Chapter 30
Study of the Hierarchy Management Model Based on Active Network Node

Wenshu Duan, Tianping Dong, Yan Ma and Lunpeng Liu

Abstract Active network can provide a programmable interface to the user where users dynamically inject services into the intermediate nodes. However, the traditional prototype of network management does not accommodate to the management of active networks, it cannot utilize the distributed copulation capabilities that active networks provides. This paper analyzes the structure and mechanism of the active network management system, introduces a pattern of active network management, and studies the structure, management mechanism, design outline, and each connection of the management system. The paper also studies the network topology discovery and traffic.

Keywords Active network · Network management · Active node

30.1 The Management of Active Networks

Due to traditional network management using the centralized management, we manage the network without using the computing power of the active network node. Therefore, the traditional networks neither effectively implement the active network management, nor reflect the advantages of active network. In order to

W. Duan · T. Dong · Y. Ma (✉)
College of Computer and Information Science, Chongqing Normal University,
Chongqing 401331, China
e-mail: mayan@cqnu.edu.cn

L. Liu
College of Physics and Electronic Engineering, Chongqing Normal University,
Chongqing 401331, China
e-mail: 751140151@qq.com

Z. Zhong (ed.), *Proceedings of the International Conference on Information Engineering and Applications (IEA) 2012*, Lecture Notes in Electrical Engineering 219,
DOI: 10.1007/978-1-4471-4853-1_30, © Springer-Verlag London 2013

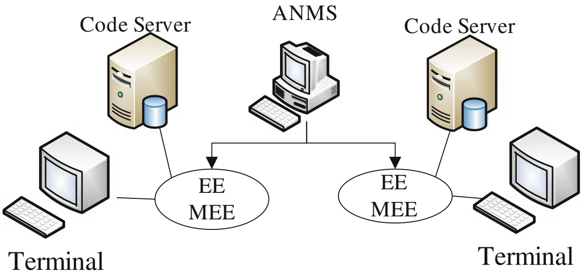


Fig. 30.1 The architecture of active network management system

adapt to the characteristics of the active network, the management model of active network should break through the asymmetric management model of traditional network. Necessarily, active network include the network control, management of workstations, and the active node perfectly, so as to solve the key issues of manager-side in the traditional network. It also loads the dynamic business and manages dynamic MIB. The structure of active network management (ANM) is shown in Fig. 30.1.

In the Fig. 30.1, it is known that active node is the main object of the active network management in the system of ANM [1]. It is an equivalence relation between active node and control management workstation, and instead of the relation between client and server in the simple network management protocol (SNMP). Intuitively, ANMS contains the following features, such as ANMS is an interface of the network managers for controlling and managing active network. Active node is the main management object of network management system that is responsible for handling the initiative letter bag. Execution Environment (EE) provides the environment which must operate and handle the active envelope. MEE represents the overall management functions of active node. Code server (CS) provides the logical method which is necessary for network element equipment to collect data. And the terminal system uses the service of active node to run the active application.

30.2 The Structure of the Active Network Management System

In this paper we propose an active network management model based on the management features and the structural characteristics of the active network. The management model based on the node is the core management, which make full use of the advantages about active network initiative, dynamic, and intelligent to achieve the active network distributed intelligent management.

The structure of this management mode is shown in Fig. 30.2 [2]. In this mode, the management system must complete the node management, configuration, analysis, and monitoring which consists of the network management node and the

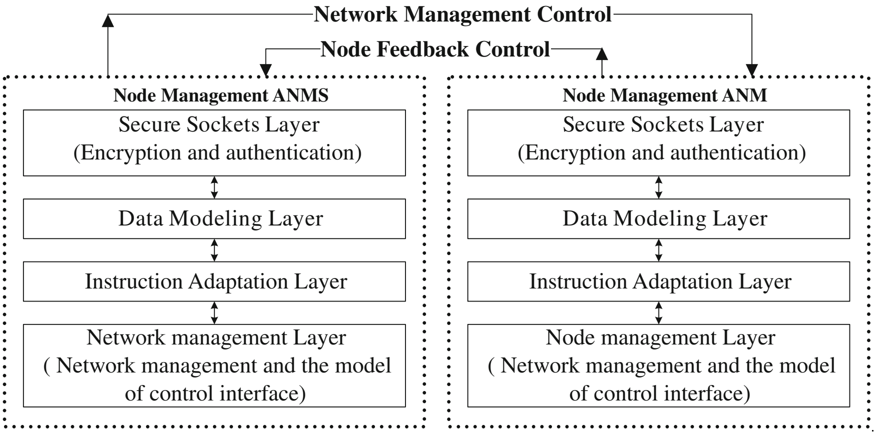

Network Management Control
Node Feedback Control
Node Management ANMS
Secure Sockets Layer (Encryption and authentication)
Data Modeling Layer
Instruction Adaptation Layer
Network management Layer (Network management and the model of control interface)
Node Management ANM
Secure Sockets Layer (Encryption and authentication)
Data Modeling Layer
Instruction Adaptation Layer
Node management Layer (Network management and the model of control interface)

Fig. 30.2 Hierarchical model of network nodes and active nodes

active node, the node management (Local Mgrs SW), the modeling layer (Modeler) and the instructions of the adaptation layer (Instrumentation). Management system will achieve the following functions. For instance, to manage node configuration, failure and performance with the control EE of node, to visit and configurate node by the node OS API issue commands, to provide a set of API interface for EE in order to make active application (App) can adapt and configurate the network resources and monitor the performance of the network dynamically.

30.3 The Structure and Forwarding Mechanism of Management Message

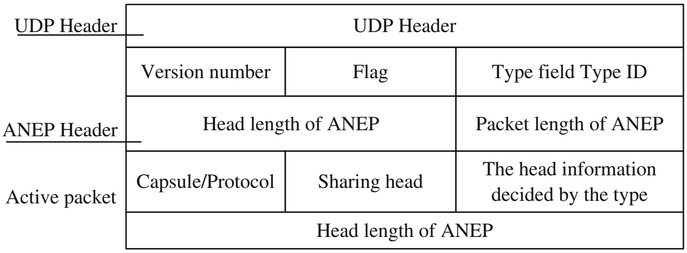
30.3.1 The Structure of the Management Message

In the design of this system, the active packet is encapsulated into UDP and ANEP. The active message consists of the UDP header, ANEP header, active message subject and effective load, which is shown in Fig. 30.3. Next, we will introduce the usage and meaning of fields in the active message subject.

Active message subject follow the construction of ANTS encapsulation body form, and there are several domain in its head.

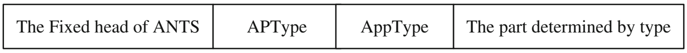
Capsule/protocol: This field is used to describe this text that belongs to the code segment, the code group, and the corresponding agreement.

Sharing head: It contains the source address, the destination address, a node address and version information, and so on, and it is the common domain of the different types of package body.



UDP Header	UDP Header		
	Version number	Flag	Type field Type ID
ANEP Header	Head length of ANEP		Packet length of ANEP
Active packet	Capsule/Protocol	Sharing head	The head information decided by the type
	Head length of ANEP		

Fig. 30.3 The encapsulation format of management packet



The Fixed head of ANTS	APType	AppType	The part determined by type

Fig. 30.4 The format of management packet

The head information decided by the type: Different types of active packet have a different field, and the number and size of the field are not the same.

In this system, we will divide the active messages into the direct implementation messages and active application messages according to the code distribution mechanism of a message. The transmission of the direct implementation messages is “package”, that is to say the small program code is transmitted by the encapsulated messages directly. The active application message becomes complicated and it uses the code distribution mechanism known as on-demand to obtain. The active messages only carry code identification. Therefore, we join two fixed fields in the head information, then the unity format of active message as shown in Fig. 30.4.

APType field: it is used to indicate the type of active message. 0 means direct execution of the messages, and 1 means active application of massages.

AppType field: it indicates that the massage belongs to the application type. For example, 0 means getting ordinary network management message, 1 said common network management message; and 2 denotes patrol message. The purpose of the establishment of the field is to make the active packet which could complete the implementation of the traditional node.

30.3.2 The Forwarding Mechanism of the Management Message

(1) One–One mode forwarding

One–One forwarding mode is one of the simplest forwarding modes. It is divided into two structures according to the package in the section. One structure is similar to the current end-to-end communication mode, which is not to enforce middle node. This approach is mainly used to access to the specified node as shown in Fig. 30.5a. Another structure, as shown in Fig. 30.5b, is a forwarding

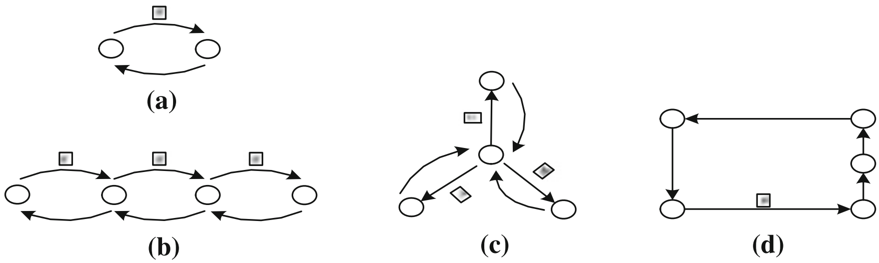


Fig. 30.5 The schematic of One–One mode forwarding, BFST forwarding mode and DFST forwarding mode

mode which is calculated along the transmission path. In this mode, package body packets are necessary to carry the program executed in the node in accordance with a specified intermediate node. Through the implementation methods, management node can put the concentrated tasks into practice along the entire transmission path.

(2) Breadth first search traversing (BFST) forwarding mode

This mode, as shown in Fig. 30.5c, is a kind of parallel control modes. When the message of the package reaches an active node, it is directly sent to the neighbor nodes which connected to the current node. The same implementation will be playback when the message reaches the adjacent nodes. Obviously, the network will come up a lot of the copies of the encapsulation body message after one transfer. When these copies to reach the next node, they are copied and forwarded to their neighbor node, then a copy of the message is in turn forwarded continue until the messages traverse the entire network.

(3) Depth first search traversing (DFST) forwarding mode

This mode, as shown in Fig. 30.5d, is a kind of serial control modes. In this mode, the package message is directly forwarded to a neighbor node which connected with the current node, when it reaches an active node. Then the message was forwarded to a neighbor of the neighbor in turn until the messages traverse the entire network.

30.4 The Analysis of Active Packet Path Forwarding Algorithm

The network is defined as the connectivity graph $G = (V, E)$ [3], where V represents a set of nodes, E represents the two-way connection between the nodes. Every active node consists of fast forwarding(FF) unit and EE, etc. All packets have delay in forwarding when they go through each hop, and the forwarding

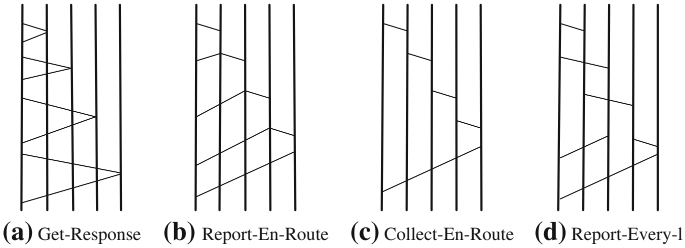


Fig. 30.6 The schematic of active packet forwarding mode

delay includes the propagation delay and queue delay. Some packet may be processed by EE, so this packet also includes processing delays. If a communications link treats the pack using FIFO order, the service program of EE also follows FIFO order. The head of a packet mainly contains the source, purpose, application identifier and other information. FF matches packet headers through a set of Filters, if there is a match then turn it over to EE processing, or directly forwarding to the destination address. However, delays of each package are limited on the node. Simply, we assume that FF delay is banded by constant C, and the executable code delay in EE is defined as function P (*k*). Then, C and C + P (*k*) represent the delay for forwarding packets on the node and the delay for executing in EE, respectively [4].

Here, we ignore the cost of the transfer of programs, and we could assume that most of the active network code can be obtained from the node cache [5]. Because of the node in the treatment could copy the package in order to send messages to the EE, the function P (*k*) mainly depends on the package in the EE calculation, so P (*k*) must be a linear function at least [6]. And we have the assumptions as following:

$$\mathrm{P}(k) = \mathrm{P_C} + k\mathrm{P} \tag{30.1}$$

In the above formula, *k* means the length of the packet and P represents a constant. Normally, we can ignore the $\mathrm{P_C}$, so the above equation could simply rewrites as:

$$\mathrm{P}(k) = k\mathrm{P} \tag{30.2}$$

In order to analyze the performance, we define the following concepts.

TC(*n*): time complexity. To measure the time of a task from beginning to end.
MC(*n*): message complexity. To measure the node number of the active packet in a task.

In the structure represented in Fig. 30.6, Link-A is seen as the nodes of a management center and the algorithm injected into the corresponding active packet.

Next, we will analyze several algorithms used in this paper.

(1) The Get-Response is similar to the Request-Respond in the SNMP.

In the Get-Response algorithm:

$$\mathrm{TC}(n) = n\mathrm{P} + \sum_{i=1}^{n-1} 2i\mathrm{C} \tag{30.3}$$

$$\mathrm{MC}(n) = \sum_{i}^{n-1} 2i \tag{30.4}$$

where nP is the action delay of n tasks executed in the EE, and $2i$ means the time delay for $i = 1, 2, 3, \ldots, n-1$ hop.

(2) The Report-En-Route is a forwarding request to the next node when the request which reached to the node sends the response to the source side.

In the Report-En-Route algorithm:

$$\mathrm{TC}(n) = 2n\mathrm{C} + n\mathrm{P} \tag{30.5}$$

$$\mathrm{MC}(n) = n + \sum_{i=1}^{n} i \tag{30.6}$$

where nP is the time delay in the implementation of all the nodes, and $2n$C means transmission delay that includes the time cost in sending to the destination and the returning to destination.

(3) The Collect-En-Route algorithm is that the request which arrived at node carries the response information to the next node, and the request will directly returns to the source when it reaches the destination node.

In the Collect-En-Route algorithm:

$$\mathrm{TC}(n) = 2n\mathrm{C} + \sum_{i=1}^{n} i\mathrm{P} = 2n\mathrm{C} + \frac{n(n+1)}{2}\mathrm{P} \tag{30.7}$$

$$\mathrm{MC}(n) = n \tag{30.8}$$

The iP is the time delay in the implementation of node. Because of the length of the package will add a unit after a hop, obviously, the biggest message complexity of the node n does not exceed $2n$.

(4) The Report-Every-l algorithm is a compromise definition of Collect-En-Route algorithm and Report-En-Route algorithm. The time complexity and message complexity of Report-every-l algorithm are appropriate.

The thought of Report-Every-l algorithm is described as that n will be divided into n/l section, each section length is 1, and we would send a fixed size massage which is initialized to collect-en-route algorithm to all the n/l section. So, the section i started in Collect-En-Route until cost more than the $(i - 1)$ (C + P) time unit.

So in the Get-Response algorithm:

$$\mathrm{TC}(n) = (n - l)(\mathrm{C} + \mathrm{P}) + \sum_{i=1}^{l} (\mathrm{C} + i\mathrm{P}) = \mathrm{O}(n\mathrm{C} + (n + l^2)\mathrm{P}) \tag{30.9}$$

$$\mathrm{MC}(n) = O(n) + \sum_{i=1}^{n/l} (l + il) = O\left(\frac{n^2}{l}\right) \tag{30.10}$$

We assume $l = \sqrt{n}$, then the TC (n) is linear, and the message complexity is $O(n\sqrt{n})$.

In order to keep balance in the two complexities, we make $l^2 = {}^{n^2}\!/_{l}$, and then the two complexities are $O(n^{3/4})$.

30.5 Conclusion

This paper discussed the active network management model which has independent module and accurate task. And each layer can be dynamically updated to adapt to the volatility of the active node in the active network and the expansion of the active application. Ultimately, the stability and scalability of network management are improved obviously, and the active network management meets the needs of the modern network management commendably.

Acknowledgments This work is sponsored by Chongqing Higher Education Educational Reform (NO: 102118) and Chongqing Education Committee (KJ090809).

References

1. Di Fatta G, Gaglio S, Lo Re G, Ortolani M (2000) Adaptive routing in active networks. IEEE Openarch 2000, Tel Aviv Israel, 23–24 Mar 2000
2. Di Fatta G, Lo Re G (2001) Active network. an evolution of the internet. In: Proceedings of AICA 2001—39th annual conference, Cernobbio, Italy, 19–22 Sept 2001
3. Munir S (2000) Active networks: a survey[EB/OL].http://www.cse.ohio-state.edu/jain/cis788-97/ftp/activenets/index.htm, 2000-07-02
4. Al Shaer E (2000) Active management framework for distributed multimedia systems. J Netw Syst Manag 8(1):49–72
5. Brunner M, Stadler R (2000) Service management in multi-party active networks. IEEE Commun Mag 38(3):281–286
6. Calvert KL (1998) Directions in active networks. IEEE Commun Mag 36(1):72–78