Application Layer Multicasting Overlay Protocol-NARADA Protocol

E. Jagadeeswararao[1], G. Sankara Rao[2] and N.Sai Prathyusha[3]

[1] Jawaharlal Nehru Technological University

Received: 14 December 2013 Accepted: 1 January 2014 Published: 15 January 2014

Abstract

The conventional wisdom has been that Network Layer Internet protocol(IP) is the natural protocol layer for implementing multicast related functionality but it is still plagued with concerns pertaining to scalability, network management, deployment and support for higher layer functionality such as error, flow and congestion control. In this context, an alternative architecture is, Application layer multicast (End Systems Multicasting), where at Application layer, implements all multicast related functionality including membership management and packet replication. This shifting of multicast support from routers to end systems has the potential to address the most problems associated with IP multicast. In Application-layer multicast, applications arrange themselves as a logical overlay network and transfer data within the overlay network (between end hosts). In this context, we study these performance concerns in the context of the NARADA protocol (an application layer multicasting protocol). In Narada, end systems self-organize into an overlay structure using a fully distributed protocol. We present details of NARADA and evaluate it using NS-2 simulations. Our results indicate that the performance penalties are low both from the application and the network perspectives. We believe the potential benefits of transferring multicast functionality from routers to end systems, significantly outweigh the performance penalty incurred.

Index terms— multicast, end system multicast, graph, network, random numbers, routers, links, bandwidth, latency, minimum cost spanning tree, unicast, datagram, i

1 Introduction

ecently, more and more group communication applications (e.g., video-conferencing, onlinegaming, and long-distance education) have emerged with the increasing popularity of the Internet. To support such multi-user applications, multicast is considered as a very efficient mechanism since it uses some delivery structures (e.g., trees or meshes) to forward data from senders to receivers, aiming to reduce duplicate packets, whereas a separate delivery path is built for each sender-receiver pair when simple unicast scheme is adopted.

Initially, multicast is implemented at the IP layer,in which a tree delivery structure is usually employed, with data packets only replicated at branching nodes. In IP multicast, the multicast tree nodes are network routers. However, due to many technical and marketing reasons, such as the lack of a scalable inter-domain multicast routing protocol, the requirement of global deployment ofmulticast-capable IP routers and the lack of appropriate pricing models, etc., IP multicast is still far from being widely deployed.

To resolve the deployment issues of IP multicast, application layer multicast has been proposed as an alternative solution to realize multicast in the Internet.

This paper is organized as follows: Existing System and its Disadvantages, Advantages of the proposed system, Narada features, Narada Design, Our implementation of Narada.

2 II.

3 Existing System

IP multicast (Fig. 1) is a bandwidth-conserving technology that reduces traffic by simultaneously delivering a single stream of information to potentially thousands of corporate recipients and homes. IP Multicast delivers application source traffic to multiple receivers without burdening the source or the receivers while using a minimum of network bandwidth. and it is also introduces a lot of complexity and has scalability constraints. ? The second problem is that IP Multicast tries to conform to the traditional separation of network and transport layers. This worked well in the unicast context but other features like reliability, congestion control, flow control and security are difficult to implement. ? The third and final problem is that it requires changes at the infrastructure level and hence it is not easy to deploy.

III.

4 Proposed System

An alternative to this proposed system is the Application Layer Multicast (Fig. 2) in which all the functionality of multicast is pushed to the end systems or end hosts. Application layer multicasting can implement many complex features of multicast functionality basically constructs an overlay structure among all hosts in the network and then sends messages to the either end hosts in the overlay structure, implementing all other features of multicast is easier at application layer rather tat network layer. ? The overlay structure is built on existing physical links. so we may have multiple overlays over a single physical link hence there will be redundant traffic across the links. ? No more routers need to maintain the per group state information. And the end systems or end hosts take up this responsibility. Since these end systems are part of very few groups it becomes easy to scale the systems. ? Supporting higher layer features such as error, flow, and congestion control can be significantly simplified by leveraging well understood unicast solutions for these problems, and by exploiting application specific intelligence.

IV.

5 Narada Features

Narada is the protocol to implement End System Multicasting. It has many features like:

6 Self organizing

The construction of the end system overlay in fully distributed fashion and is adaptive to dynamic changes in group membership.

7 b) Overlay efficiency

The tree constructed is efficient both from application and network perspective and the number of redundant packets transmission is kept minimal. However the definition of efficiency differs for every application.

8 c) Self Improving

The end systems gather network information in a scalable fashion. So the overlay structure improves as more information becomes available.

9 d) Adaptive to network dynamics

The overlay created adapts to long term variations in internet path characteristics and it is resilient to the inaccuracies in the measurement of these quantities.

V.

10 Narada Protocol Design a) Tree and Mesh Creation

Narada creates a mesh, a highly connected graph between all the nodes (end systems) in the group. It then creates a minimum cost spanning tree among all the end hosts using the mesh. A mesh based approach is used for multi source applications. Also a single shared tree is susceptible to a central point of failure. They are not optimized for a single source. It is important to create a good mesh for creating good trees. A good mesh has the following properties: Firstly, quality of a path between any two members is comparable to the unicast path between the two members. Secondly, each member is connected to a limited number of neighbors in the mesh. Narada runs a variant of standard distance vector routing algorithms and it creates reverse shortest path spanning trees for each source.

11 b) Group Management

Narada keeps the mesh connected, to incorporate new members into the mesh and to repair possible partitions that may be caused by members leaving the group or by member failure. The burden of group maintenance is shared jointly by all members. To achieve a high degree of robutness, our approach is to have every member maintain as list of all other members in the group. Since Narada is targeted towards medium sized groups, maintaining the complete group membership list is not a major overhead. Every member's list needs to be updated when a new member joins or an existing member leaves. The challenge is to disseminate changes in group membership efficiently, especially in the absence of a multicast service provided by the lower layer. We tackle this by exploiting the mesh to propagate such information.

12 c) Member Join

The joining member randomly selects a few group members from the list available to it. And sends the messages requesting to be added as neighbor, it repeats the process until it gets a response from some member, when it has successfully joined the group. Having joined, the member then starts exchanging refresh messages with its neighbors.

13 d) Member Leave and Failure

When a member leaves a group, it notifies its neighbors, and this information is propagated to the rest of the group members along the mesh. We also need to consider the difficult case of abrupt failure. In such a case, failure should be detected locally and propagated to the rest of the group. In this project, we assume a failstop failure model, which means that once a member dies, it remains in that state, and the fact that the member is dead is detectable by other members.

14 e) Mesh Performance

The constructed mesh can be quite sub -optimal, because 1. Initial neighbor selection by a member joining the group is random given limited availability of topology information at bootstrap. 2. Partition repair might aggressively add edges that are essential for the moment but not useful in the long run. 3. Group membership may change due to dynamic join and leave. 4. Underlying network conditions, routing and load may vary.

Narada allows for incremental improvement of mesh quality by adding and dropping of overlay links.

15 VI.

16 Data Delivery

On the top of the mesh, Narada runs the distance vector protocol. Each member maintains a routing cost to the destination and also the path that leads to that node. A member M that receives a packet from source S through a neighbor N forwards the packet only if N is the next hop on the shortest path from M to S. Further, M forwards the packet to all its neighbors who use M as the next hop to reaches (fig. 7).

17 VII.

18 Narada Implemenation & Results

19 Mesh Creation

We use the network entities given by JNS (Java Network Simulator) to create a mesh (Fig. 3). We create entities like nodes, links, routers etc. We'll assign weights to the links manually or can be done using a random number generator. The nodes have names 1, 2 ?etc. the number of edges in the network for a number of nodes is also generated by random numbers. We try to have a highly connected graph. All those nodes which are not connected have a weight of a constant high valued number. In Narada every member of the group contains a list of all members in the group to which it is connected. So a Group Member object has a Node object and an array of nodes and costs to reach them in it. If a member is not connected to a node it has the constant value representing an unreachable node in it. A group is defined as a list of Group Member objects.

20 c) Member Join

When a new node wants to join a group, it brings along with it some information about its distance to any existing group member with it. The group join algorithm works as follows (fig. 4).

In the first step, the list of the joining node is updated. All those elements to which it's not connected are added with unreachable weight to its list. Then it is added to the lists of all existing group members with corresponding weights. Finally it is added to the list of members of a group. When data routing has to be done a new spanning tree will be created with this node. When a member leaves the group gracefully it informs other group members that it is leaving. Accordingly when he leaves his list is deleted and his record is deleted from the its of all other existing group members(fig. 5).When data routing has to be done a new spanning tree will be created without this node. The entire structure of network consisting of all nodes and weighted edges is given to the spanning tree algorithm. We then use the Kruskal's algorithm to construct the minimum cost spanning tree (fig. 6) among these nodes. We also calculate the start and end times for each message of the spanning tree and also the hop number in the tree. The user enters a source and we consider the last node as the destination.

We then extract a path from the spanning tree from the source to the destination. We then give the edges in the path to the simulator which sends the messages along those paths at the specified start times (fig. 7).

21 Results Analysis

We have considered two Parameters to measure the mesh (network) performance. One is the Throughput. And the other is the Latency(Delay). Throughput is nothing but, number of packets sent per unit time successfully. Latency refers to the time taken for a packet to reach the destination after their transmission. We conducted several Experiments to observe the mesh performance. Application Layer Multicasting Overlay Protocol - NARADA Protocol others for medium sized group member's mesh. Fig. 12 shows the delay vs group size, but for small size groups delay is neglible while using narada protocol.

22 Conclusion

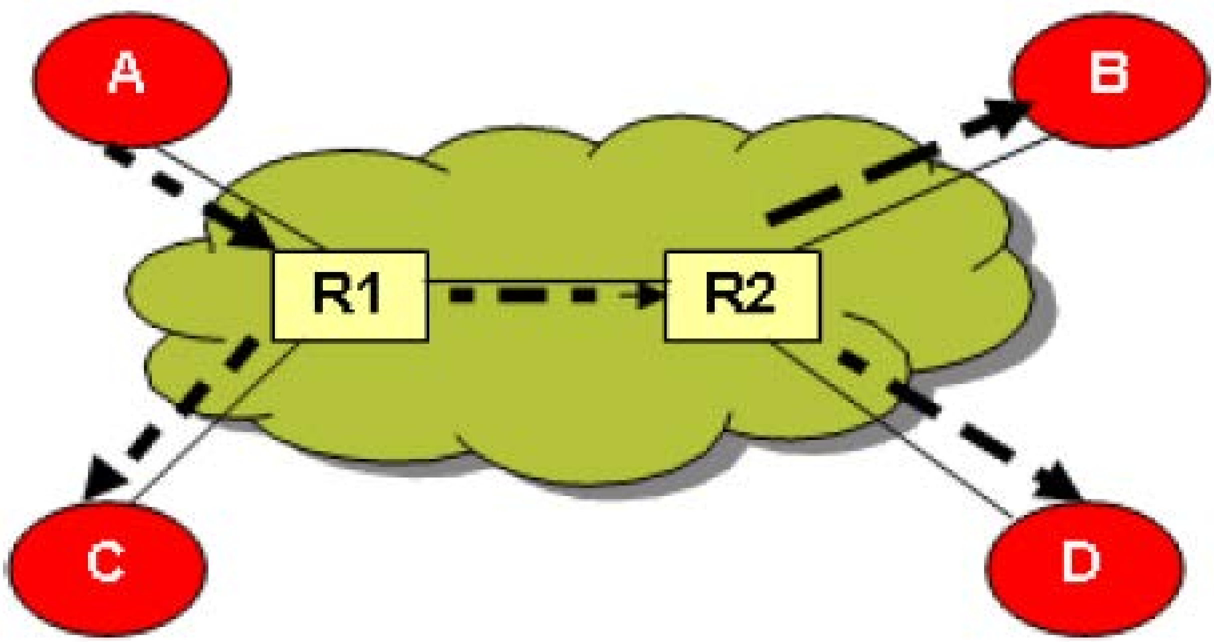
End systems overlay is feasible. End Systems (Application Layer) Multicasting Addresses the problems associated with IP multicasting. Application layer Multicasting is easy to maintain. NARADA is Better for small sized groups from the results we drawn.

23 References Références Referencias

1

Figure 1: Figure 1 :

[1] [2]



2

Figure 2: Figure 2 :

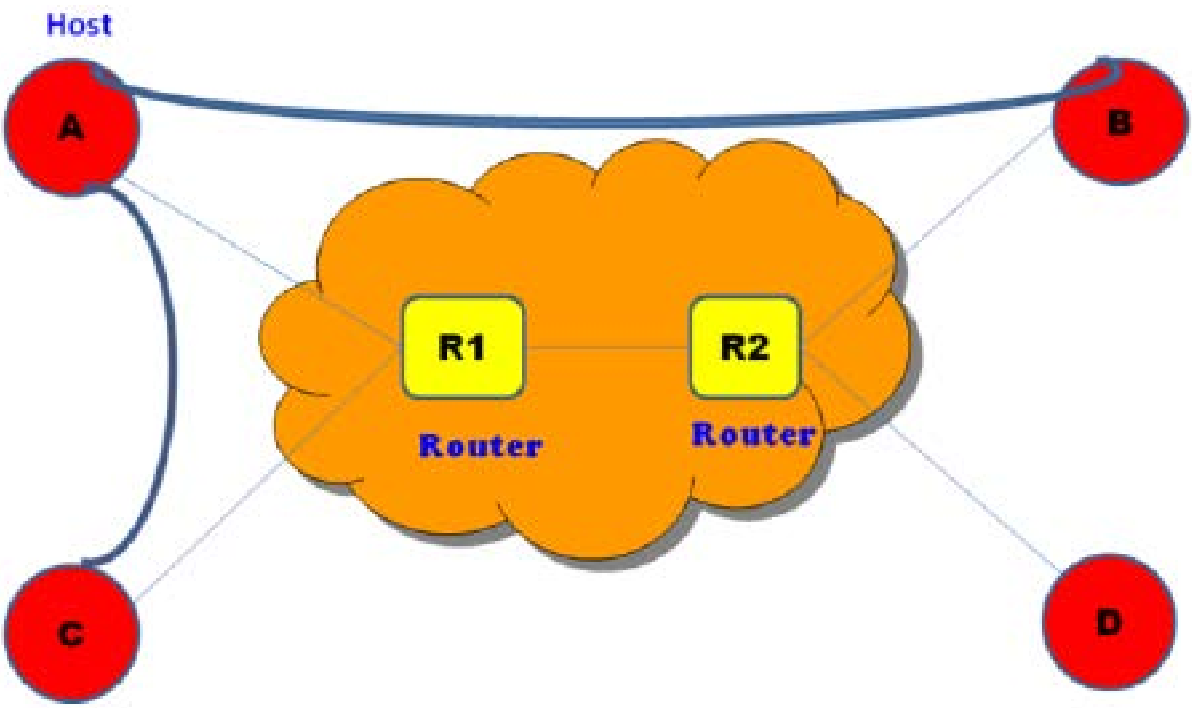


Figure 3: Global

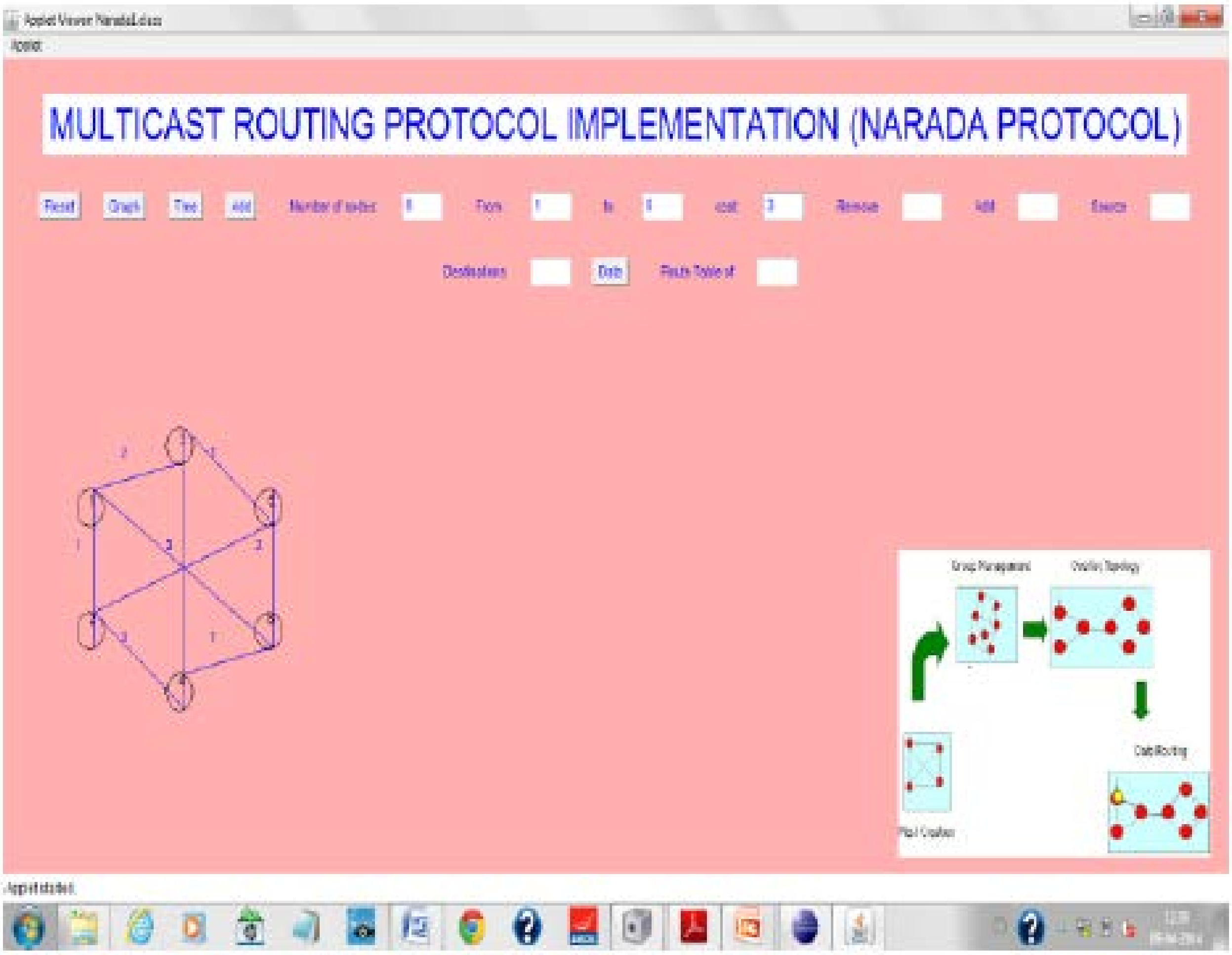
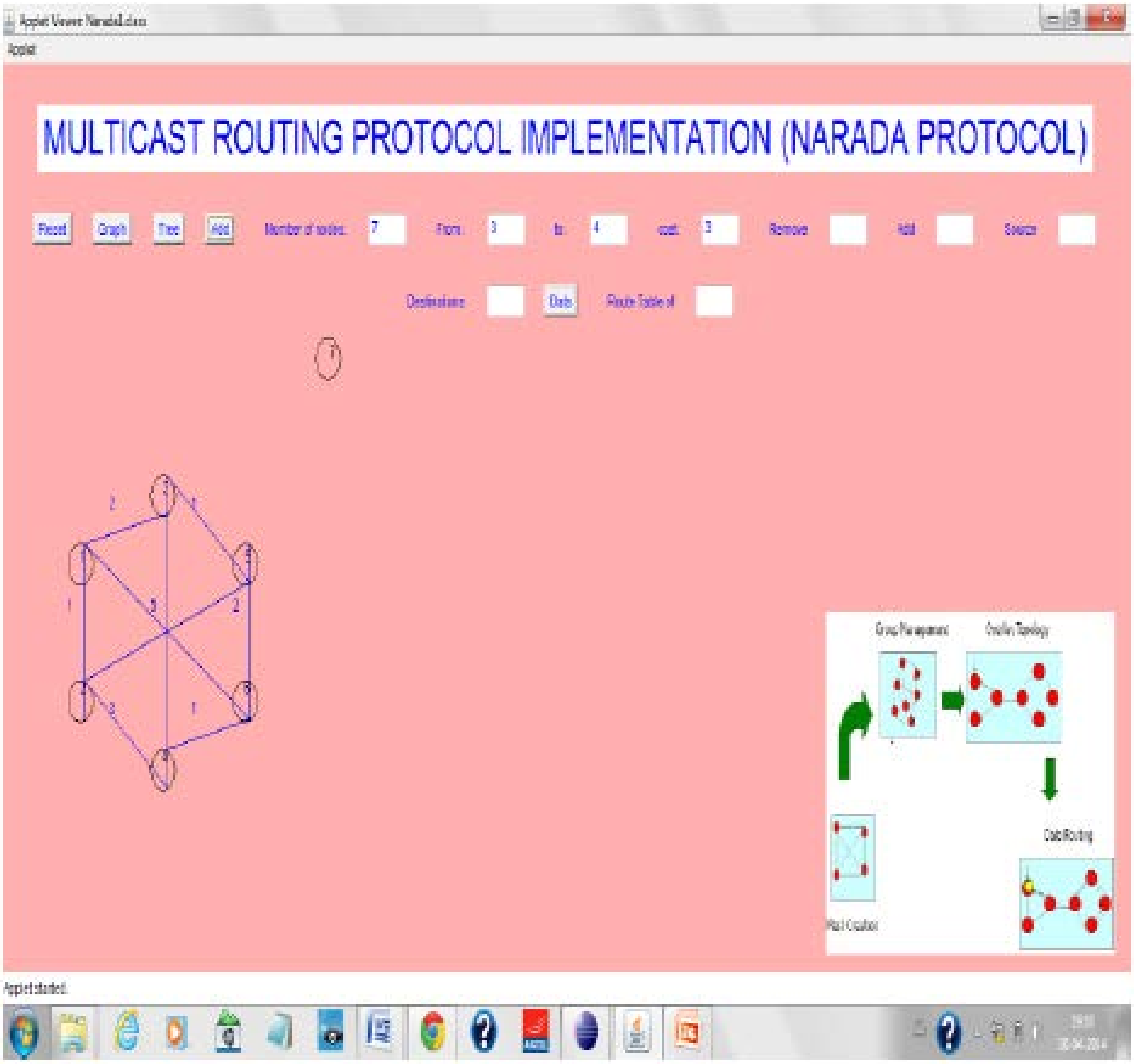
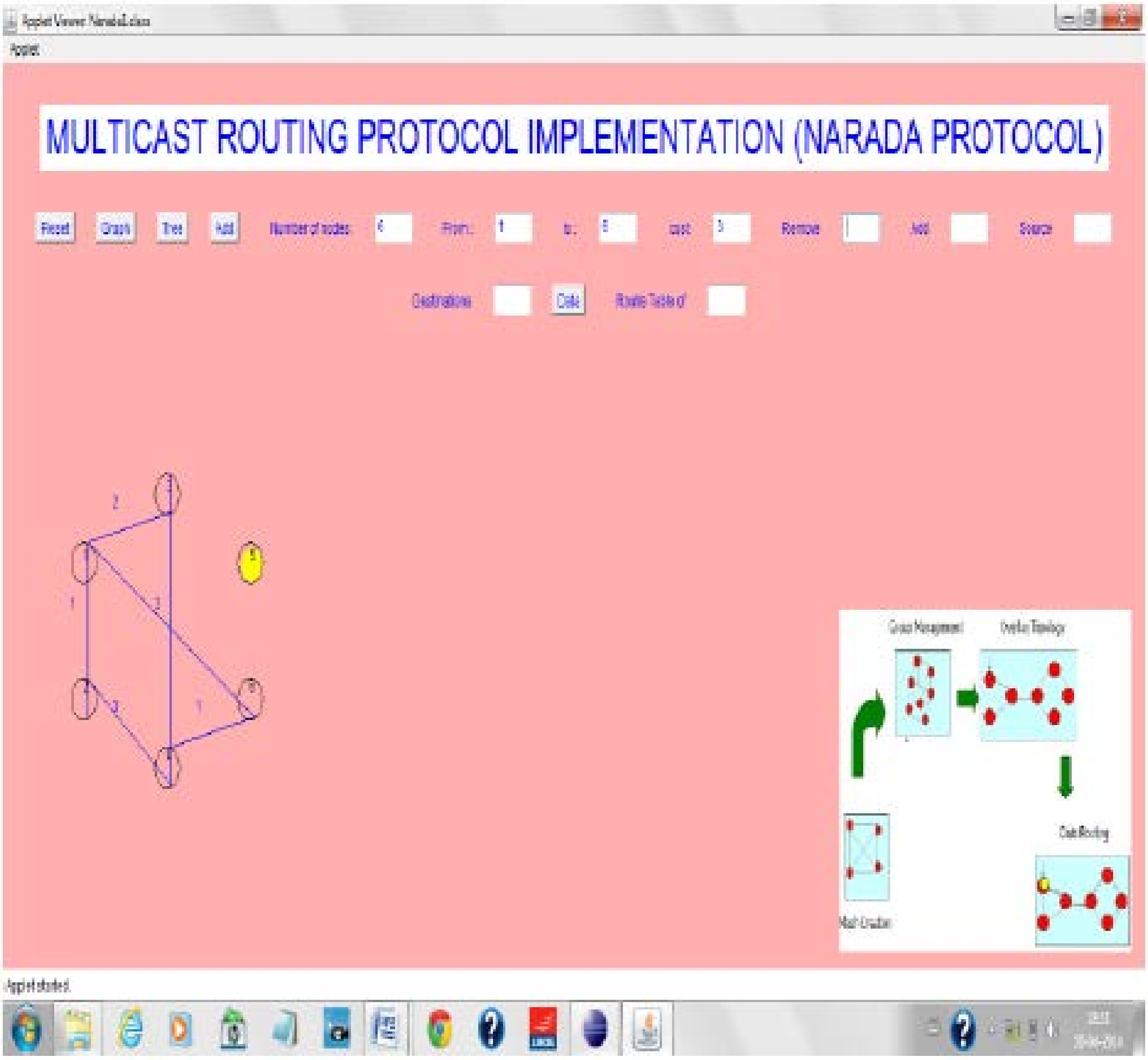
3 

Figure 4: Figure 3 :



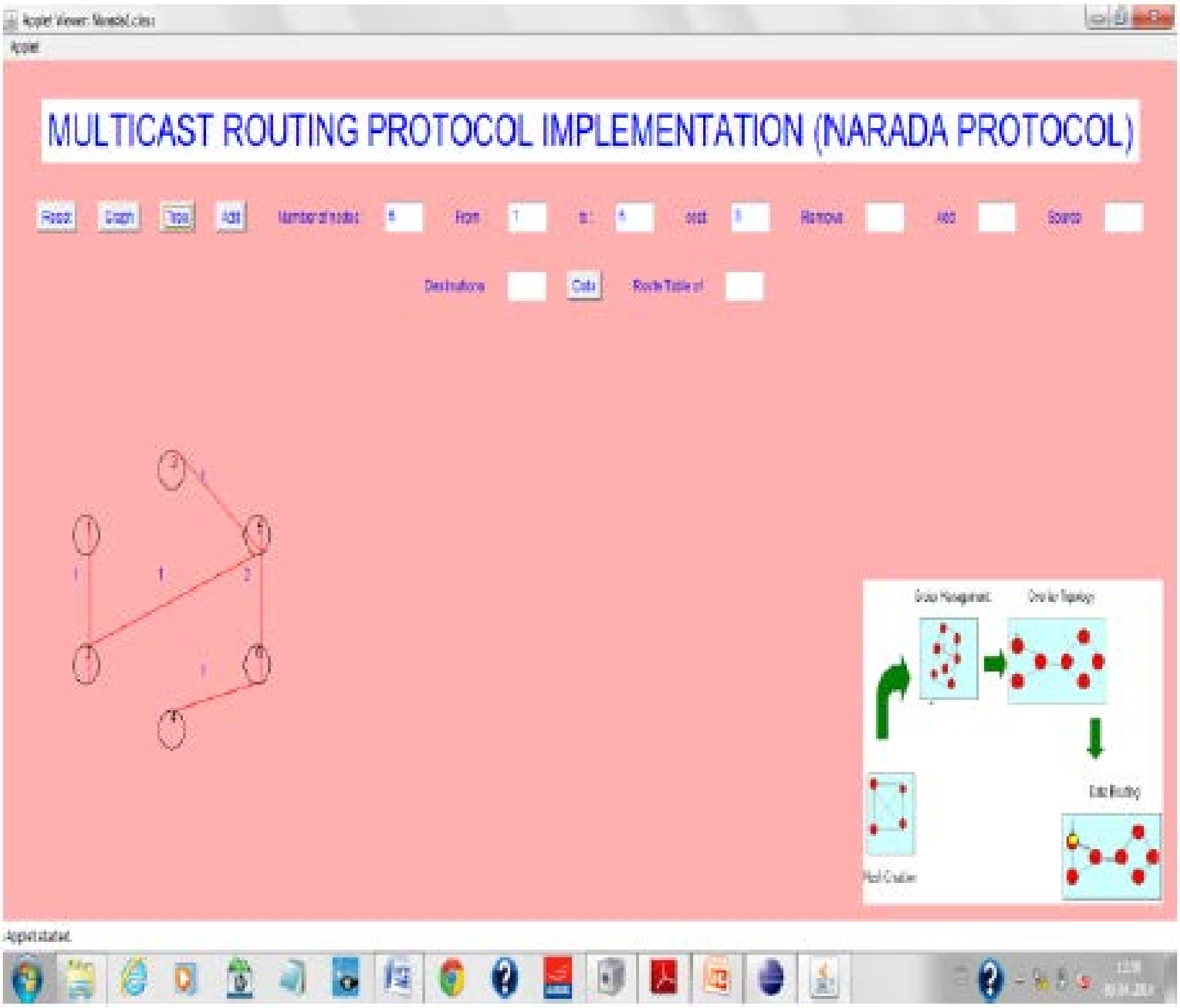
4

Figure 5: Figure 4 :



5

Figure 6: Figure 5 :



6

Figure 7: Figure 6 :

7

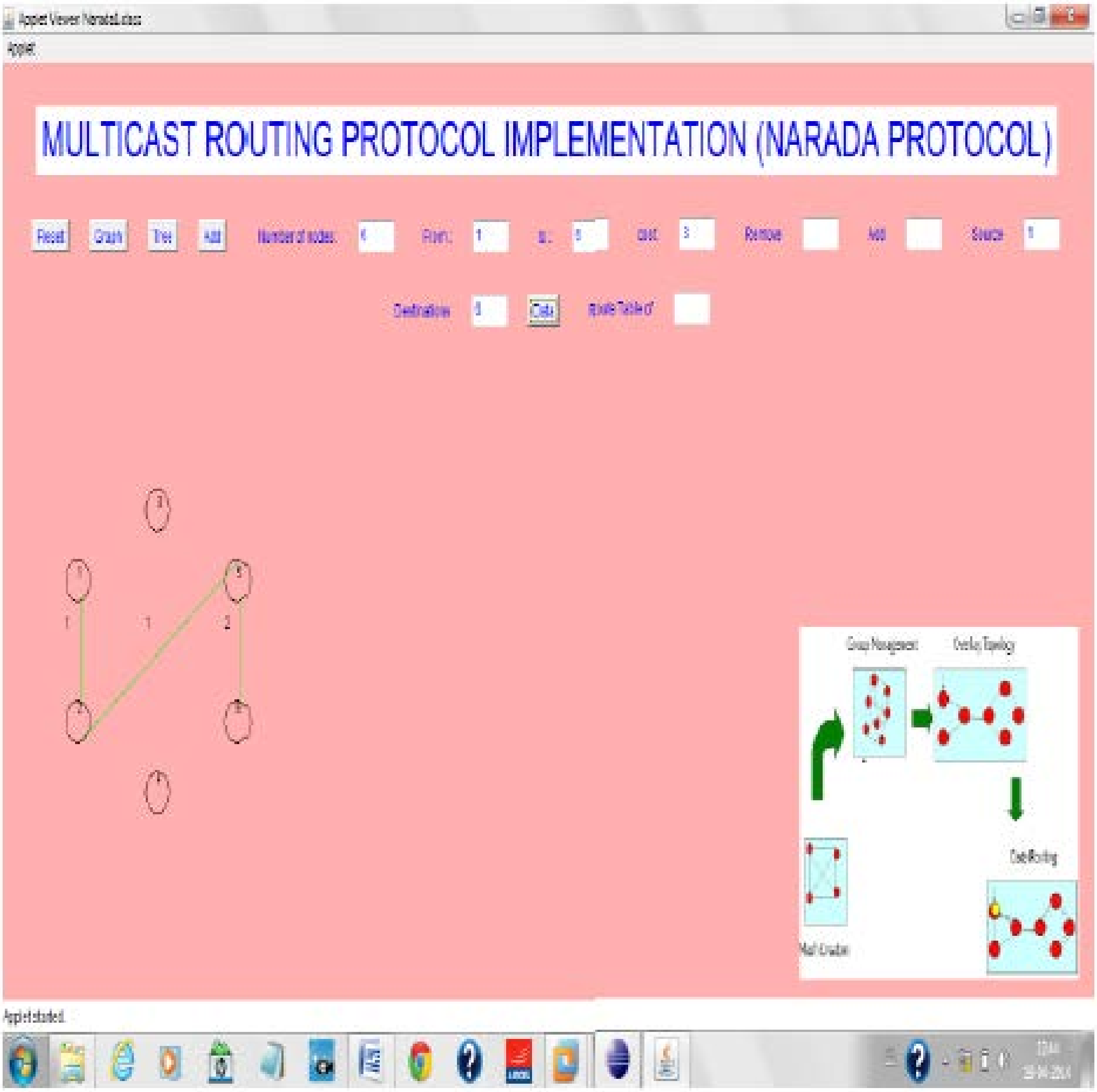


Figure 8: Figure 7 :

8

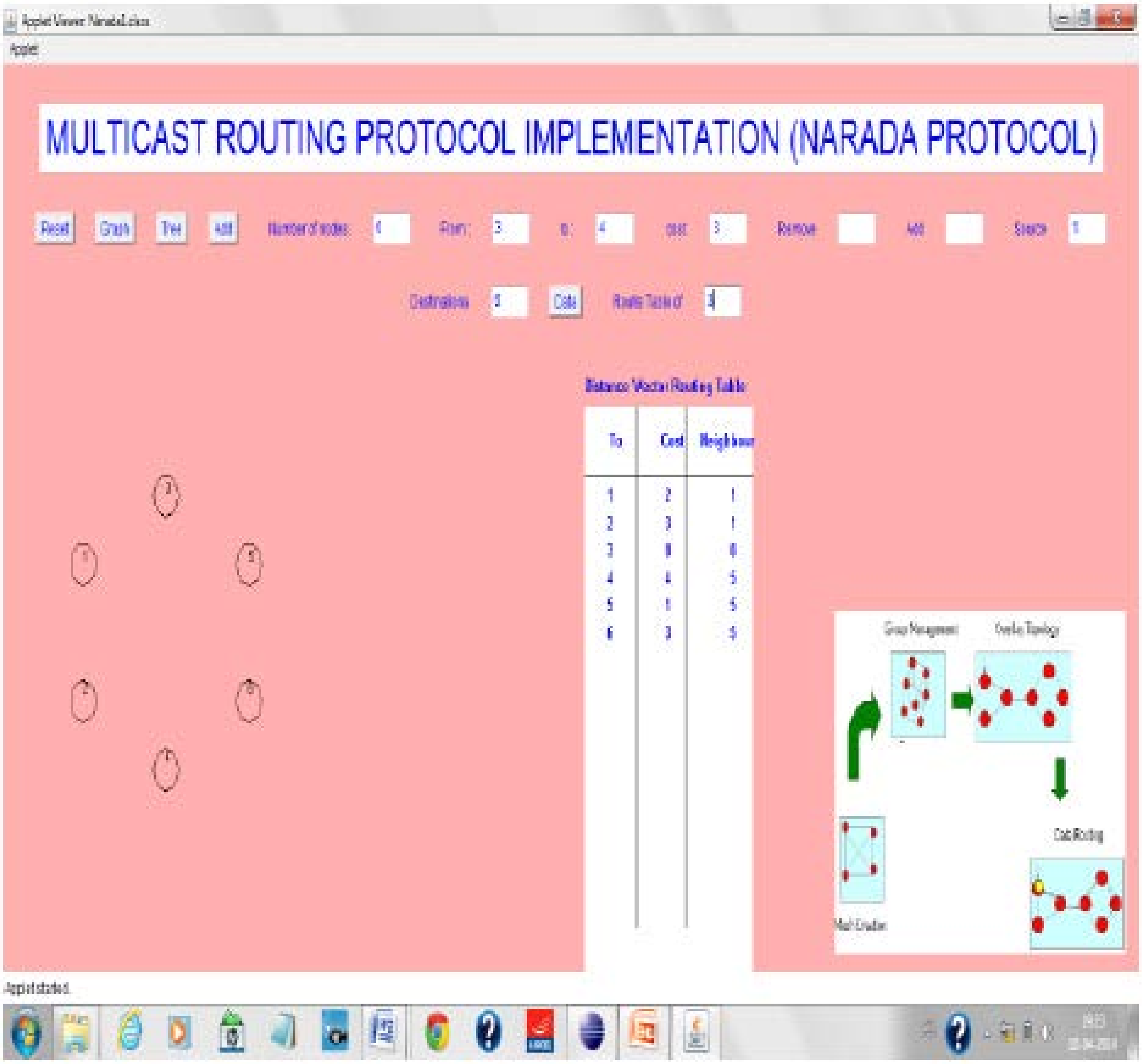


Figure 9: Figure 8 :

10

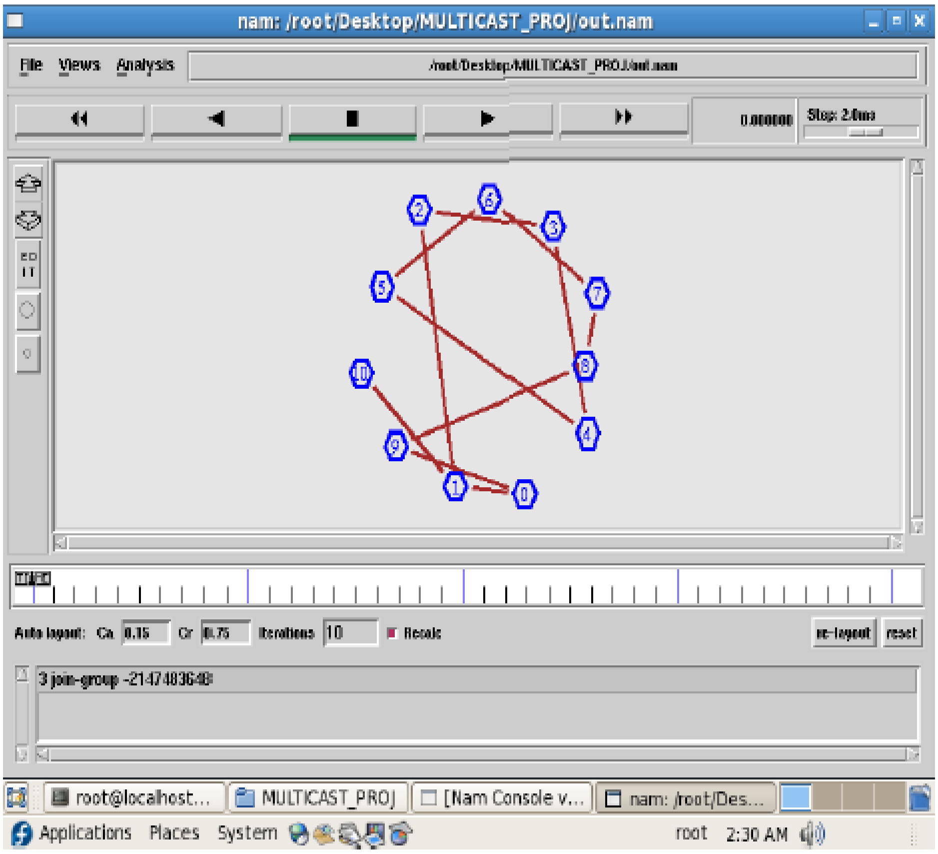


Figure 10: Figure 10 :

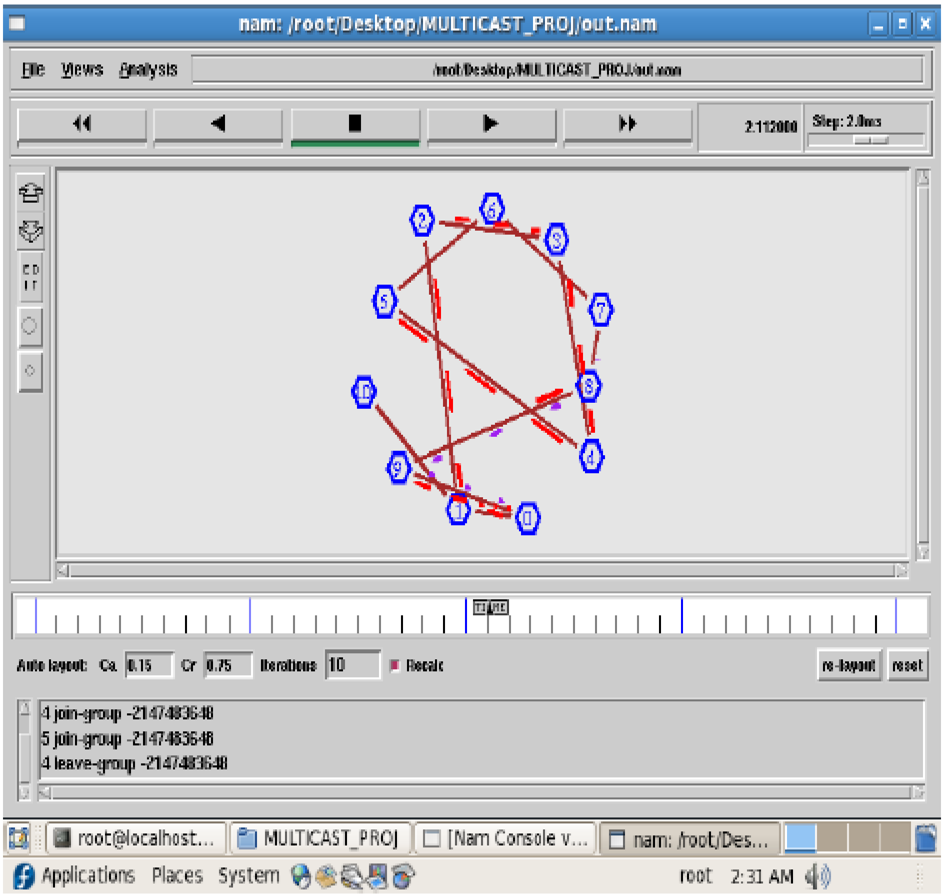

nam: /root/Desktop/MULTICAST_PROJ/out.nam
File Views Analysis
/root/Desktop/MULTICAST_PROJ/out.nam
2.112000
Step: 2.0ms
Auto layout: Ca 0.15 Cr 0.75 Iterations 10 Recalc
re-layout reset
4 join-group -2147483648
5 join-group -2147483648
4 leave-group -2147483648
root@localhost...
MULTICAST_PROJ
[Nam Console v...
nam: /root/Des...
Applications Places System
root 2:31 AM

Figure 11:

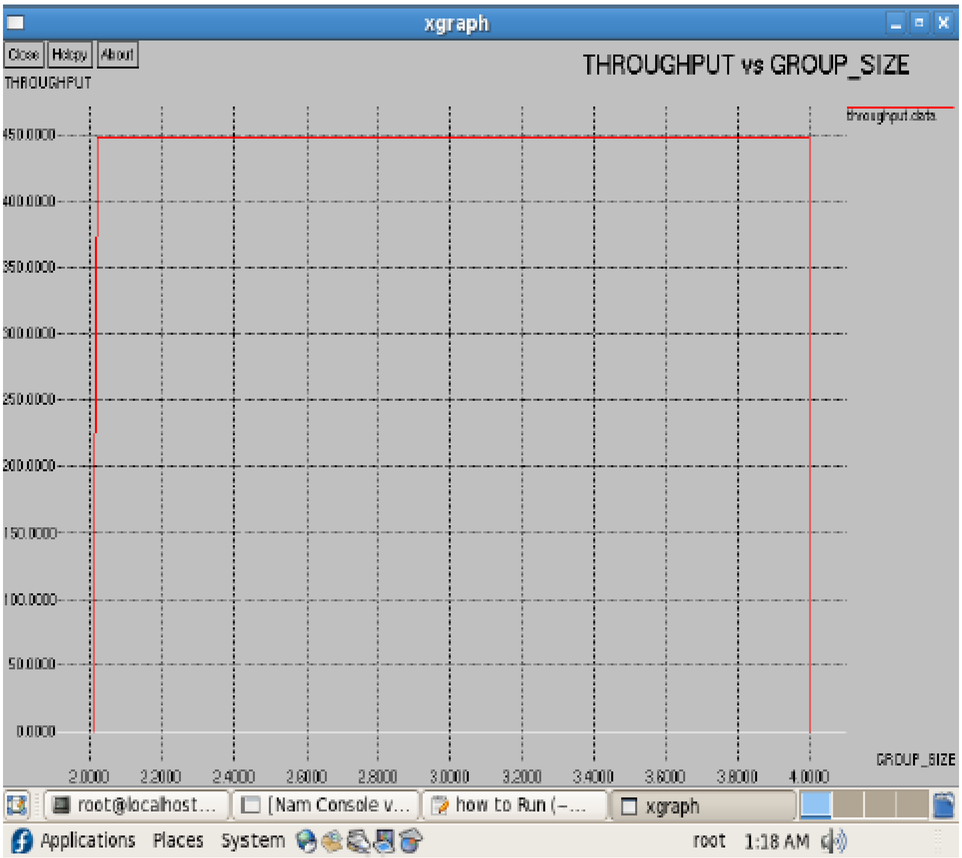
11 

Figure 12: Figure 11 :

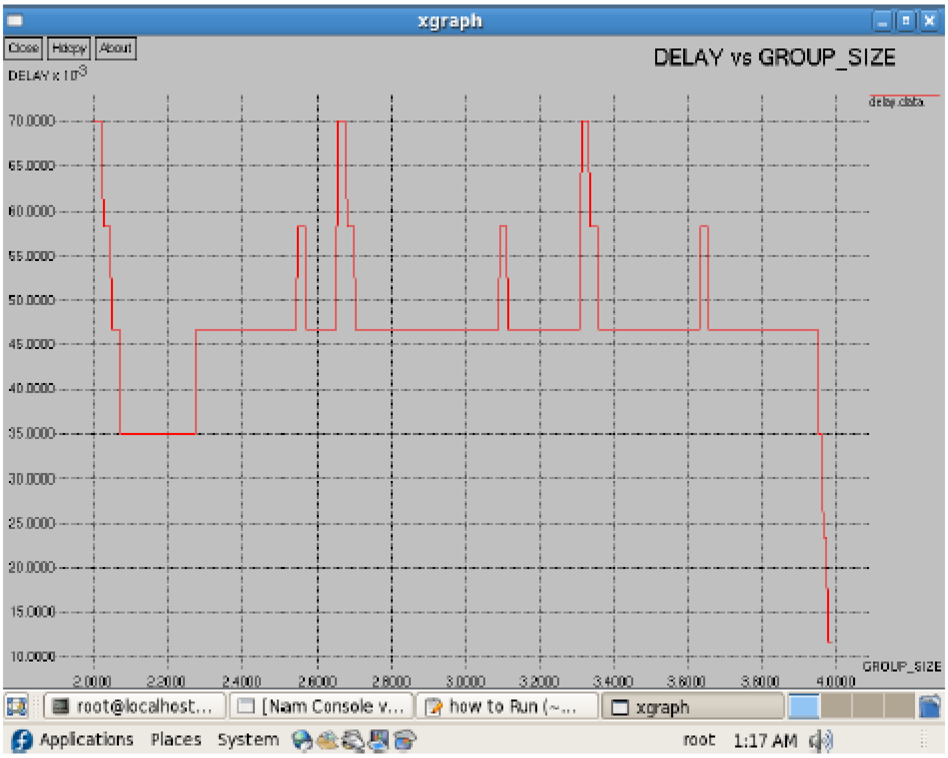
12 

Figure 13: Figure 12 :

[Chu et al. (2000)] ‘A case for end system multicast’. Y.-H Chu , S G Rao , H Zhang . *Proceedings of ACM Sigmetrics*, (ACM Sigmetrics) June 2000.

[Pendarakis et al. (2001)] ‘ALMI: An application level multicast infrastructure’. D Pendarakis , S Shi , D Verma , M Waldvogel . *Proceedings of the 3rd USNIX Symposium on Internet Technologies and Systems*, (the 3rd USNIX Symposium on Internet Technologies and Systems) Mar. 2001.

[Jannotti et al. (2001)] ‘Application-layer Multicast with Delaunay Triangulations’. J Jannotti , D Gifford , K L Johnson , M F Kaashoek , J W O’tooleJr . *IEEE Globecom*, November 2001.

[Liebeherr et al.] ‘Application-layer multicasting with delaunay triangulation overlays’. J Liebeherr , M Nahas , W Si . *IEEE Journal on Selected Areas in Communications*

[Partridge et al. ()] *Distance Vector Multicast Routing Protocol. RFC 1075*, C Partridge , D Waitzman , S Deering . 1988.

[Jannotti et al. (2000)] ‘Reliable Multicasting with an Overlay Network’. J Jannotti , D Gifford , K L Johnson , M F Kaashoek , J W O’tooleJr , Overcast . *Proceedings of the Fourth Symposium on Operating System Design and Implementation (OSDI)*, (the Fourth Symposium on Operating System Design and Implementation (OSDI)) October 2000.

[Shi and Turner (2002)] ‘Routing in overlay multicast networks’. S Shi , J S Turner . *Proceedings of IEEE INFOCOM*, (IEEE INFOCOM) June 2002.

[Banerjee et al. (2002)] ‘Scalable application layer multicast’. S Banerjee , C Kommareddy , B Bhattacharjee . *Proceedings of ACM SIGCOMM*, (ACM SIGCOMM) Aug. 2002.

[Chawathe ()] *Scattercast: An Architecture for Internet Broadcast Distribution as an Infrastructure Service*, Y Chawathe . 2000. (Ph.D. thesis)

[Francis] *Yoid: Extending the Multicast Internet Architecture*, P Francis . http://www.aciri.org/yoid/