IOSR Journal of Engineering (IOSRJEN)
e-ISSN: 2250-3021, **p**-ISSN: 2278-8719
Vol. 3, Issue 7 (July. 2013), ||V2|| PP 34-42

"Performance Analysis of Active Network Service Composition"

Mohammad Masudur Rahman
Department of Business Administration, East West University, Bangladesh

Abstract: **-** Active Networks is novel approach to networking, which provides an architectural support for dynamically deploying new protocols in an existing network topology. The nodes in an active network are able to download and execute customized user codes on themselves and thus rendering the node recognize and run totally new protocols without making any changes to the architecture of the network. This research implements and tests such specialized Active Networks security service known as the firewall and the ping service in Active network. Active network environment will be implemented on a small scale test scenario, in order to study the performance and characteristics of active networks.

Key Words: *Active network, Firewall, Ping, Security Service.*

I. INTRODUCTION

Active networking started within the Defense Advanced Research Projects Agency (DARPA) between 1994 and 1995. The idea was to find a new direction in solving some of today's networking problems such as difficulty in the integration of new technologies, poor performance due to redundant operations at certain protocol layers, and difficulty in integrating new services within an existing infrastructure [1].

Active networking is meant to reduce these problems by creating a virtual platform which is flexible, adaptive and extensible in nature, and one which will allow, amongst others, the rapid and distributed deployment of new services, according to user and/or application requirements efficiently.

Conventional networks passively transfer data from source to destination node, without changing the user data. This implies that the user data is opaque to the system and is never modified by the system. Computation within such a system is extremely limited, as the header of a protocol data unit (PDU) provides all the necessary information to ensure that the data is sent to the correct address. The actual contents of the data are ignored, and the amount of computation within the network is restricted to header processing and signaling (for switched circuit networks).

Active network is a novel approach to network architecture in which, nodes may be programmed to the user and/or application requirements and is therefore more dynamic and flexible in nature. This may be accomplished by injecting small or mini programs at certain 'active' nodes as packets are transmitted over the network. Tennenhouse [2] explains the concept of an active network as one where nodes – the switches, routers, hubs, bridges, gateways etc, - have the ability to allow customized computations to be performed on the user data. This kind of network is called active because new computations can be dynamically injected into the network, thereby allowing the network to behave differently with different data types. Packets in an active network (also known as capsules) carry mini programs, in addition to the original data.

Replacing passive packets with capsules that are active means that new services may be added to the network, or existing services may be enhanced/customized, independent of the underlying infrastructure, as and when necessary. Also, specific functions according to a particular application may be deployed at certain locations within the network. Some immediate applications of active network technology include firewalls, web proxies, mobile computing and application services [1]. However, the potential uses of this sophisticated technology may yet be unexplored.

II. OBJECTIVES

To resolve the issues of efficiency and fairly managing both the processing and bandwidth resources in programmable active networks and providing better QoS guarantees to the competing flows, this research work has been focused on firewall and ping services implementation on active network. This research will pursue the following objectives.

1. Implement and test such specialized Active Networks security service known as the firewall and the ping service.
2. To enable the rapid deployment of new services onto the network with greater ease;
3. To enable user-controllable functionality to be added to the network;
4. To facilitate the enhancement/customization of the network according to user/application requirements dynamically.

III. LITERATURE REVIEW

The following list summarizes some potential applications of active networking in today's environment, as mentioned by Tennenhouse and Calvert [1], [2] and [3].

Firewalls filter packets that enter a particular domain based on certain application- or user-functions. The use of active networking will automate this process by allowing only approved user/applications to authenticate themselves at the active node (firewall). In mobile computing, active networking may be used to adapt and adjust the mode of transmission or activate certain functionality according to the type of connection of the end system to the network. If, for example, the end system is connected by a low bandwidth link such as a phone line, then the active node may do more file caching or link compression. In addition, certain security features may be invoked if the end system is linked by an insecure/external connection. Active networking may be used to distribute the responsibility of multicast re-transmission from the original sender to other receivers down the multicast tree. Here, all receivers are active nodes and are aware of the states of other (or nearby) receivers, and will be able to provide retransmission in the event of transmission loss – thereby removing the load of re-transmission from the original sender, reducing delay and increasing reliability. Active networking will allow state and processing to be done within the network. In ensuring guaranteed quality of service over networks, many approaches require that the originating node detect the conditions of the network and adapt accordingly to prevent congestion or loss during transmission. In such cases, the inherent delays between detection and adaptation of the sender, along the path of transmission, may degrade the performance of the network especially over long links. It is therefore desirable to distribute the state of awareness and adaptation to other nodes within the network. Thereafter, the network would be able to adjust to the correct mode of adaptation as when necessary. Web caching is usually done to reduce delays and congestion when a web application is accessed. Objects are normally cached from servers to locations close to clients using a manually configured static hierarchy. In dynamic conditions, active networking may provide more efficient means of caching. This includes adaptive web caching [4],[9] using small caches with information on neighboring nodes' contents [5], and using an active network to route cache requests to pre-configured locations [6]. The usual method of polling and checking for abnormalities restricts the management intelligence of the network to certain management stations only. This causes delays before the anomaly can be rectified. Using an active network approach may improve the management capabilities of a network since each node may be programmed to be aware of and adaptive to the state of the network. Projects in this area include Netscript at Columbia [7], and Smart Packets at BBN Systems and Technologies [8].

IV. METHODOLOGY

The active network architecture (which is under development) discusses the common basic functionality of the active node programming interface – the part which is visible to the end user. Here, matters pertaining to addressing, end-to-end services, capsule/packet processing, and node resources are defined generically, without specific reference to any language, under the DARPA active networks program [1].

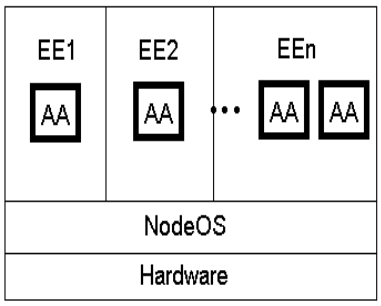

Fig.1: Proposed Active Network Architecture.

As seen in figure 1, the common functionality of active network architecture is separated into two entities, namely, the execution environment (EE), which is similar to a shell program that acts as an interface providing end-to-end services, also known as active applications, AA to users, and the node operating system (NodeOS), through which the EEs draw the necessary resources at the node. These resources include, amongst others, transmission, bandwidth, computing and storage. Implementation of active network API is handled by the EEs, whilst the NodeOS is responsible for managing access of the EEs to these local resources, hiding their details as well as the existence of other EEs in the node. An execution environment may provide only a basic service, controlled by parameters provided by the user, or a more extensive one, such as that of a powerful interpreter - depending on the implementation of the EEs.

V. RESULT & DISCUSSION

Network Topology under Test for firewall

The testing environment for firewall service has the following components:

1. Number of nodes: 16 nodes (from 0 to 15)
2. Data rate from 9 to 6 and 2 to 6: 10KB/s
3. Type of link between nodes: Duplex Link
4. Intra-Domain Bandwidth: 10Mbits/s and 56Mbits/s Fig.
5. Intra-Domain prop Delay: 5ms and 35ms

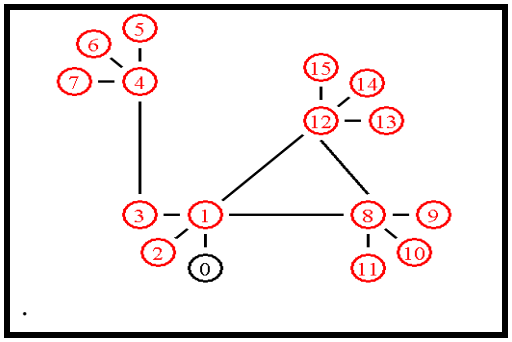

Fig. 2: A small scale topology

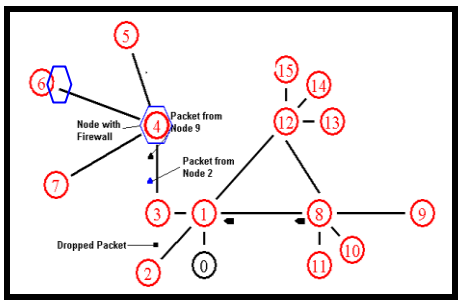

Fig.3: Operation of Firewall Service in NS-2

Figure 3 shows the operation of active network firewall. Node 6 has been equipped with the firewall capsule, which is why it is represented by a blue hexagon around it. The black packets or unwanted packets are being sent from node 9 to node 6 and the blue packets are being sent from node 2 to node 6. What I can observe from figure 3 is that the black packets of node 9 are being dropped, whereas the blue packets of node 2 are being received by node 6, since the firewall of node 6 blocks incoming traffic from node 9.

Several graphs were plotted based on the results obtained by simulating topology of figure 2. The first and most important parameter to analyze is the throughput of figure 4. Figure 4 shows the total number of packets generated at all nodes. I can see that a maximum of 55 packets were generated at any particular time.

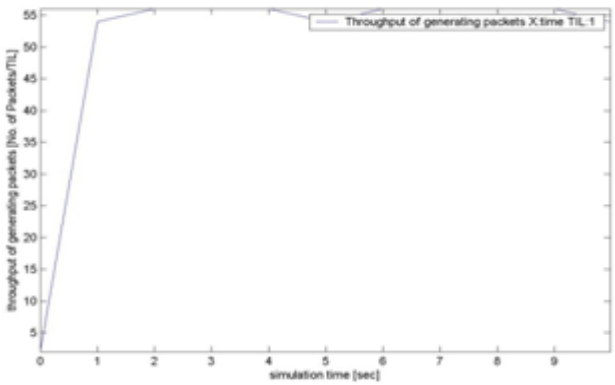

Fig. 4: Throughput of generating packets

Figure 5 shows the total number of packets generated at all nodes. I can see that around 30 packets were dropped. This shows that only packets from node 2 were allowed and the packets of node 9 were dropped.

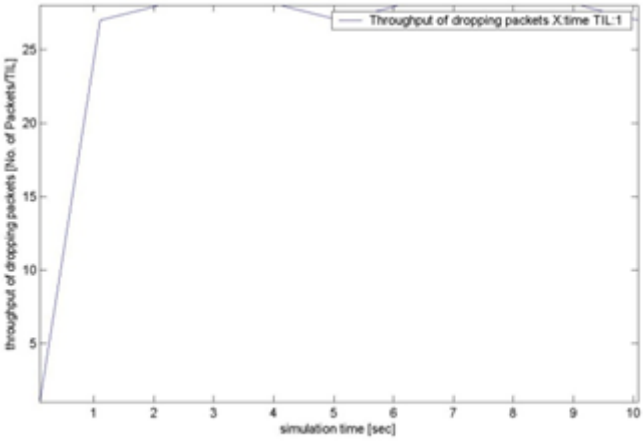

Fig. 5: Throughput of dropping packets

Figure 6 shows the total number of packets generated at the sending nodes together with their destination nodes. Figure 6 also shows that node 2 generates packets for node 6 and node 9 also generates packets for node 6.

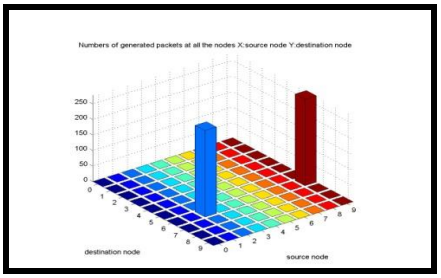

Fig. 6: Packets generated at nodes 2 and 9

Figure 7 shows the total number of packets received together with the source node. Figure 7 also shows that packets from node 2 are accepted by node 6 and packets from node 9 are rejected.

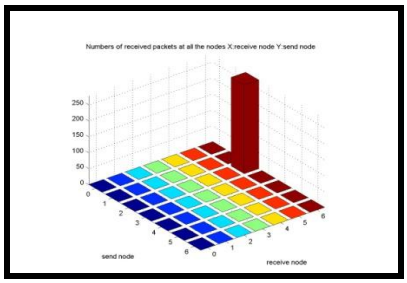

Fig.7: Packets accepted by node 6

Active network allows special packets to be treated differently by executing special programs at nodes. These specialized nodes then have the ability to cater the needs of the special data packets. The ability to execute and especially cater for certain packets gives active network the ability to dynamically develop and test new protocols.

Result and discussion for Ping

The Ping stands for Packet Internet Groper which command is used to verify that a network connection exists between two computers. Ping command always gives the following statistics for each host:

- Number of sent/received packets for this host
- Loss percentage
- Minimum/average/maximum RTT as measured from these packets

For small topologies I choose hand built network or for large networks choose a predefined binary tree network as well as the deployment mode.

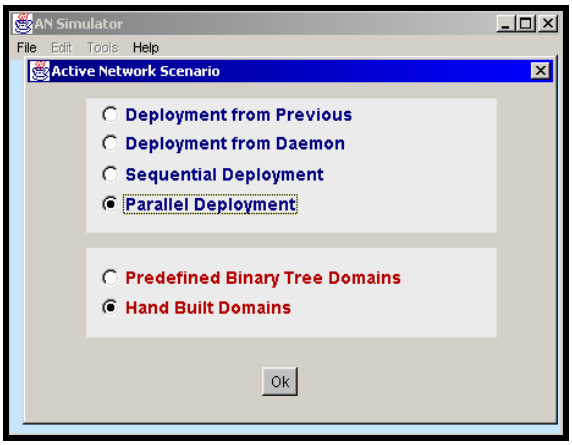

Fig.8: AN Scenario

From figure 8, I can see that there is an "Active Network Scenario" and I choose parallel Deployment and Hand build Domains because of small network topology.

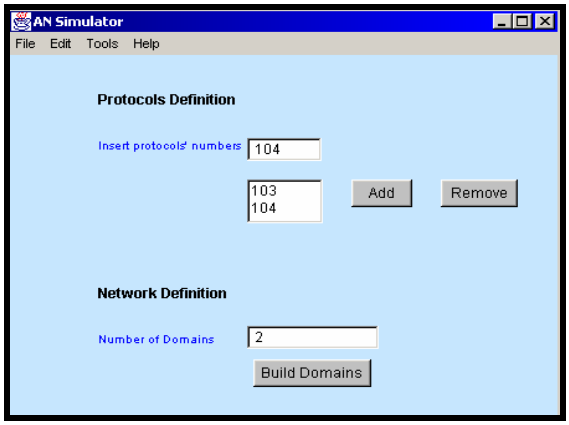

Fig.9: Active network Simulator

In figure 9, this network scenario uses two protocols namely 103 and 104 with two domains. Specifically the number of Hosts per domain, the domain Adjacency Matrix, and the external connections specifying what nodes interfaces connect to the external port of the domain. Other parameters like the domain network address and network mask as well as the control node of the domain are also specified for building a domain.

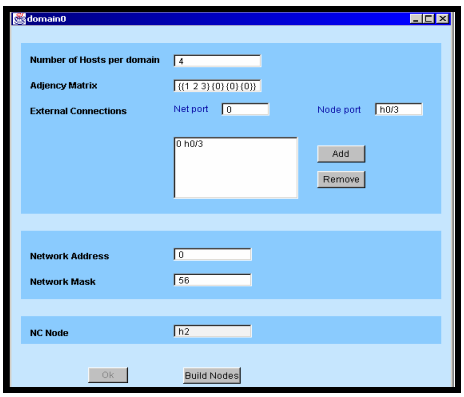

Fig.10: Domain 0

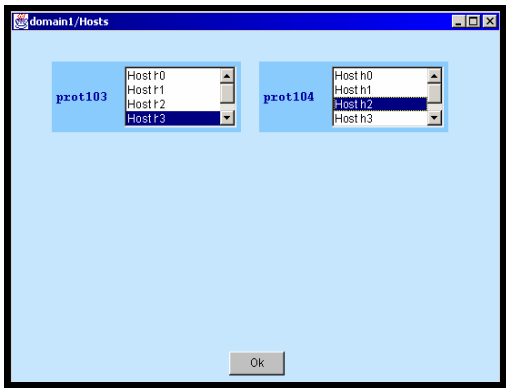

Fig.11: Domain1/Hosts

Figure 11 shows that protocol 103 and 104 are just simple examples of active applications. The protocol 103 functions in fact like a **Ping**, where the sending node sets a timer and waits for a response before running out of the timer. Intermediate nodes will simply route the packets from source to destination by executing the code associated to the protocol 103.

The protocol 104, functions like a **Ping** without return, so when the packet reaches its destination a message 'Hello World' is sent to its output port. In order to be able to send packets of the protocol 103 from the node domain0/h3 to the node domain1/h1 and packets of the protocol (prot104) from the node domain1/h2 to the node domain0/h3, the control node h2 of the (domain0) should possess the protocol 104 and the control node h3 of (domain1) should possess the protocol 103.

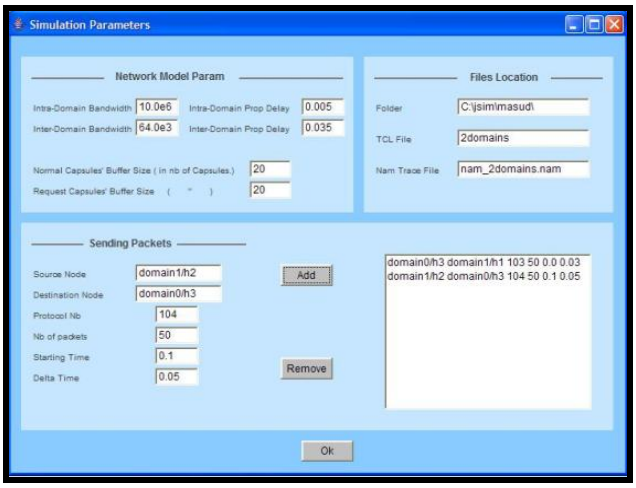

Fig.12: Simulation parameter

In figure 12, Simulation Parameter frames which contain Intra-Domain Bandwidth, Intra-Domain Prop Delay, and Normal Capsules' Buffer Size and request Capsules' Buffer Size are shown.

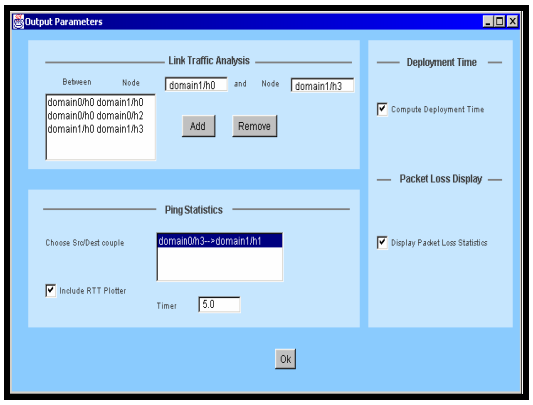

Fig.13: Output parameter

In figure 13, Output parameter is created and I have to choose Deployment time, Packet Loss and Round Trip Time (RTT plotter).

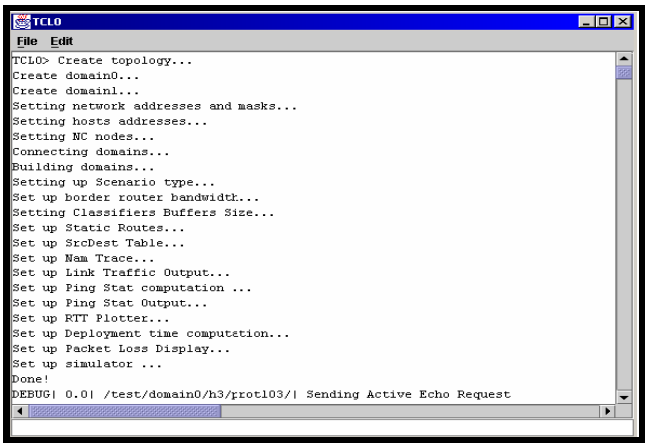

Fig.14:TCL0

Figure 14 show that when traffic begins to flow, this same window can be used for monitoring in order to trace packets.

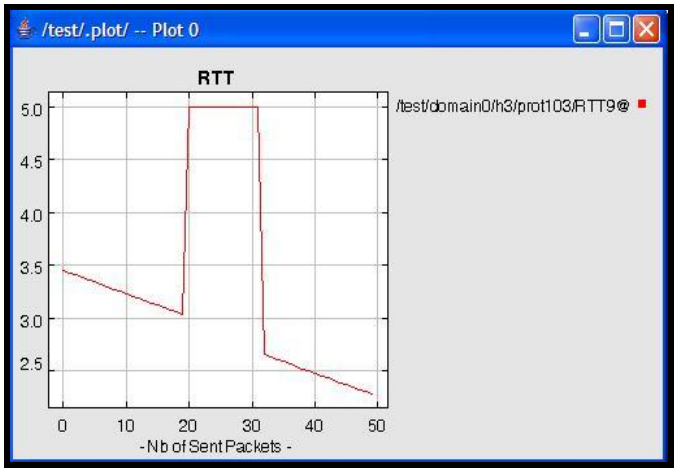

Fig.15: RTT plotter

On the graph that packets from 20 to 31 have been lost. Their lost is represented by a RTT equal to the **Timer** fixed as a parameter. These losses are due to the saturation of the normal capsules buffer. This RTT plotter shows up in real time and traces the Round-Trip-Time as a function of the number of the transmitted packet between a source and a destination node.

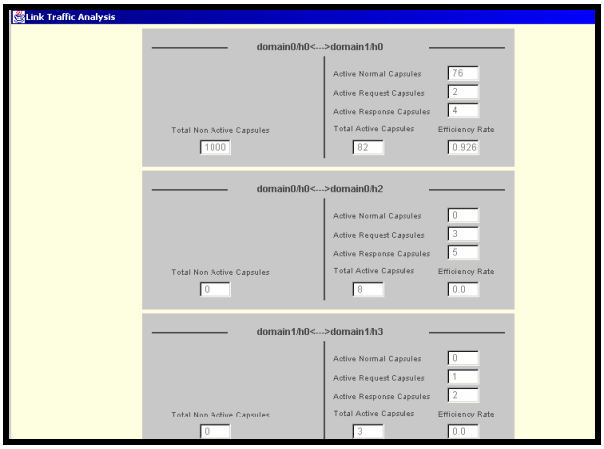

Fig.16: Link traffic analysis

In figure 16, 0utput parameters window, 3 links has been chosen.

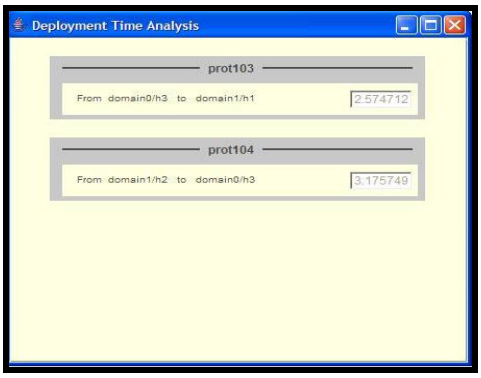

Fig.17: Deployment time analysis

Ping Statistics

This frame displays for each packet of the protocol 103 it's RTT, the maximal, minimal and average RTT as well as the percentage of lost packets. Results are available for every couple source & destination.

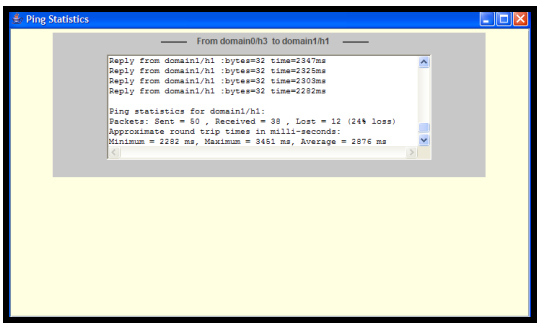

Fig.18: Ping statistics

From the figure 18 I can see the ping statistics. Total 50 packets are sent, 38 packets are received and 12 packets are lost.

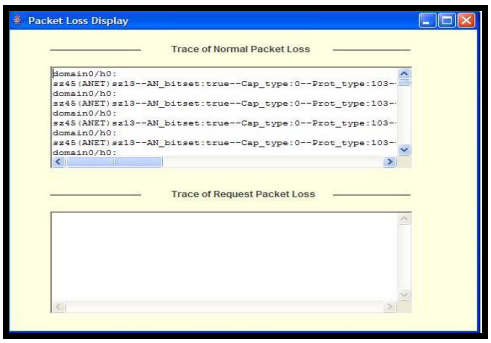

Fig.19: Packet loss display

Figure 19 show that trace of normal packet loss.

VI. CONCLUSION

In this research I have demonstrated that Firewall and Ping service can be successfully implemented a specialized Active Network security service. The test results indicate that this is a value option in Active Network security implementation. This research implements and tests one such specialized Active Networks security service known as the firewall. Usual firewalls run on network gateways as either software or hardware. In my experiment, I designed a firewall such that it runs as an active network service so that each node on the network can download this particular service for self-defense. This research motivates the need for an architectural framework for the parallel deployment and Hand Builds Domains of active network service in a small scale network environment.

REFERENCES

[1] Tennenhouse D. L., Smith J. M., Sincoskie W. D., Wetherall D. J., and Minden G. J. "A Survey of Active Network Research" IEEE Communications Magazine, January 1997, pp 80-86.
[2] Tennenhouse D. L., and Wetherall D. J. "Towards An Active Network Architecure" Computer Communications Review, 26, 2 (1996), pp 5-18.
[3] Calvert K. L., Bhattacharjee S., Zegura E., and Sterbenz J. "Directions in Active Networks" In IEEE Communications Magazine, October 1998, pp 72-78.
[4] Zhang L., Michel S., Nguyen S., Rosenstein A., Floyd S., and Jacobson V. "Adaptive Web Caching: Towards A New Global Caching Architecture" In 3rd International WWW Caching Workshop, 1998.
[5] Bhattacharjee S., Calvert K. L., and Zegura E. W. "Self-Organising Wide-Area Network Caches" In IEEE Infocom '98, 1998.
[6] Legedza U., Wetherall D. J., and Guttag J. "Improving the Performance of Distributed Applications Using Active Networks" In IEEE Infocom '98, 1998.
[7] Yemini Y., and Silva S. da, "Towards Programmable Networks" In IFIP/IEEE International Workshop on Distributed Systems: Operations & Management, L'Aquila, Italy, October 1996.
[8] Schwartz B., Zhou W., Jackson A., Shayer W. T., Rockwell D., and Partridge C. "Smart Packets for Active Networks" BBN Systems and Technologies, http://www.nettech.bbn.com/smtpkts/smart.ps.gz, 1998.
[9] Legedza U., Wetherall D. J., and Guttag J. "Improving the Performance of Distributed Applications Using Active Networks" In IEEE Infocom '98, 1998
[10] http://www.j-sim.org/start.html
[11] http://www.isi.edu/nsnam/ns/