynamic informations. To work out second problem, we introduce the connection criteria in the hierarchy of management systems, and finally evaluate the efficiency of a suggested protocol during cooperative negotiation among the managing systems.

* 정회원, 부경대학교 컴퓨터멀티미디어공학부 교수

1. Introduction

Distributed management problems (Langsford, 1988) will be aggravated by the facts that the network will consist of diverse network equipments, networking services provided by different carriers, and new introducing technologies. To ensure all network resources interwork correctly and effectively, high-reliable and distributed management will be needed in information network environment which can be managed.

The OSI management model has several problems. The key problems are that it does not fully address the problem of how to develop communication protocol in support of DSM, and how to classify the management connection criteria.

In this paper , to solve first problem described above , we propose the connectionless CMIP for supporting distributed management to assist management tasks, and indicate its service primitives; this protocol is available to negotiate among the managing systems, to handle the dynamic informations in distributed systems management environment.

This protocol is available to achieve truly distributed cooperative management in the area of managing system to managing system interaction which have not yet defined by ISO management model[16]. It will not be occurred the development of truly distributed cooperative management without such protocol in the area of managing system to managing system interaction. To work out second problem, we introduce the connection criteria for management system's hierarchy which includes Sub-manager to deal with the dynamic management information. And we then indicate its ideal management hierarchy. And we evaluate finally the efficiency of a proposed protocol during negotiation among the managing systems, and then present its efficiency.

2. Distributed Systems Management Environment

2.1 Objectives And Definitions

Generally, the networks are composed of

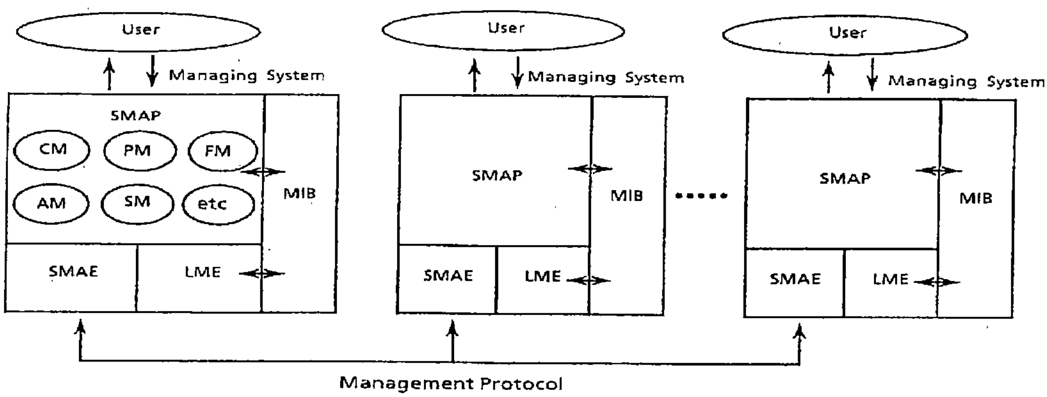

Fig. 1 Several conformant management entities

different types of devices, systems that have grown over time to satisfy different user requirements. For the network user, the term "network" refers to all of the resources including physical components (e.g., terminal, switches, concentrators, host processors, etc.), logical components(e.g., applications, protocols, data-bases, etc.), and aggregate components(e.g., subnetworks, domain, the whole network, etc.) that make up today's information networks. The goals of managing a network are to achieve automated operations that reduce operational complexity, increase staff productivity, provide the needed functionality, and improve network availability by eliminating human error.

DSM is a relatively new field and there is as yet no agreed definition. We will use a more rough definition as follows (Park, 1991).

DSM is one in which several autonomous management system supporting management functions interacts in order to cooperate to achieve an OSI management-based inter-operability. The management functions coor-dinates their activities and exchange management information by means of management protocol transferred over an information networks.

In this paper, the term DNMS is used to refer to the autonomous management system with OSI management-based interoperability architecture for the monitoring, control and coordination of resources within the OSI environment and OSI protocol standards for communicating information pertinent to those resources.

2.2 Configuration Of Management Systems

Because of the complex and diverse network entities, and demand for high-quality services and the difficulties to handle in the real-time events, it is indispensable to use the intelligent facilities in the network system (Cronk, 1988; Feridum, 1996b; Feridum, 1988).

The aim of distributed systems management is to provide a single set of tools for managing all the network resources within a network. DNMS is an automatic network management system with additional intelligent facilities. A DNMS structure based on DSM requirements corresponds to each management entity in Figure 1. In this model, SMAP can be divided in accordance with network environment to perform network management functions : Management APplication (MAP) module and other resources shown in Figure 2, Figure 3 (Feridum, 1996a; Park, 1992; Park, 1997; 박, 1999).

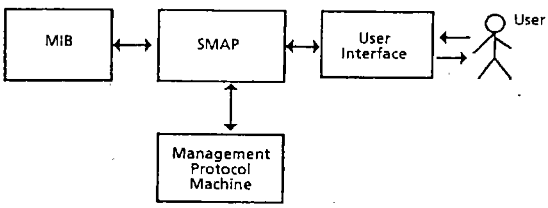

Fig. 2 DNMS structure

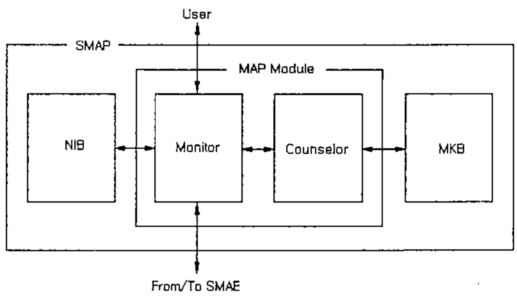

Fig. 3 Configuration of SNMP

3. Management Protocols For DSM Environment

3.1 Network Management Protocol and communications

There are several typical protocols which are widely used for systems management, e.g., Simple Network Management Protocol (SNMP) (Case, 1990) and Common Management Information Protocol over TCP/IP (CMOT) (Warrier, 1990), Common Management Information Protocol(CMIP), etc. (Cassel, 1989; ISO/IEC 9595).

They may be, roughly, classified as two types of protocols in views of management models : In event-driven basis protocol, for example, CMIP/CMOT, the managed systems only sends informations to the managing system in case of need to report the occurred events. On the other hand, polling-driven basis protocol, for example, SNMP, is used when managed systems are polled for a certain information and return this information synchronously to the managing system.

The development of distributed systems management is unlikely to occur without the standards in the area of managing system to managing system (i.e., Manager- Manager) interactions which have not yet defined by ISO/IEC.

In this reason, we want to point out that there should increase a demand for broadcasting management protocols to achieve truly distributed management environment. Intrinsically, CMIP is connectoin-oriented protocol (IEEE, 1993). Two CMIP application entities can exchange management operational message, only if they establish and maintain an association with each other, i.e., OSI association. It is not so suitable for broadcasting, mainly due to the enormous information overhead resulted from message replications, as a prerequisite to operational communication.

But, the results obtained from the Ref. (Ben, 1990) leads us to indicate that compared with SNMP, the large networks, e.g., WAN, ISDN, ATM networks, radio and satellite network etc., must be taken to adopt CMIP in point of number managed systems even if very large information overhead, because this protocol is an event-driven basis scheme.

And also, considering that future management systems will be extended to OSI-based management implementations, there will be necessary to study the connectionless CMIP (Kobayashi, 1990) to accomplish the large scaled DSM environment, which this environment may be dynamically changing due to additions and deletions of managing systems. For these reasons, connection oriented protocol is not suitable to handle the dynamic information of which its value changes frequently, because heavy communication overhead is incurred to establish connection phase and clearing phase.

We will restrict our discussion only on connectionless CMIP suitable for dynamic changing distributed environment.

The service primitives of connectionless CMIP are presented in Section 3.2.2 for negotiation process among DNMSs. In essence, both connection-oriented CMIP and connectionless CMIP may be based on request/reply scheme.

3.2 The Connectionless CMIP

3.2.1 A View of Cooperative Negotiation Process

Before entering this protocol, we now, roughly, describe a negotiation process for management activities among DNMSs.

A negotiation is established by the process of local mutual selection based on a bidirectional transfer of information.

This process, in brief, may be described that how overloaded DNMS with tasks to be executed can be found the most appropriate idle DNMS to execute these tasks which means workload, i.e., dynamic information as described in Section 4.1.

The negotiation process among DNMSs is illustrated in the following scenario:

--Each DNMS executes autonomously independent of other DSMs under normal conditions.

--If the DNMS are overloaded in a certain situation, it may broadcast the request messages to another DNMSs to be supported.

--The DNMS to be requested (we call them responder hereafter) from overloaded DNMS replys to the requesting DNMS which is overloaded (also call requester) to his load state if the requester's requests are executable.

--The requester can distribute its workload to the most available responder with most lightly load state in disregard of no replying responders due to failures, etc.

This is basic process for negotiation, and is described in more detail in Section 3.2.2. The basic idea of this process is derived from a Ref. (Stenkovic, 1984) in which a algorithm has been accepted as a possible heuristic methods for cooperation among distributed systems.

3.2.2 The Basic Service Primitives

Before describing the services, we suppose that a request in a certain DNMS broadcasts to all DNMSs in the network, insuring that it will

Table 1 Relative rating of connection criteria for management hierarchy

Case	Each Connection Type			Relative Rating of Criteria				
	Manager to Sub-Manager	Sub-Manager to Sub-Manager	Sub-Manager to Agent	Bandwidth	Propagation Delay	State	Reliability	Effectiveness
1	CO	CO	CO	Very Low	High	High	High	Very Low
2	CO	CO	CL	Low	Medium	Medium	Medium	Low
3	CO	CL	CO	Low	Medium	Medium	Medium	Low
4	CO	CL	CL	Medium	Low	Low	Low	Medium
5	CL	CO	CO	Low	Medium	Medium	Medium	Low
6	CL	CO	CL	Medium	Low	Low	Low	Medium
7	CL	CL	CO	Medium	Low	Low	Low	Medium
8	CL	CL	CL	High	Very Low	Very Low	Very Low	High

Note : CO ··· Connection oriented
CL ··· Connectionless

be certainly received by all DNMSs. The simplest method to accomplish this is flooding techniques (Frank, 1985).

We use this method because it is simple to implement in despites of large bandwidth requirements.

To reduce the wasteful network traffic, we include the hop-count to prevent indefinite reforwarding of the same messages between requester and responders in the consideration of network environment shown in Table 1.

We now introduce the service primitives of proposed protocol and the processing of each primitive.

A) Support Announcements

Normally, individual DNMS executes autonomously a certain management activities. When it takes an overloaded situations in a DNMS, the overloaded DNMS (requester) broadcasts to another DNMSs (responders) to be supported with the **support announcement service primitive.** In this case, it is possible to negotiate a connection in which each requester can send support announcement service to the other.

B) Announcements Evaluation

Available responders evaluate support announcements service made by several requesters.

If the requested tasks are able to be executed, then the available responders store the requested contents in its received list. Then the available responders make reply to the originated requesters with **announcements evaluation service primitive** including **load margin**, which calculates from its current load state about the accepted support announcements service. The load margin represents the executable amount by available responder. From this time, the requester does not wait forever for responder's replys.

If their replys have not been submitted within arbitrary fixed time (timeout facilities), they are ignored because it is not happened the inconsistency to the processed results.

But it may have an influence on systems management performance.

C) Degree Of Demand Evaluation

When the load margin is received from the available responders, the requesters calculate the **degree of demand** toward each available responder using a received load margin sent by responders. And then submits this service primitive with degree of demand to the available responders. Here, we define the degree of demand like this. It represents the amounts that the requesters, in some measure, are dependent on each responder toward its execution. For example, the degree of demand of requester becomes large if such case arises that its request can only be resolved by a certain responder. On the contrary, if another responders can also resolve its request, then this degree becomes small.

D) Predicting Execution

Available responders which had received the degree of demand calculate the **predicting workload amount** (PWA) on the assumption that the responders execute all received degree of demands. Then the PWA value is also registered in its list, returns to the requesters. This amount can be used to prevent the double awards and contentions to a single responder. It is described in more detail in E) Awards Precessing.

E) Awards Processing

In principle individual responders choose, and receive the requester's requests which have the highest value degree of demand.

E-1) if the degree of demands is the same values on the its list, each responder accepts the first registered request by first-come first-serviced principle.

E-2) but, if the requester has a possibility of double awards toward more than responders (this fact permits degradation of all systems management's performance because a certain requester is not awarded), then the second time award is neglected at that time. After other awards are finished, an unlinked responder searches a certain requester which can not be awarded on its list, then pass over the right of control(i.e., right of choosing an available responder) to that requester.

E-3) provided that the contention does not occur, a certain requester which has received the right of control can award other responder in order of PWA values, i.e., the highest PWA value is selected first and so on.

E-4) but, it enters again the broadcasting state when the responder to be processed does not exist.

By doing so, it can resolve the contentions and double awards among responders due to full use of all available responders.

Here we show the general flow of protocol processing steps in Figure 4. The contents of service primitives is described in detail in ref.(Park, 1992): The address of requester and responder are needed in 'Direction to' field. To determine the shortest route because of dynamically changing informations, the hop count, which prevent indefinite reforwarding of the same message between requester and responders in consideration of network environment, must be defined by designer. We

```
STEP 1. Overloaded requester broadcast to
responders to be supported
STEP 2. if adjacent responders exist
        then available responders respond with
             current load margin to the originated
             requester, and go to STEP 3
        else enter again in broadcasting state
             after passed arbitrary time interval
STEP 3. Requester calculates the degree of demand
      based on the received load margins
STEP 4. if another requesters does not exist
        then responders compute PWA value using
            the degree of demmand, and go to
            STEP 5
        else enter in broadcasting state, and repeat
            from STEP1 to STEP3, and go to
            STEP 4
STEP 5. Requester awards its load to the most
        appropriate responder
STEP 6. if occur double awards to a responder
        then select next better responder
            if next better responder exists
              then go to STEP7
              else enter again broadcasting
            state
        else STEP6
STEP 7. It enters linking state
```

Fig. 4 Flow of proposed protocol processing steps

also described the informations conveyed between the requester and the responder in 'Parameter' field. And also in that field, when a certain DNMS broadcasts to be supported at first time, the information must be identified according to management information types as described in Section 4.1.

Acknowledge (Ack) mechanism will be needed in 'Parameter' field for development of reliable protocol which each service primitives must indeed reach each management system.

4. An Evaluation And Discussion

4.1 Connection Criteria For Management Systems

The connection criteria among management systems can be evaluated according to bandwidth, propagation delay, state, reliability, efficiency. Here we define that the 'bandwidth' represents the sum of the number of messages sent over all channels in unit time, and the 'state' terminology means the amount of storing the information which can include logical identifiers, list of address, flow control mechanism, etc.

Table 1 rates 8 cases of connection type against these criteria for management hierarchy shown in Figure 5.

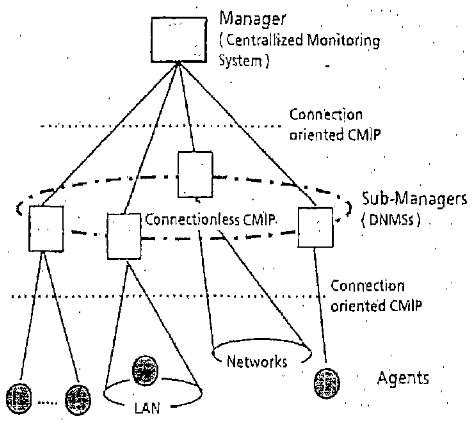

Fig. 5 Hierarchy of management system

The five relative values per criterion are very low, low, medium, and high. A case such that bandwidth, propagation delay, and state are very low, reliability and efficiency are high ratings, is considered to be best.

To deal with the infrequent dynamic management information, as described below, which can not be processed in his Sub-Manager, Sub-Manager broadcasts the messages to the other Sub-Manager; they do not need to establish and then maintain connections.

This reduces the processing overhead even if low efficiency. Here we consider the 'hop count', as described in Sect. 3.2.2, which consequently the efficiency will be increased.

However the trade-off is that the sender of message can not be sure that the message has reached its destination. Thus we introduce an acknowledgement mechanism for development of more reliable protocol.

In consideration of above reasons, we select the three cases, i.e., case 2, case 3, case 5, in Table 1. Among them, the case 3 is chosen in the result as following reasons.

Here we can classify the management information of managed objects which are monitored by agents as follows: the MIBs can be classified into two cases according to the property of management information; while the static information (e.g., network topology, network connections, naming scheme's name and address attributes, etc.), which value is almost not changed, can be stored in the Sub-Manager's MIB, the dynamic information (e.g., CPU utilization, amount of free memory, etc.), can be stored in the agent's MIB because of heavy communication overhead by the frequent change of information's value. In this manner, the amount of communication for access can be reduced by classifying the MIB which stores the static or dynamic management information.

Therefore it is easy to achieve the better performance of overall network, extensibility, and realization of mechanism compared with the case

Table 2 Comparison of each systems management

Method	Centralized Systems Management (Conventional Method)	Distributed Systems Management (proposed method)
Advantages	· Easiness of unified management · High speed responses · Fit for small scale network	· High reliable management · Flexible, extensible network configuration · In dependency of network · easiness of management mechanism intelligence · Fit for heterogeneous network · Suitable for current network
Disadvantages	· Not suitable for high reliability · Possibility of overloads · Not fit for becoming larger heter-ogeneous network	· Security · Ambiguity of management range · Complex, parallel control algorithm

of only one MIB used. For many management activities, the Sub-Manager must recognize the events on an occurrence basis or on a periodic basis from the agents. So, CMIS (ISO/IEC, 7498) provides two event report services to accommodate this need : confirmed m-event-report, non-confirmed m-event -report.

In practice, the ideal management environment of networks shown in Figure 5 will be the case 3 in Table 1. And here we compare each systems management to make clear the advantages and disadvantages of conventional method and proposed method as described in Table 2. In these methods, we may be thought that distributed systems management will be significant for managing the future diverse networks as compared with conventional method though it exists disadvantages.

4.2 Evaluation Of Management Workload Distribution

There may be no previous reports about managing system to managing system interactions in distributed systems management. To truly achieve the distributed management environment, it should be necessary to investigate distributed systems management protocol. This may be the first paper which deals with such a topic.

This proposed connectionless CMIP facilitates the distributed control of cooperative workload execution with efficient negotiation among DNMSs in OSI management environment. It may have a means how overloaded DNMS with workload to be executed can find the most appropriate available DNMs to execute these workloads among DNMSs.

We apply the processing of each service primitives to the negotiation process algorithm of distributed systems management of network. And this proposed protocol will be more suitable for satellite and radio network compared with hardware networks. This algorithm is replicated at each DNMS and is invoked by any DNMSs asynchronously.

Before entering a discussion, we suppose that each DNMS has an infinite size which stores in suitable order more than one protocol message arriving simultaneous, they know nothing about the topology of the network, the requester does not wait forever for responder's replys, and network itself is error-free and reliable.

We assume a case that each requester for the management workload broadcasts the existence of the overloaded states to other responders with support announcements message, available requesters evaluate support announcements made by several requesters, and submits replys on those for which they are suited.

After negotiation process is finished through the processing of each service primitive as described in Section 3.2.2, it is shown that the results of management workload distribution in Table 3. In that table, we assumed that (i) each responder accepts the first registered request by FCFS principle, (ii) the executable amount by each responder to each requester is given in the upper term of the correspondent ij th entry. For example, the executable amount of responder 2 to the requester 1, 3, 4 is 10, 10, 5 respectively. Each request is homogeneous. (iii) the processing capacity of responder 1, 2, 3, 4 is at most maximum value in its replys toward each requester.

The columns represent the requesting DNMSs (requesters) which want to be supported, and the row stands for supporting DNMs (responders). The upper term of ij-entry represents the load margin which available responder j replys to the requester i, denoted by LoadMargin (i,j). The middle term shows the degree of demand toward the available responder, Degree Of Demand (i,j), the lower term indicates the PWA (i,j). We now formulate the degree of demand and PWA as follows : Degree Of Demand (i,j) = LoadMargin (i,j) / $\sum_{K \in D}$ LoadMargin (i,k), PWA (i,j) = Degree Of Demand (i,j) / $\sum_{K \in D}$ Degree Of Demand (i,k) , where D denotes the collection of all available responders.

Here, the value of Degree Of Demand (i,j) becomes a maximum value (i.e., 1) when the workload of a requester is accepted only by a certain responder. In Table 3, the ' / ' notation represents the requester itself which wants to be supported since it is overloaded, ' ★ ' notation denotes that workload of a requester does not accept to the responder due to perhaps its failures. Because, in principle, individual responders receives the requester's workload which has the highest value degree of demand, as a result, be done such a workload distribution denoted by ' ◎ ' notation in Table 3.

Note that ' ○ ' notation indicates the second time award. It has bad effect on all systems management's performance. These solution are

Table 3 Evaluation of management workload distribution

	Responder 1	Responder 2	Responder 3	Responder 4
Requester 1	/	10	5	5
		◎ 0.50	0.25	○ 0.25
		0.38	0.21	0.50
Requester 2	15	/	5	*
	◎ 0.75		0.25	
	0.75		0.21	
Requester 3	5	10	/	5
	0.25	0.50		◎ 0.25
	0.25	0.38		0.50
Requester 4	*	5	10	/
		0.33	◎ 0.67	
		0.24	0.58	

Note: · Upper term of ij-entry represents Load Margin
· Middle term of ij-entry represents Degree of Demend
· Lower term of ij-entry represents PWA

shown in Section 3.2.2, particularly, E) Awards Processing. Here we compare two cases; the first one is that the contentions and double awards to a single responder due to the selection of the most lightly management workload system are occurred. The second one is that the contentions and double awards to a single responder can be resolved by using the PWA value by which all available responders are fully used.

While the first case permits the bad effect on all systems management's performance owing to no full use of all available responders, the second case prevents the contentions and double awards owing to full use of all available responders. For example in Table 3, in case of first case, the requester 3 can not distribute its workload to the most available responder due to the second time award of requester 1 to the responder 4. This fact permits degradation of all systems management performance because requester 3 is not awarded. But in the second case, requester 3 can award its management workload to the next appropriate responder using by PWA value.

Thus, we presented an example by using the negotiation process. This process avoids the contention among responders. there are no DNMSs whose failure can completely block its negotiation process.

5. Interactions of DNMS and Software Structure

This section illustrates the sequence of operations usually performed by each DNMS's modules when user requests to access network management systems in heterogeneous system environment.

First, the user issues management requests to access local as well as remote resources in a

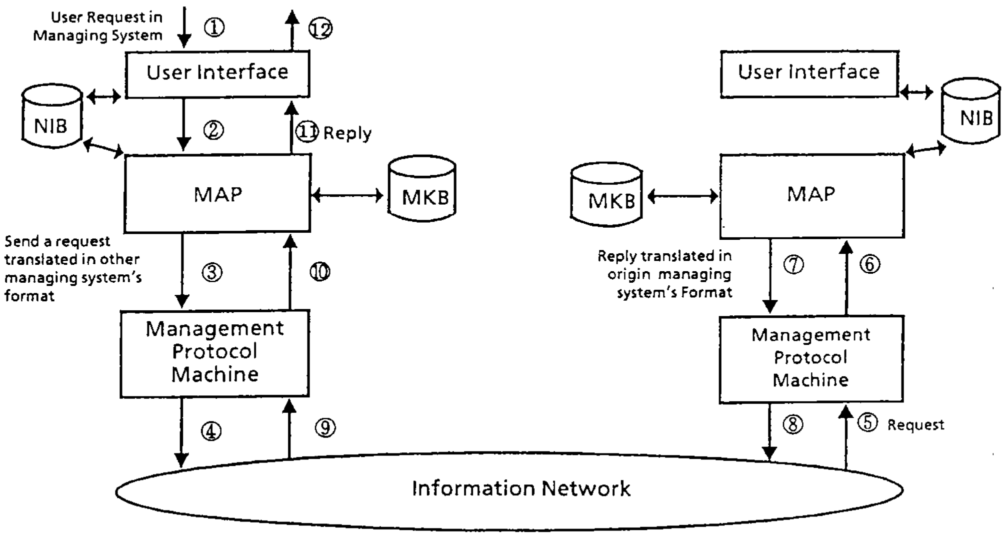

Fig. 6 Interactions of distributed systems management

format compatible with one's management system. Then, MAP module in DNMS structure translates remote requests into a format compatible with other management systems by using translation rules of data representations, command in NIB. Next, the translated requests are carried out through connectionless CMIP to other management systems. Finally, the replys generated in the other management systems have to be translated into a format that can be understood by the requesting user.

These interactions are illustrated in Fig. 6 which include its operation orders. In Fig. 7, we outline the overall software structure of DNMS and describe the behavior of the its modules, when a request is received, in Fig. 8.

Finally, Ref. (Park, 1991) compared each systems management to make clear the capabilities and advantages of distributed systems management. Among these methods, it may be thought that distributed systems management is good a candidate for managing today's multivendor networks.

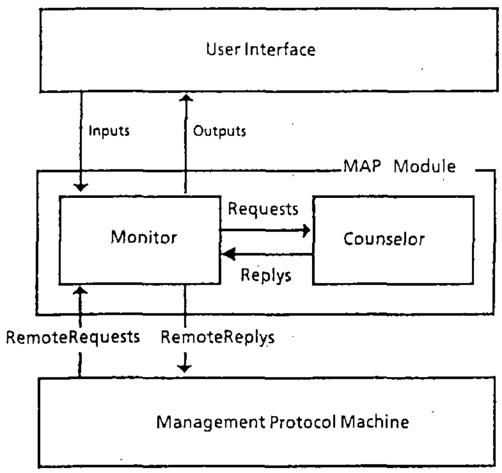

Fig. 7 Software structure of DNMS

```
ManagementProtocolMachine Task :
LOOP
   read local ManagementProtocolMachine
   SEND RemoteRequests TO Monitor ;
   delay ( period )
END
Monitor Task :
Var Replys = ( Internal, External ) ;
LOOP
   SELECT
    RECEIVE Inputs FROM UserInterface ;
   OR
    RECEIVE RemoteRequests FROM ManagementProtocolMachine ⇒
     SEND RemoteRequests TO Counselor
     IF invalid RemoteRequests THEN
       SEND RemoteRequests TO
                   origin ManagementProtocolMachine
   OR
    RECEIVE Replys FROM Counselor ⇒
     CASE Replys OF
       Internal : BEGIN
            SEND Outputs TO UserInterface
            END
       External : SEND RemoteReplys TO
                        ManagementProtocolMachine
     END
   END
END
CouselorTask :
LOOP
   RECEIVE Requests FROM Monitor ⇒
     SEND Replys TO Monitor
END
```

Fig. 8 Behavior of the DNMS modules

6. Conclusions

In this paper, we described the requirements to accomplish full distributed management environment, such as user interface level, OSI communication level, management data level, and management functions level. And we presented the basic structure for DNMS with intelligent facilities which could reduce the human expertise needed for system management.

We proposed the connectionless CMIP fit for dynamically changing distributed management environment. Without such protocols in the area of managing system to managing system interactions, it may not be occurred the development of truly distributed cooperative management. We evaluated the connection criteria among management systems and then indicated its ideal management hierarchy, and an example of cooperative negotiation process using proposed protocol is presented along with discussion on the evaluation of management workload distribution. And also indicate its efficiency.

We believe that the proposed structure of DNMS and the connectionless CMIP are most useful for distributed systems management environment, will be much better suited for managing heterogeneous network as shown in Table 2, and will expand greatly the network management performance.

Acknowledgement

The author wish to thank to Dr. S. Noguchi, who the president of Information Processing Society of Japan(IPSJ) and Prof. N. Shiratori at Tohoku University, Japan for fruitful discussions. Thanks are also due to Dr. Y. Kobayashi IBM Japan, Ltd. and Dr. N. Yoshikai at NTT Laboratory for their helpful suggestions, which improved this paper.

요 약

정보통신네트워크의 대규모와 복잡성의 확대와 더불어 네트워크 관리의 유연성, 고신뢰성 등을 향상시키기 위해서 분산시스템의 관리가 중요한 과제로 등장한다. 기존의 OSI 관리모델은 다음과 같은 여러 가지 문제점이 있다. 1) 분산관리시스템을 지원하기 위한 통신프로토콜개발, 2) 관리접속의 기준 등이다.

본 논문은 상기 서술한 첫 번째 문제점을 해결하기 위해서, 분산관리시스템을 효율적으로 관리하기 위한 비접속형 공통관리정보프로토콜(CMIP)을 제안하고 그 효율성을 제시한다. 제안된 프로토콜은 동적 정보를 취급하기 위해서 관리 시스템사이의 교섭에 유용하다.

두 번째 문제점을 극복하기 위해서 관리 시스템의 계층적 구조에 있어서 접속 기준을 나타내고, 관리 시스템들 사이에 협조적인 교섭을 할 동안 제안된 프로토콜의 효율성을 평가한다.

References

1) Ben - Artzi, A., Chanda, A., Warrier, U., Network Management of TCP/IP Networks : Present and Future, IEEE Network Magazine, 1990, pp. 35-43.
2) Case, J. D., Schoffstal, M., Davin, J., A Simple Network Management Protocol (SNMP), RFC 1157, 1990.
3) Cassel, L.N., Patridge, C., Westcott, J., Network Management Architectures and Protocols : Problems and Approaches, IEEE Jour. On Sel. Areas in Comm. Vol.7, No.7, 1989.
4) Cronk, R. N., Callahan, P.H., Bernstein, L., Rule-Based Expert Systems for Network Management And Operations : An Introduction, IEEE Network, 1988, pp. 7-21.
5) Feridum, M., Heusler, L., Nielsen, R., Implementing OSI Agent/Managers for TMN, IEEE Comm. Magazine, pp. 62-67, Sept. 1996a.
6) Feridum, M., Implementing OSI Agent/Managers for TMN, IEEE Communication Magazine, pp. 62-67, Sept. 1996b.
7) Feridum, M., Leib, M., Nordne, M., Ong, J., ANM : Automated Network Management System, IEEE Network, Vol.2, No.2, 1988,

pp.13-19.

8) Frank, A. J., Wittie, L. D., and Bernstein, A. J., Multicast Communication on network computers, IEEE Software, 1985, pp.49-61.

9) Kobayashi, A., Trends of OSI Management protocol and new subjects (in japanese), The 1st Information and communication networking architecture symposium, 1990, pp.5-1 ~ 5-8.

10) IEEE Special Issue : OSI Network Management Systems, IEEE Communication Magazine, Vol. 31, No. 5, , May, 1993.

11) ISO / IEC 7498-4 : Management Framework.

12) ISO / IEC 9595-9596 : CMIS / CMIP.

13) Langsford, A., Distributed Systems Management - Overview of a COST - 11 ter Project, Computer Networks and ISDN Systems, Vol. 15, 1988, pp. 61-65.

14) Park, S. S., Shiratori N., Noguchi, S., Distributed Systems Management For OSI Environment : Components And Their Interactions, IPSJ 43th National Conference (I-295), 1991.

15) Park, S. S., Distributed Systems Management: Problems and Approaches, Proceedings of the KICS Fall Conference, pp. 987-990, Nov. 1997.

16) Park, S. S., Shiratori, N., Noguchi, S., A Study of Communication methods for Distributed Systems Management in OSI Environment, IEICE Technical Report, Vol. 92, No. 237, AI 92-56, 1992, pp. 25-32.

17) Park, S. S., Shiratori, N., Noguchi, S., Distributed Systems Management For OSI Environment : System Configuration, Communication Methods And Their Behavior, IEICE Transactions Communication, Vol. E75-B, No. 8, 1992.

18) Stenkovic, J.A., A perspective on Distributed Computer Systems, IEEE Trans. On Computer, Vol. C-33, No. 12, 1984.

19) Warrier, W., Besaw, L., LaBarre, L., Handspicker, B., Common Management Information Services and Protocol for the Internet (CMOT and CMIP), RFC 1095, 1990.

20) 박승섭, "분산시스템관리를 위한 관리 프로토콜에 관한 연구", 한국항만학회 '99 추계학술대회논문집," pp. 163-175, 1999.