Design of a Distributed Active Network Toolkit

P. Jayashree, K.S. Easwarakumar[1], M. Chandrasekar, P. Ramya and M. Vijay

Department of Information Technology, Anna University, MIT
[1]Department of Computer Science and Engineering, Anna University, CEG

ABSTRACT

Active networks add flexibility in deploying networking applications by enabling the routers to be active. The routers can perform any customized processing as specified by the user/application on the packets traversing through them rather than just predefined routing functionality. ANTS is a simulator used for testing active network applications. The existing architecture of ANTS enforces restriction on the number of active nodes that can be instantiated without substantial performance degradation. For real time testing in Active Networks, a reasonable size topology need to be defined and hence is the necessity for ANTS to be extended. In this work an attempt is made at enhancing ANTS simulator to support distributed deployment by creating different ANTS run time environments in different systems and by enabling communication and coordination between them using graph theoretical properties. No performance degradation is noticed compared to simulation in a single system and the solution deployed at application level is generic and can be ported to any network simulator.

Keywords: *Active network, ANTS, distributed system, graph partitioning, synchronization.*

1. INTRODUCTION

Active Networks differ from traditional networks in that user defined programs can be executed within the network so that the processing of packets is specific to the user/application. ANTS (Active Node Transfer System) is a simulator toolkit used for simulating active network topology. ANTS supports a maximum of ten active nodes to be defined in network topology with acceptable level of performance characteristics. In this paper a design solution is proposed and discussed to develop a distributed or networked version of ANTS test bed. The deployment issues in distributed environment such as interoperability and time synchronization are addressed. Solutions to overcome these issues are discussed without any performance degradation. Two models to support distributed simulator implementation have been proposed and discussed.

2. ACTIVE NETWORK TOOLKIT

In Active Networks the switches are sensitive to the data transmitted through them and perform computations in them. The packets that are transferred through Active Networks consist of code to be executed at each node and the actual data to be sent. They are processed and can be changed on their ways to the destination. The main objective of Active Networks is to enable programming in network infrastructure to promote deployment of new protocols dynamically.

ANTS(Active Node Transfer System) is a toolkit for simulating Active Network topology. ANTS promotes new protocols to be deployed dynamically. It expresses new protocols in terms of computations to be performed at nodes[6]. The packets are substituted by Capsules. They consist of both data to be transmitted and code segment to be

executed at each active node. Each node here consist of programming environment to perform computations on the capsules, a soft cache to store the code segment for the processing of subsequent capsules of the same type. So the capsules can consist of either the code segment or a pointer to the code already cached in the node. Each node should be able to manipulate capsules, execute control operations on the capsules and access the routing table and other information. Initially the code group to be executed for a particular type of capsules is distributed across the nodes. When the code is not available in the node it is obtained from the previous node[12,13]. When a capsule of particular type arrives, the corresponding routine is found and cached in order to enable fast processing of subsequent capsules of the same type. For this Demand Loading Protocol and Code Distribution Schemes are used.

2.1 Limitation in ANTS

With the existing architecture of ANTS a maximum of 10 nodes can be simulated without considerable performance degradation. This is due to increase in need for memory and other resources for each node and so the overhead increases and the performance decreases.

2.2 Need for Distributed Simulation

For any real time testing in active network using a reasonable size network topology, the simulator should have the support for defining the topology. ANTS on the other hand supports a maximum of 10 nodes in an instance of ANTS. Developing a distributed simulation environment in ANTS provides the solution for the same. To incorporate the distributed simulation, ANTS[9,10] need to be extended with provisions for optimal routing and other interoperability issues. The distributed environment is created by partitioning the nodes of the network topology into sub networks and enabling communication and coordination between them .

3. ISSUES TO BE FOCUSED

The distributed simulation environment is created by partitioning the network topology into many groups of nodes called domains and by simulating each domain in different systems using ANTS. Any distributed simulation has pitfalls like

- Decrease in simulation capacity
- Increase in communication cost across domains
- Increase in the network load
- Increase in delay
- Additional cost in synchronization of nodes across domains.

The above issues are addressed and solutions are proposed for an efficient distributed simulation in ANTS.

4. DISTRIBUTED SIMULATOR DESIGN

The distributed simulator addresses the issues related to partitioning the network topology into domains, providing synchronization across domains and other general interoperability functions.

4.1 Topology Partitioning

The network topology is defined as a connected graph. The nodes are partitioned into sub domains and configured in different systems using efficient graph partitioning methods. The partitioning of nodes is carried out in view of the following demands.

- Minimum cross-machine communication to ensure minimum cost for communication across the domains and hence to increase the simulation capacity
- Load balancing in each machine to reduce the load on the network
- Compatible link capacity provision when edges are split across domain.

4.2 Time Synchronization

When the capsules are transmitted from one domain to other, as they traverse through underlying [1] TCP connection associated with the real network, a real time delay is associated with the capsules. In order to combat the real time delay, time synchronization is done by attaching a time stamp

field to the capsules in case of cross-domain communication.

4.3 Interoperability

Interoperability[5] for achieving effective communication and coordination between domains is addressed in terms of physical connectivity, logical connectivity and routing across domains.

4.3.1 Physical connectivity

The sub topology for each domain is configured in computer systems individually. Nodes in that system with their IP addresses, the connectivity between them, the application deployed in each node, the source and target addresses and global routing table for the entire topology. This approach of storing the node configuration in individual domains is beneficial because of low memory requirements and low overhead in managing information. So the configuration files, overall routing table and the local routing information are maintained locally for each domain in each system.

4.3.2 Logical connectivity

For nodes distributed across different domains is established using TCP/IP protocol mechanism. TCP servers are run in all the remote endpoints. Whenever there is a need to transmit capsules or state messages TCP client connection is initiated to an appropriate remote endpoint. Every connection is identified with an IP address and a port number for cross domain communication.

4.3.3 Inter-intra domain routing

Global routing table is maintained in each domain. Local routing table is stored in each node to enable routing of capsules in each domain. In each ANTS domain two extra dummy nodes are added similar to ghost nodes dedicated for routing in case of cross domain communication. This node termed as connection router acts as a source and sink for the domain. All capsules that are destined to an active node in different domain are routed to the connection router. The time stamp is attached to the capsules and then a client TCP connection is initiated to the connection router in destination domain. Then the capsules are routed through TCP stream to the destination domain. Its main purpose is dedicated routing in case of cross domain configuration. The connection router deployment is decided based on

- Connectivity to every node in the domain leads to other domain
- Compatibility in link capacity
- Accessibility to other connection routers in other domains

The capsules that are destined for a node in different domain are routed to the connection router and that routes them to the destination node. The partitioned topology with their connection routers are shown in figure 1.

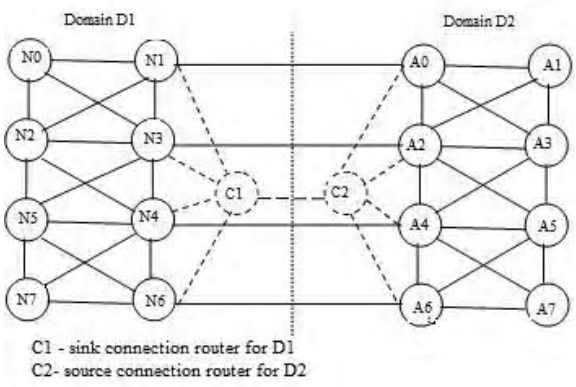

Fig. 1: Topology partitioning with connection routers

5. DISTRIBUTED SIMULATION MODEL

In this section two models for implementing the simulator ANTS in distributed environment is presented namely peer-peer model and client-server model.

5.1 Peer-Peer model

In Peer-Peer model, each sub domain is configured in different systems and all domains are considered as peers to each other. Each domain consists of the nodes defined in the sub topology and two additional nodes acting as connection routers. Every active node in a domain that needs to communicate with a particular node in another domain need to

send the capsules via connection routers. The connection router checks each and every capsule it receives for the destination whether the corresponding destination. If it belongs to the same domain it performs conventional ANTS routing. If not it simply transmits the capsules to the connection router of the destination domain which in turn delivers it to the appropriate destination. Here the number of nodes supported increases linearly. The IP addresses for the connection router of all the other networks must be maintained in every domain.

5.2 Client-Server model

In this model each domain is configured in different machines and one ANTS domain is treated as a Server domain. Other domains act as Clients. Each individual ANTS domain consists of active nodes and a connection router. The server domain connects different clients who need to communicate between each other. Every client which sends capsules that is destined to a different domain is routed to the connection router in the server domain. It further routes the capsule to the desired destination domain. A node in server is fixed to receive capsules from different clients. And different nodes are defined as routers for sending the capsules to different clients i.e., one fixed connection router for each client. This information is maintained in the server itself.

Every node that needs to communicate with a particular node in other ANTS domain sends the capsule to the connection router. The connection router checks each and every capsule it receives whether the corresponding destination belongs to the same domain or to different domain. If it belongs to the same domain it performs conventional ANTS routing. If not it sends the capsules to the connection router of the server domain. The connection router in the server domain receives the capsule and passes to the router in the server domain responsible for the destination domain. And that routes to the connection router of the destination domain. Based on the destination address in the capsule it delivers to the appropriate receiver. In this model each domain should know the address of the connection router in the server and the IP address of the connection routers responsible for sending to other domains. Here the number of nodes increases proportional to the number of clients.

6. IMPLEMENTATION STRATEGIES

6.1 Assumptions

All the systems used for implementing distributed or networked simulator are assumed to be robust and have uniform architecture. During topology partitioning the edges across domains are split into a sequence of edges via connection routers. The sum of link capacities is maintained to the same as the original link capacity.

6.2 Timestamp

In order to achieve time synchronization across the domains a timestamp is added to the capsules at the connection router in case of cross domain communication. This timestamp field is retrieved and the time in receiving machine is synchronized to it.

6.3 Capsule format

In addition to the fields in the original capsule format in ANTS[2,3,6], two more fields are added as depicted in figure 2. The timestamp field is used for time synchronization across domains and the domain IDs field is added to identify the destination domain in both the proposed models.

Fig. 2: Capsule Format

6.4 Routing

The routing strategy involved in both the models is given in the following pseudo code listing.

6.4.1 Peer-Peer model- Routing strategy

Forwarding process flow at any node

```
/*global routing table is followed*/
retrieve next hop address using global routing table
if (intra domain)
```

use default shortest path routing
else if (inter domain))
route to sink connection router of current domain
At sink connection router at same domain
find the destination domain
add timestamp field to the capsules
forward[4] to source connection router at the destination domain
At source connection router in destination domain
receive capsules from sink connection router of source
domain
reset time to the timestamp value in the capsules
retrieve destination address
forward directly to the next hop address

6.4.2 Client-Server – Routing Strategy

Forwarding process flow at any node
*/*global routing table is followed*/*
retrieve next hop address using global routing table
if (intra domain)
use default shortest path routing
else if (inter domain))
route to sink connection router of current domain
At sink connection router at same domain
find the destination domain
add timestamp field to the capsules
forward to source connection router at the server domain
At source connection router in server domain
receive capsules from sink connection router of source domain
retrieve destination address
forward directly to the sink connection router responsible for destination address
At sink connection router responsible for destination domain
receive capsules
find the destination domain
forward to source connection router at the destination domain
At source connection router in destination domain
receive capsules from sink connection router of the server domain
retrieve destination address
forward directly to the next hop address

7. SIMULATION RESULTS

The distributed simulator is implemented in LAN with homogenous systems and tested for a topology with the maximum number of nodes supported by ANTS, that is separated into various domains and configured in different machines. After establishing the physical and logical connectivity between domains, the design issues like time synchronization and interoperability are implemented. The performance is analyzed and compared to that of simulating in a single system. It is noticed that there is no considerable increase in simulation time and no considerable performance degradation. The packet transmission time as measured in a single system, peer-peer model and client server model of ANTS are charted in figure 3. The analysis of the drop rate of the capsules in the proposed models as well as the single system implementation is shown in figure 4. The graphs are plotted taking the average of the measurements carried out using varying number of peers and clients. It is noticed that the increase in drop rate by a nominal amount is constant for various size topologies when compared to that of ANTS simulation in a single system and hence proved to be scalable. The memory requirements for nodes and their links are shared across the systems. The simulation results show that any active network application with any

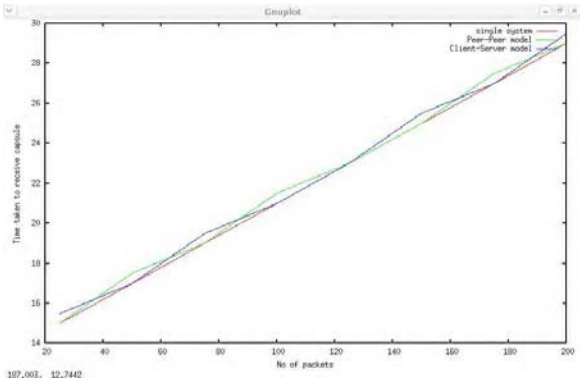

Fig. 3: Transmission time for different models

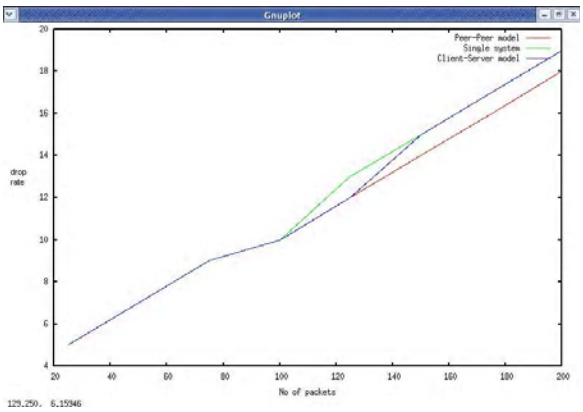

Fig. 4: Packet drop rate for different models

reasonable size topology can be ported using the distributed ANTS simulator in distributed environment and performance analysis can be easily matched without much degradation in the distributed implementation.

8. CONCLUSION

Design and development of the distributed version of the active network simulator ANTS, provides the potential for active applications to define any reasonable size network topology. The toolkit now supports real time testing in ANTS by any active application. Further it employs the graph theoretical properties to effectively partition the given network topology into sub topologies and overcomes the major issues in distributed computing and communication like reduction in simulation capacity and increase in real time delay by schemes like time synchronization and interoperability across domains. As the solution is deployed at application layer, it is generic and portable to any network simulator which has support for enhancement.

REFERENCES

1. Van C.Van, "A defense against address spoofing using Active Networks", Department of electrical engineering and computer science at Massachusetts Institute of Technology.
2. David J.Wetherall, John V.Guttag, David L.Tennenhouse, "ANTS: A toolkit for building and dynamically deploying network protocols", APRIL 1998.
3. David J.Wetherall, John V.Guttag, and David L.Tennenhouse, "ANTS: Network Services Without the Red Tape", *IEEE*, 1999.
4. David Wetherall, "Developing network protocol with the ANTS toolkit", *Design Review*, August 1997
5. David L. Tennenhouse, and David J. Wetherall "Towards an Active Network Architecture" Telemedia, Networks and Systems Group, MIT.
6. David Wetherall, "Active network vision and reality: lessons from a capsule-based system", Department of Computer Science and Engineering University of Washington. *17th ACM Symposium on Operating Systems Principles (SOSP '99) Published as Operating Systems Review 34(5):64–79,* Dec. 1999.
7. Ken Yocum, Ethan Eade, Julius Degesys, and David Becker, "Towards scaling network emulation using topology partitioning", Department of computer science,Duke university.
8. George F. Riley, Mostafa H. Ammar, Richard M. Fujimoto, Alfred Park, Kalyan Perumalla, and Donghua Xu, "A Federated Approach to Distributed Network Simulation",Georgia Institute of Technology.
9. David J.Wetherall, David L.Tennenhouse, Jonathan M.Smith,W.David Sincoskie, and Gary J.Minden. "A Survey of Active Network Research".
10. David J.Wetherall, "Developing network protocols with ANTS toolkit", Design review, MIT, August 1997.
11. Yan Ma, Yanmin Niu, and Min Lei, "Design and Study of Active Router Structure",*Journal of communication and computers*, ISSN1548-7709,USA, September 2005, pp. 75-79.
12. K.L. Eddie Law, Roy Leung, "A Design and Implementation of Active Network Socket Programming".
13. Yannick Carlinet, Virginie Galtier, Kevin L. Mills, Stefan Leigh, and Andrew Rukhin, "Calibrating An Active Network Node".
14. David J.Wetherall, "Service Introduction in an Active Network".