

Mobile Agents IN DISTRIBUTED NETWORK MANAGEMENT

By helping to disperse centralized network management tasks to subnet hosts, mobile agent technology helps conserve network bandwidth and improves management efficiency by decreasing network traffic.

Only a few years ago, most networks were LANs with limited linkages between equipment. During that era, networks were managed via the centralized architecture. Since this architecture is simple and effective, network administration tasks were carried out on a personal computer or workstation. But the recent growth of Internet applications has led to an explosion in the number of networks worldwide, which has placed a heavy burden upon the networks as well as the administrators. It is not surprising then, that the conventional centralized architecture is not sufficient to manage such changes. In this architecture, the central network server routinely requests the status information of local units, which consumes a significant amount of bandwidth. In addition, communication data may be lost during transmission. It is therefore common today to share management tasks with other network management servers in a distributed framework [6].

In a distributed network, the network administrator monitors the trend of network flow to evaluate network performance and diagnose abnormal situations. The analysis data can be obtained from the management information base (MIB) of network equipments. The MIB maintains many data objects for network management, such as system data, equipment status, link status, and communication status, among

others [7]. The MIB data is organized in groups and preserved in a tree-like structure. This provides an opportunity to manage the complicated network tasks in the distributed network management environment.

The new trend in networks involves using mobile agents [1, 3, 4] to manage the distributed network system [2]. The mobile agent can be used to retrieve data from the MIB to monitor the network flow in the distributed environment. That is, management tasks are assigned to an agent and the agent can be sent to remote hosts to execute the assignment. After completing the assignment, the results are carried back to the sender by the agent. In such a case, local subordinate network (subnet) hosts can share the workload with the central network server. This approach has several advantages as described by [9]. In addition, the mobile agent can transform the conventional centralized network architecture into a distributed administrative environment. In case the linkage between the network management server and the subnet breaks down, the mobile agent and the data can temporarily reside at the agent box of local hosts and be retracted after the linkage is restored. This design is suitable for an unstable network environment.

With recent research [8] suggesting the next strategy for executives is to outsource information technology to Internet-based services instead of having their own proprietary information systems, we will undoubtedly see a dra-

matic increase in network traffic in the near future. This strategy calls for more effective distributed network management, which gives the mobile agent-based network management a chance to show its promise (see the sidebar). Here, we describe the framework and characteristics of a prototype system created to demonstrate the advantages of mobile agent-based distributed network management.

A Framework of Mobile Agent-Based Distributed Network Management

To illustrate the mobile agent-based distributed network management system, we propose a framework for implementation shown in Figure 1. Here, the IBM Aglet with Java Virtual Machine is the platform and the development package is the Aglets Software Development Kit (ASDK), available at sourceforge.net/. The subnet server uses the API of AdventNetSNMPv1 (www.adventnet.com) to access the MIB of network equipment. Note that the commands in the SNMP communication protocol are low-level and simple. The network management system only issues simple commands such as get-request, get-next-request, and set to acquire the MIB objects of subnet equipment. The network data groups are defined in MIB-II of RFC-1213. The mobile agent collects the MIB data in the local hosts and brings the data back to the network servers. The MIB has a tree-like data structure that includes groups of system, interface, at, ip, icmp, tcp, udp, egg, transmission, and snmp. These groups are defined to provide a means of assigning object identifiers and a method for the implementation of managed agents to know which objects they must implement (please refer to RFC1213.txt at www.ietf.org). Each node in the MIB tree represents a variable, which is assigned an object identifier (OID). An object is retrieved by the OID through SNMP. For example, the system variables of the network connection equipment are

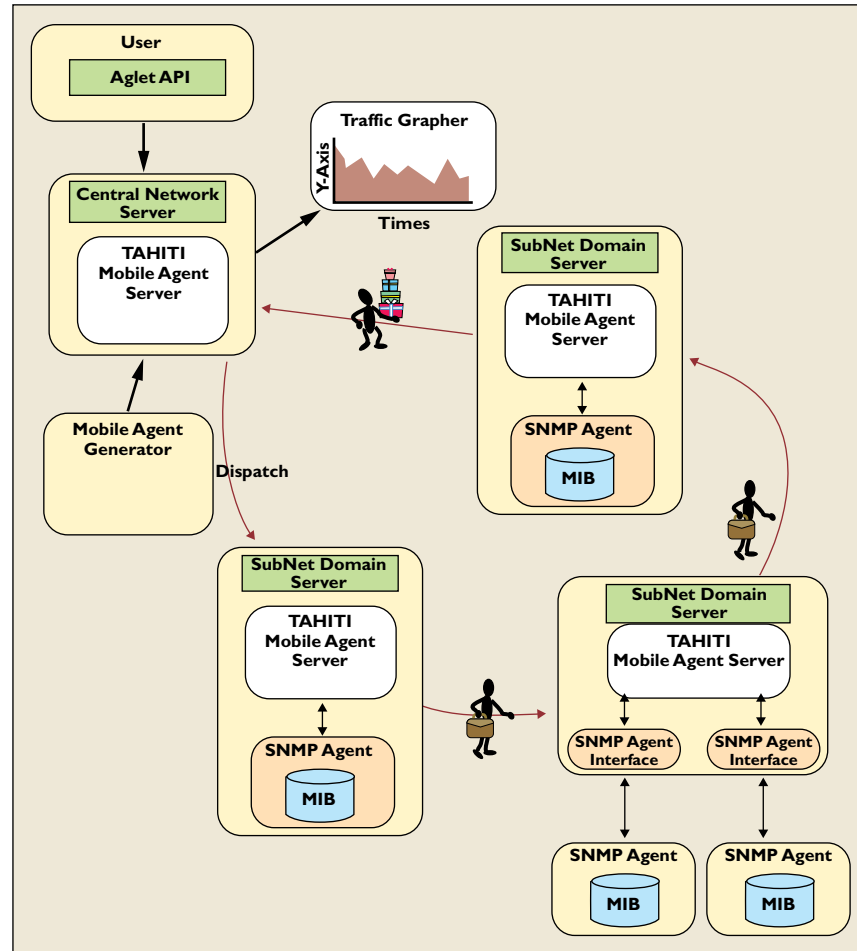


Figure 1. A framework of a mobile agent-based distributed network management system.

defined in the sysDescr of system. The variable is defined as:

```
iso      org      dod      internet mgmt mib-2
1        3        6        1        2        1

mib-2    system sysDescr
1        1        1
```

Therefore, the network management system issues OID 1.3.6.1.2.1.1.1.0 to the network equipment for SNMP data. The description of system data can be retrieved from the MIB of the network equipment.

There are three types of data groups in the network: the non-tabular data group, string variables, time variables, and integer variables such as the system group; the tabular data group, such as the multiple object index, which maps to the network interfaces as the identity of the data retrieval, such as the interface group; and the index group, in which the performance indices used by the network administrator are defined, such as network utilization, the degree of network health, and network signal/error ratio. The first two groups contain simple MIB

objects, while the latter indices provide more information than the MIB objects.

In the SNMP communication protocol, a get-request command can only acquire an MIB object value. The network performance index is normally obtained by combining many simple MIB objects. The collection of network variables is costly. It takes 351 bytes to acquire a single MIB object using the sysUpTime of system group in MIB-II, for example. If there are 100 network hosts in the centralized network management architecture and the MIB object is updated every five seconds, the bandwidth will be occupied by about 606MB per day (351 bytes X 100 hosts X 17,280 samples) [11]. But if the network variables are filtered by the mobile agent in the subnet hosts and only return the calculated results to the central server, the network traffic will decrease significantly.

In order to demonstrate the proposed distributed network management framework, we built a prototype of a mobile agent system that allows users to choose network equipment, set up a SNMP community string and a MIB object ID, and assign tasks to the mobile agent. The mobile agent carries out the assignments and retrieves or sets up MIB objects at subnet hosts. After completing the assignments in a local host, the mobile agent moves to the next network host following a predefined itinerary. However, due to the fact the current IBM Aglet version cannot communicate with the network equipment, the mobile agent operates through SNMP agent interfaces. The agents return to the network management system with collected data when all the assignments are completed. If the management system wants to collect long-term data, the mobile agent will be sent regularly, and the collected data can be used for analysis. To do the job, the system needs to have three modules: a mobile agent module, a data collection module, and a network flow analysis module.

The mobile agent module is stored in both the central network management server and subnet hosts. Both sides must install JDK 1.1 (the Java development kit) and ASDK 1.1 (the Java language for developing IBM's Aglet). A main network management

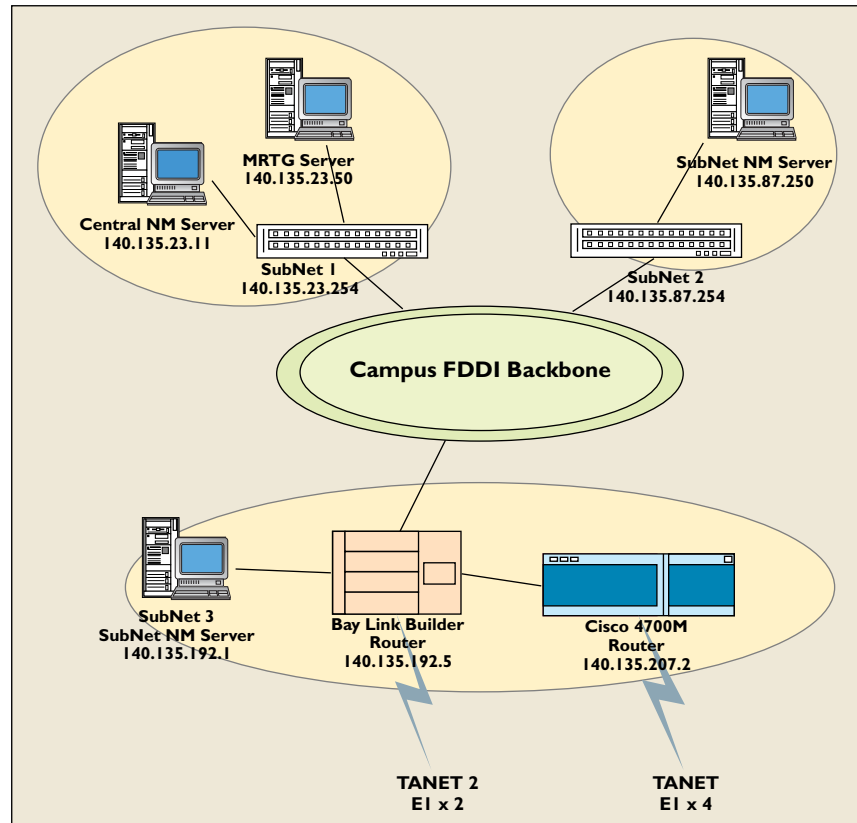


Figure 2. The demonstration network architecture.

server is the core component of this design, and it includes the Tahiti server for executing Aglet programs, and two Aglet programs: Master and Circulate. The Circulate Aglet is a mobile agent program that moves the agent to local hosts and brings the collected data back to the central server. On the other hand, the Master Aglet is a stationary agent of the main server responsible for sending and receiving the mobile agent to local hosts. Similarly, the local host also has both the Tahiti server as the mobile agent execution environment, and the SNMP agent program for collecting the MIB data of subnet equipment. The agent program is composed of the information generated from the network data collection module.

Residing in the local hosts, the network data collection module is written in the API of AdventNet-SNMPv1 (see www.adventnet.com) to implement the SNMP agent program. The API includes commands such as get-request, get-next-request, get-system, get-table, and so on. The module prepares equipment information for the mobile agent, such as network equipment IP, SNMP community string, and request SNMP MIB object ID. The network data collection module issues SNMP packets to network equipment requesting MIB data after receiving the requests delivered by the mobile agent.

The network flow analysis module adapts the functions of the multi-router traffic graphic (MRTG) system from Tobias Oetiker and Dave Rand (available at

people.ee.ethz.ch/~oetiker/webtools/mrtg/). This module can be installed in the main server or in another network flow analysis server. The module analyzes the MIB data taken back by the mobile agent. To install the module, the server needs to have Perl 5 for Win32 from ActiveState (www.activestate.com) and the parameters of the mrtg.cfg file must be the same as those of the mobile agent. Then, the network flow can be analyzed in terms of the date, week, month, or year.

The system was implemented in a campus network connected by Ethernet, Fast Ethernet, FDDI, and wireless laser bridge. The IP address belongs to class B (140.135.xxx.xxx). In order to separate the flow volume between departments and to keep the network expansibility, the sub-network mask was set to 255.255.248.0. In the campus network, a total of 32 subnet domains can be set. In order to have a feasible experimental environment, only three subnet servers are used to monitor three subnet domains. The architecture is shown in Figure 2. It is composed of one 4700 series router from Cisco, one BLN series router from Bay Networks, one third-level exchanger of CoreBuilder 3500 from 3Com, several third-level fast Ethernet exchangers of SuperStackII 3300 from 3Com, several second-level fast Ethernet exchangers of SuperStackII 1100 from 3Com, and four NT servers.

In the experimental network architecture, Java (JDK1.1.8 from Sun Microsystems) is the programming language for system development, IBM Aglet is the mobile agent language, and TCP/IP is the Internet protocol for SNMP. The flow analysis is carried out by an MRTG flow diagram while the SNMP agent interfaces for the mobile agent to retrieve MIB objects are the functions of SNMPv1 API Release 3.1. The MIB is designed to the RFC-1213 standard (MIB-II). The functions of SNMPv1 API Release 3.1 for communication with MIB follow the standards of RFC-1155 and RFC-1157. The functions include SNMP-get, SNMP-get-next, get-SNMP-table, and SNMP-set, which respectively retrieve data, find the next data, find the table information, and set the data in the MIB of network connection equipment based

on the object ID of the MIB tree.

The demonstration evaluates both operation accuracy and system performance. The network operations are compared with the conventional network management system to ensure accuracy, and the per-

Network Node	Centralized Architecture		Distributed Architecture	
	Network Packet	Total bytes	Network Packet	Total bytes
1	16	1504 bytes	18	9456 bytes
2	32	3008 bytes	18	9499 bytes
5	80	7435 bytes	18	10039 bytes
10	160	14872 bytes	19	10807 bytes
20	320	30060 bytes	21	12350 bytes
30	480	45090 bytes	23	13892 bytes
40	640	60160 bytes	24	15370 bytes

Note: The data size of an MIB object at a network node is about 43 bytes, and adding the Aglet program code, the total is about 9456 bytes.

Table 1. The network flow volume of the system group in different architectures.

formance is evaluated in bandwidth consumption and network utilization. The Sniffer of Network Associates is used for network packet analysis, and Protocol Inspector of Fluke¹ is used for network packet collection.

During the demonstration, the subnet domain server collects the MIB values of subnet equipment and sends them to the network central server through the mobile agent. Then the subnet domain host retrieves MIB values through routers from the tabular

Equipment	Network interface	Non-filtered data packet	Total non-filtered data (bytes)	Filtered data packets	Total filtered data (bytes)
Bay BLN Router	5	19	9627	18	9351
3Com 2500 SW	11	19	9805	18	9401
3Com 3000 SW	31	22	13164	19	9858
3Com 3300 SW	102	30	21748	19	10033

Table 2. The network utilization indices for the network flow volume analysis carried by Aglet agents.

data group of subnet equipment, such as the ipRouteTable group in MIB-II. The HP OpenView network management system is used before and after data retrieval to show the results and verify the operational accuracy. Figure 3 shows an example of the ipRouteTable group in MIB-II.

Finally, the subnet domain host obtains the network flow volume of the subnet equipment by using D-Link's D-View network management software. The data will be sent back to the central network server through the mobile agent, and used to evaluate the network performance.

Performance Analysis

The network bandwidth analysis software, Sniffer from Network Associates, is used to collect the

¹See www.flukenetworks.com/us/LAN/Monitoring+Analysis+Diagramming/Protocol+Inspector/Overview.htm.

information communication packet. The communication packet between the network management system and network equipment is collected and filtered for bandwidth analysis. Because MIB data sizes differ between equipment, we repeat the same procedure with the same equipment several times to prevent data bias of equipment selection. Table 1 exhibits the network flow volume for outward communication using the system group of MIB-II as an example. It reveals the network flow is larger in the centralized architecture than in the distributed architecture.

When the number of nodes grows, the network flow doubles the growth. In contrast, the mobile agent creates the network flow overhead for agent programs; however, this overhead is insignificant when the number of network nodes increases, implying that the distributed architecture is more suitable for a growing network environment.

To analyze network utilization, we use Aglet mobile agent to carry the data of the utilization index. The subnet hosts calculate the utilization of network interfaces before the mobile agent arrives and submits the results to the agent upon request. Table 2 shows the flow volume for the network utilization index. Since the transmitted data has been filtered and calculated, the total network flow is not changed significantly between five and 100 network interfaces. Using 102 network interfaces as an example, the total interface flow volume is about 10K when the mobile agent is used. If the number is updated every five seconds, the total volume is about 172MB daily (10K bytes X 17,280 samples). On the other hand, the centralized architecture consumes about 618MB daily (351 bytes X 102 hosts X 17,280 samples) which means the distributed approach can decrease the network flow volume considerably.

Conclusion

The rapidly expanding connectivity in today's network environment requires a distributed network management system. By helping to disperse centralized network management tasks to subnet hosts, mobile agent technology conserves network bandwidth, and improves management efficiency by decreasing network traffic. As shown in the demonstration described here, however, if the network facilities in LANs are few, the traffic for transmitting the mobile agents will be higher than the network traffic of the centralized management architecture. Therefore, the mobile agent-based distributed network management system should be used only for a

Destination	Gateway	Type	Mask	Interface
default	140.135.132.5	remote	0.0.0.0	CoreBuild
140.135.0	140.135.132.32	remote	255.255.248.0	CoreBuild
140.135.16	140.135.132.1	remote	255.255.248.0	CoreBuild
140.135.40	140.135.132.2	remote	255.255.248.0	CoreBuild
140.135.48	140.135.55.254	direct	255.255.248.0	CoreBuild
140.135.56	140.135.55.254	direct	255.255.248.0	CoreBuild
140.135.64	140.135.71.254	direct	255.255.248.0	CoreBuild
140.135.72	140.135.79.254	direct	255.255.248.0	CoreBuild

large-scale network with a large number of nodes.

Since most current network equipment supports SNMP instead of a mobile agent environment, agent operations must rely on SNMP as the interface to execute management tasks. Fortunately, new network equipment supports Web-based management using Java Virtual Machine as the management platform. The platform coincides with the development trend of Java-based mobile agents. It is highly possible the system can be integrated in the near future which means performance can be further improved in the following ways:

- Mobile agents can be integrated with the network management platform to improve the data retrieval capacities of data modules, such as MIB-II and RMON MIB.
- In the distributed network management, the subnet hosts can be considered as probes of the network. The probes can precede network management and data collection, and be the execution environment for mobile agents—similar to the RMON probe in SNMPv2, which has higher flexibility for different environments.
- Artificial intelligence can be incorporated into current mobile agents to carry out a distant diagnosis. The distributed management environment can be further developed into a cooperative paradigm. A cooperative paradigm is different from the centralized and hierarchical architectures since it is goal-oriented. That is, the administrator only sets the goal and the agent will do the rest. Intelligent agent technology is one example of this. Furthermore, if network management tools for propriety devices, such as intelligent home electric

Figure 3. The output of MIB ipRouteTable group retrieved by HP OpenView.

The Network Management System

The network management system is a tool for collecting data for managing a network. The network administrator controls, monitors, and manages the network performance throughout the system. This system needs a network management standard, which should be compatible with network hardware and software. Also, in order to exchange information between types of equipment, a communication standard is needed. The well-known TCP/IP (Transmission Control Protocol/Internet Protocol) is the most commonly used protocol on the Internet, and supports the Simple Network Management Protocol (SNMP). The first version of SNMP was published by RFC1157^{*} in 1990 and has now been improved in the form of SNMP Version 2. SNMP provides a simple protocol for the network management system to communicate with subnet types of equipment.

Currently, there are three popular kinds of network management models: the centralized model, the hierarchical model, and the distributed model. The centralized model, which is the oldest framework, relies on the central mainframe to manage the network operations. As has been stated, it is not efficient when the network

nodes grow significantly. An example of a management protocol is SNMPv1. The hierarchical model, which shares management tasks in a hierarchy, is currently the most commonly used model. Examples of this model are TMN, CMIP, and SNMPv2. Although the hierarchical model shares the management load, the management tasks still depend on administrative nodes. This prevents the networks from dynamically adjusting to the environment. In contrast, the distributed model assigns the administrative tasks to local networks. This model can be implemented through either distributed objects or mobile agents. For example, RMON (Remote MONitoring) proposed by the Internet Engineering Task Force (IETF) collects statistics from the subnet hosts to monitor traffic activity. Researchers have proposed a mobile agent-based network management framework with four functions: manager applications, a mobile agent server, a mobile agent generator, and mobile agents [5, 6]. Their framework allows mobile agents to aggregate several MIB values, acquire snapshots of SNMP tables, and filter the contents of the tables. Furthermore, the Perpetuum Mobile Procura Project at Carleton University [10] developed an infrastructure to integrate the mobile agent in SNMP, because SNMP has gathered substantial node information and the mobile agent can provide code mobility, security, and robustness while keeping complexity to a minimum. **C**

^{*}The Request for Comment (RFC), a kind of network document that discusses protocols, algorithms, and system mechanisms, is the most important reference for network researchers and developers. Currently, there are more than 63 RFCs approved by the IETF (Internet Engineering Task Force, www.ietf.org). Examples are: IP (Internet Protocol, RFC791), TCP (Transmission Control Protocol, RFC793), and PPP (Point-to-Point Protocol, RFC-1661).

equipment, can be written in Java, a higher potential for integration can be achieved, rendering a bright future for networked appliance industry. **C**

REFERENCES

1. Anonymous. *Grasshopper Basics and Concepts: Release 2.2*, IKV++ GmbH., Berlin, Germany, Mar., 2001.
2. Baldi, M., Gai, S., and Picco, G.P. Exploiting code mobility in decentralized and flexible network management. In *Proceedings of the First International Workshop on Mobile Agents (MA97)*, Springer-Verlag, Berlin, 1997, 13–26.
3. Cockayne, W.R. and Zyda, M. *Mobile Agents*. Manning, Greenwich, CT, 1998.
4. Fuggetta, A., Picco, G.P. and Vigna, G. Code mobility. *IEEE Transactions on Software Engineering* 24, 5, (1998), 346–361.
5. Gavalas, D., Greenwood, D., Ghanbari, M., and O'Mahony, M. An infrastructure for distributed and dynamic network management based on mobile agent technology. In *Proceedings of the IEEE International Conference on Communications (ICC99)*, 1999, 1362–1366.
6. Gavalas, D., Greenwood, D., Ghanbari, M., and O'Mahony, M. Advanced network monitoring applications based on mobile/intelligent agent technology. *Computer Communications* 23, (2000), 720–730.
7. Goldszmidt, G. Network management views using delegated agents. In *Proceedings of the Sixth IBM/CAS Conference*, Toronto, Canada, 1996.
8. Hagel, J. III and Brown, J.S. Your next IT strategies. *Harvard Business*

Review 79, 9 (Oct. 2001), 105–113.

9. Lange, D. B. and Oshima, M. Dispatch your agents; shut off your machine. *Commun. ACM* 42, 3 (March 1999), 88–89.
10. Pagurek, B., Wang, Y. and White, T. Integration of mobile agents with SNMP: Why and how. Network Operations and Management Symposium (NOMS 2000), J.W. Hong and R. Weihmayer, Eds. IEEE/IFIP, 609–622.
11. Zapf, M., Herrmann, K., and Geihs, K. Decentralized SNMP management with mobile agents. In *Proceedings of the Sixth IFIP/IEEE International Symposium on Distributed Management for the Networked Millennium* (1999), 623–635.

TIMON C. DU (timon@cuhk.edu.hk) is an associate professor at The Chinese University of Hong Kong and a professor at the Chun Yuan Christian University in Taiwan.

ELDON Y. LI (eli@calpoly.edu) is a professor and the coordinator of the MIS program at California Polytechnic State University, San Luis Obispo, CA. He is the executive director of the International Consortium of Electronic Business and the president-elect of the Western Decision Sciences Institute.

AN-PIN CHANG (anping@cycu.edu.tw) is a CNE certified network engineer at the Chung Yuan Christian University Computer Center in Taiwan.