We present here a framework together with a set of rules for real life networks-based active monitoring of network systems. In our proposed framework, intelligent agents are used to perform network monitoring, maintenance, filtering and congestion control functions. Such agents can detect basic events in the network or correlate existing events that are stored in the rule database (RD) to overcome network congestion.

A system administrator can securely modify network monitoring and maintenance policies and remove the problems of the network without human intervention. The framework presented here includes intelligent monitor agents and intelligent inspector agents. The policies and functionalities of these agents can be modified with the help of expanding the RD dynamically. In response to certain trigger events, agents may change their functionalities to correlate the present state of the network. Managing and monitoring large networks with hundreds of computers has become a challenging and tedious task for today's system administrators. A typical computing infrastructure in a medium to large-scale organisation contains many nodes, possibly of different kinds, organised into multiple local-area networks and administrative domains. Administration functions require periodic upgrading of networks, maintaining the overall network and monitoring of user activities at various nodes to defend against potential network threats. The complexity of monitoring large organisational networks with different kinds of hardware/software components frequently added to the environment, or some existing ones periodically upgraded requires new approaches of periodically loading and building network monitoring protocols and functions.

The intelligent agent technology offers several unique capabilities to address the challenges in this area. An intelligent agent technology represents a program database (PD) capable of enquiring the nodes in a network, when it is behaving erratically. Intelligent agent technology performs some designated tasks by sending out a list of questionnaires to such an erratically behaving node. The ability to migrate codes and processing functions to a remote node offers the potential benefits of reduced network overhead and limited system administrator requirements. This article presents a set of paradigms for using intelligent agent technology in installing, monitoring, maintaining, managing and trouble-shooting nodes in an organisation's local networks. Moreover, intelligent agent technology can also provide network security and can stop network intrusions. Our research in the application of intelligent agent technology in real life network systems is motivated by many factors. Many large-scale computing environments, such as university campus networks, tend to be relatively open. It is desired for a system administrator to actively monitor the environment for suspicious activities. Large distributed systems need dynamic and scalable architectures for monitoring. Dynamic structures are needed to support changes to policies for monitoring, collection, and processing of network information at all levels of a system's organisational hierarchy. It should support definition of new event types and installation of specific detection mechanisms at target nodes.

For scalability, the infrastructure should support any desired hierarchical and decentralised organisation for information collection and processing. Moreover, the system should support incorporation of new correlation and search functions across different event databases (EDs). Network installing, maintaining, upgrading, monitoring and trouble-shooting is a promising area where agents can be used to perform remote network monitoring, information filtering, packet correlation, and trouble-shooting control functions. We characterise our agent-based approach as active monitoring because it permits easy installation of new monitoring and information filtering functions by launching agents with new functionality to the network nodes. This paper presents our initial efforts towards building an infrastructure to experiment with future generations of network monitoring and management techniques without human intervention.

Problem formulation:
Following are the major contributions of the work carried out in our laboratory:
• A set of rules for mobile agent-based active monitoring that can be used as building-blocks in a monitoring system.
• A mobile agent-based monitoring framework supporting the various paradigms for agent-based monitoring. This system has been implemented using the mobile agent programming system. Here we have developed some codes in C, to monitor the network with these mobile agents.
• A simple expansion experiment using this framework is presented here to demonstrate the use of mobile agents in supporting active monitoring and dynamic extensibility for detecting compound events. We have carried out the experiment in a real network shown in Fig. 1.

Experiment number 1
We simulate a small office/home office network topology consisting of six computers running...
Windows 2000 Server Edition, as shown in Fig.1, in our laboratory. PC1 acts as a dynamic host configuration protocol (DHCP) server for local area network1 (LAN1) with an internet protocol (IP) address pool from 192.168.10.3 to 192.168.10.8 with 255.255.255.0 as the subnet mask (Class C address). The DHCP server has an IP address of 192.168.10.2 with a subnet mask of 255.255.255.0. All the computers in LAN1 are configured to obtain a dynamic IP address from the DHCP server with fixed IP address of 192.168.10.2.

The default gateway for LAN1 is the e0 interface of a Windows2000 server router with IP address 192.168.10.1/24 (Class C). The e1 interface of the router is connected to a network address translation (NAT) server, to connect the internet (not explicitly shown in Fig.1). The intelligent agent databases (event database and rule database) is maintained with the help of a SQL server running on Windows 2000 advanced server, which is connected to the router.

Now, another domain given by LAN2 is connected to the interface of the router. The expanded disjoined network domain has to be connected to LAN1 through the router. The administrator now defines the following network topology events in ED given below:

- Configure PC1 as a default DHCP server for LAN2.
- Provide an IP address pool at the DHCP server (PC1) starting from 192.168.11.2 to 192.168.11.8 with subnet mask 255.255.255.0 for the computers in LAN2.
- Configure the default gateway for LAN2 as 192.168.11.1 with a subnet mask of 255.255.255.0.
- Install DHCP relay client protocol at the 192.168.11.1 interface.
- Configure each of the PCs in LAN2 to obtain dynamic IP address from the default DHCP server.

The modified network topology is shown in Fig.2.

We defined the set of rules in the RD for loading DHCP relay client protocol at the router interface of LAN2. However, to maintain the network topological problems, we can extend the rules even further.

Conclusion

Our agent-based network framework is designed to support a dynamically extensible environment for monitoring network systems. The purpose and motivation of this framework is to make an attempt to address the issues and requirements for an emerging intelligent network system, where without human intervention, the network systems will co-operate with each other and develop the network as described in the ED by an administrator, using the mobile agent paradigm. In a large system, it may be inefficient to maintain a central database of all events.

Therefore this framework allows one to define any desired policies for decentralised filtering, collection and correlation of ED information, and these policies can be changed dynamically. It facilitates designs of hierarchical and distributed organisations for managing network routing information, but no specific hierarchy is imposed.

It also provides definitions for new events in the system, and installation of appropriate monitoring rules for different nodes in the system. Thus the whole topology and its working rules can be remotely and efficiently maintained at the monitor server.

Acknowledgement

This work was carried out in the Computer Networking Lab of the department under the COSST Project of UGC, Govt. of India.

Contact P Venkateswaran,
pwn@ieee.org or
robnon@ieee.org