Overlay multicast network optimization and simulation Based on Narada Protocol

LI Xing-feng [1,2] YAN Bao-ping[1] LUO Wan-ming [1]
[1] Computer Network Information Center, Chinese Academy of Sciences, Beijing 100080, China
[2] Graduate University of Chinese Academy of Sciences, Beijing 100039, China
lixf@cnic.cn

Abstract-In this paper, overlay multicast is studied in depth. Taking into account the advantage of the mesh structure, we propose an overlay multicast protocol suitable for the transmission of real-time multimedia, which can transcend the tree structure. This protocol optimizes the structure of multicast tree in order to make full use of each node to enhance the transmission ability of data, instead of exploiting the inner nodes of the tree, which can improve the usability of network resource and enhance the robusticity and effectivity. Furthermore, we perform a series of simulation experiments. Experimental evaluation shows that our protocol can effectively solve the problem of single point of failure and have better fault tolerance and higher scalability.

Key Words: Overlay, Multicast, Spanning tree, Mesh

1. INTRODUCTION

Overlay multicast is an important group communication model for one-to-many and multi-party communication. It is a key technique for the next generation Internet applications. In the recent years，the research trend on peer-to-peer throws new light on multicast，and overlay multicast begins to attract wide attention. Using unicast, overlay multicast is implemented to multicast data on application layer, which replicates and distributes data via not routers but terminal computers.

In order to implement overlay multicast, the critical task is to construct the structure of multicast, i.e. data flow path. The structure implies the relationship of nodes. On the basis of the relationship, the current overlay multicast protocols are classified into two types: tree-based overlay multicast and mesh-based overlay multicast. Each type of overlay multicast has respective pros and cons.

As far as multicast is concerned, the overlay tree is probably the most natural structure for multicast, which is the most efficient in terms of bandwidth and delay optimization. Furthermore, it is desirable that in tree-based overlay structure there are no routing loops formed during tree construction, which greatly simplifies the routing algorithm. However, tree-based overlay multicast is sensitive to the partition of the overlay structure because they are acyclic graphs. If any non-leaf member of the overlay tree leaves the overlay, voluntarily, or by failure, the tree is broken and there will be no path for members of the multicast group to communicate.

Mesh-based overlay multicast provides multiple or redundant paths between members. Therefore, its overlay structure has little chance of being partitioned due to node failure or departure. Alternate paths will be ready without the need to re-construct a path as is the case in tree-based overlay multicast, which is desirable when considering routing stability and offering quality of service in group communication. On the other hand, as far as mesh-based overlay multicast is concerned, it is necessary to run a routing algorithm for construction of loop-free forwarding paths between group members. Another main drawback is that the additional paths can make an excessive consumption of network resources when sending data packets, which leads to inefficiency because more than one copy of a message may use a link in the forward direction. This is not the case in a tree-base overlay multicast, nor is it necessary to run a routing algorithm once the tree is established in order to prevent loops.

Given the pros and cons of the two types, the overlay tree is probably the most natural structure for a multicast. In this paper, our aim is to improve the disadvantage of tree-base overlay and combine the advantages of tree-based overlay and mesh-based overlay to construct and maintain an efficient and robust pseudo-tree overlay, which does not only exploit the non-leaf nodes but also makes good use of leaf nodes to enhance the efficiency of data transmission. In this paper, a new overlay multicast protocol based on Narada protocol has been proposed.

2. RELATED WORKS

The Narada protocol is developed by CMU (Carnegie Mellon University) researchers and the protocol is one of the first application layer multicast protocols that can demonstrate the feasibility of implementing multicast functionality at the application-layer. Every node maintains global group membership information and periodically improves mesh quality by probing and adding or dropping links. This feature allows it to construct multicast trees with high quality, but the overhead explodes when group size increases.

3. DATA STRUCTURE

In order to illustrate the protocol, we define an ordered pseudo-tree structure, named as T. As far as each node is concerned, there are a maximum outdegree of 6, a father, a mother, and a grandfather. In addition, each node is designated as a global ID. The naming rule of the ID is summarized as follows:

(1) The root node is represented as A.

(2) The first child node of a node is represented as A, the second child node is represented as B,. . .,and the sixth child node is represented as F.

(3) Therefore, if the level of a node is n, the node is represented as a string of n letters which are composed of A, B, C, D, E, F.

The pseudo-tree nodes can be represented as follows:

```
#define NUM_CHILDREN 6
struct PseudTree{
   PseudTreeNode  *  m_Root;
}
Struct PseudTreeNode{
   CString m_ID;
   PseudTreeNode * m_Father;
   PseudTreeNode * m_Mother;
   PseudTreeNode * m_Grandfather;
    PseudTreeNode * m_children[NUM_CHILDREN];
}
```

4. NODE ORGANIZATION

1) Here, we define a special designated host, called the Rendezvous Point (RP), which is used to boot-strap the joining procedure of a new member. A new member can know the root address by querying RP node. At first, it obtains the children list of the root from the root, and from the list it can select the nearest one as its potential father. This step is repeated until its potential father accepts it. In view of the policy, bandwidth, traffic load, and so on, the potential father decides whether to accept a joining request. If the potential father rejects it, it goes back up one level and resumes the search for fathers. This process stops when it reaches a leaf node, or a node that is closer than all its neighbors.

2) Each node records the information of its grandfather, which can be ready for recovering from network partition.

3) Each pseudo-tree node has a mother node which is the standby node of the father node and is used to receive data and transfer data when there is abnormity. The mother node is selected from the leaf-nodes which are not its offspring. When a node selects its mother node, routing loop must be avoided with the aid of the naming rule. As is illustrated in Fig 1, there is a tree of 12 nodes. The ID of each node is defined based on its naming rule. For example, as far as node 6 is concerned, its level is 3 and it is the second child of node 2 and node 2 is the first child node of node 1.Therefore, the ID of node 6 is AAB. With the aid of the comparing function of String, it is not difficult to decide that the ID2 is the prefix of ID10, and ID2 is not the prefix of ID12 .For this reason, node 10 is the offspring of node 2 but node 12 is not the offspring of node 2, that is to say, node 12 can be the potential mother node but node 10 can not be the potential mother node.

4) In order to make full use of the computing ability and bandwidth resources, each leaf is provided with a series of buffers used to store the received data which can be transferred to other nodes when this node is selected as mother node.

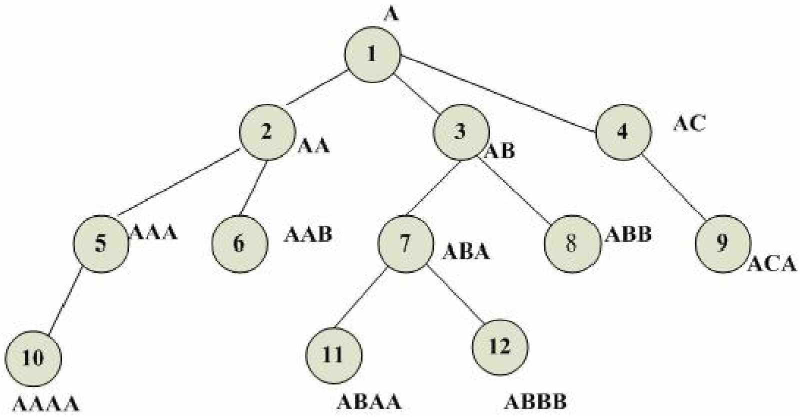

Figure 1. The Pseudo-tree Overlay Structure

5. Key Algorithms

Our protocol, named as ENarada, constructs and maintains a self-organizing fully distributed overlay structure by measuring the network path characteristics. The overlay-spanning tree is constructed in a two step process. In the first step a rich connected graph called the mesh is constructed among the terminals based on Gossip protocol. In the second step, using DVMRP-like algorithm, a reverse shortest path spanning tree is constructed from the mesh. This DVMRP-like routing algorithm is iterative, asynchronous and parallel, and the multicast tree is generated based on the cooperative work of each node. The parallel algorithm is depicted as follows:

```
/* Each node x maintains the routing data as follows :
   (1) The Distance Vector of node x: $D_x(y)=[D_x(y) :y \text{ in } N]$
   (2) The cost from x to direct neighbor y:   $c(x,y)$
   (3) The Distance Vector of each neighbor v of node x: $D_v(y)=[D_v(y) :y \text{ in } N]$
*/
Initialization:
    for each destination $y$ in N
    {
       if $y$ is not a neighbor then
           $D_x(y)= \infty$ ;
       else
           $D_x(y)=c(x,y)$;
    }

     for each neighbor $w$
        for each destination $y$ in N
        $D_w(y)=\infty$;
    for each neighbor $w$
        send distance vector $D_x=[D_x(y):y$ in $N]$ to $w$

    loop
    wait(until I see a link cost change to some neighbor $w$ or until I receive a distance vector form some neighbor $w$)
        for each $y$ in N:
            $D_x(y)=min_v\{c(x,y)+D_v(y)$
        if $D_x(y)$ changed for any destination $y$
            send distance vector $D_x$=[$D_x$(y):y in N] to all neighbours
    forever
```

After the overlay tree is generated, each non-root node $Node_i$ can send messages to request the newest data. The leaf node, which has received the request message, checks to determine whether there is the requested data in the buffer. The leaf node will respond with the requested data if the requested data is in its buffers. $Node_i$ will select the node, whose response message has been first received, as its mother node and neglect other responses. The parallel algorithm is depicted as follows:

```
Initialization:
N = { };
int count=0;
for all nodes v{
    if(!strncmp (IDm,IDi,strlen(IDm))&&(m!=i))
    then{
        add v to N;
        counter++;
    }
  if(count>=n) break;
}
for all nodes v in N{
    send the newest data request to v;
}

While(1){
    If(receive a response to the request)
    {
        record the receiving time;
        select the node as the mother node;
        break;
    }
}
```

If the father can not work normally, the mother can be exploited to receive and transfer data, and at the same time, the recovering process of failure is launched. Because the mother is the standby node of father node, there is no data loss and variable delay during the recovering process. The abnormal node can select its offspring as its father node, which can minimize the ill effect on performance of tree.

5. SIMULATION

In order to verify the performance of our new protocol we conduct a series of simulations. The network topologie in our simulation are generated using the Transit-Stub grapl model generated by GT-ITM topology generator. Al topologies have 2000 routers with an average node degre between 3 and 6. End-hosts are attached to a set of router which are chosen uniformly at random from th stub-domain nodes. In our simulations, there is no data los resulted from congestion, and no background traffic or jitte However, data is lost whenever the application-laye multicast protocol can not have a path from the source to receiver, and duplicates are received whenever there i more than one path.

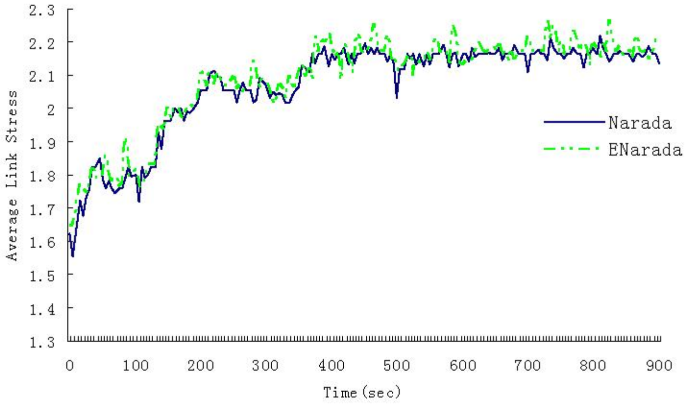

Figure 2. Average link stress

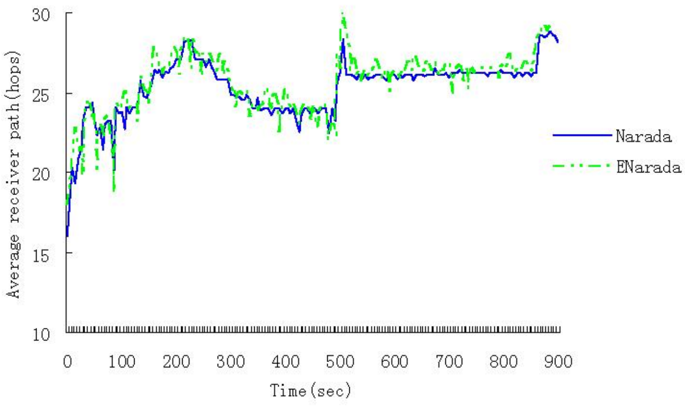

Figure 3. Average path length

Figure 2 and Figure 3 respectively show the change of average link stress and average path length during the change of multicast overlay structure. From the two figures, as far as both average link stress and average path length are concerned, there is no distinct difference between Narada and ENarada. In addition, the change of curve trend is not significant.

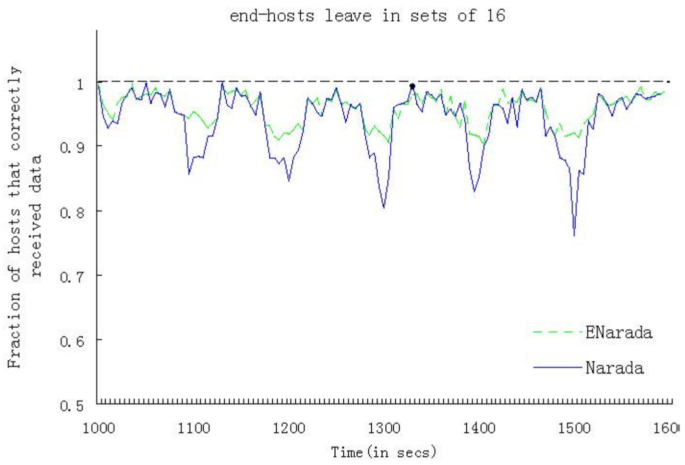

Figure 4. Fractionof members that received data packets

Figure 4 depicts the fraction of members that have received data packets. In this simulation experiment, the data source periodically sends the equal-sized (50 bytes) packets at the rate of 16 packets per second to all multicast members. Receivers calculate the loss rate of multicast packets. Form Figure 4 , the loss rate of Enarada decreases more apparently than that of Narada, and accordingly the fault tolerance is improved substantially.

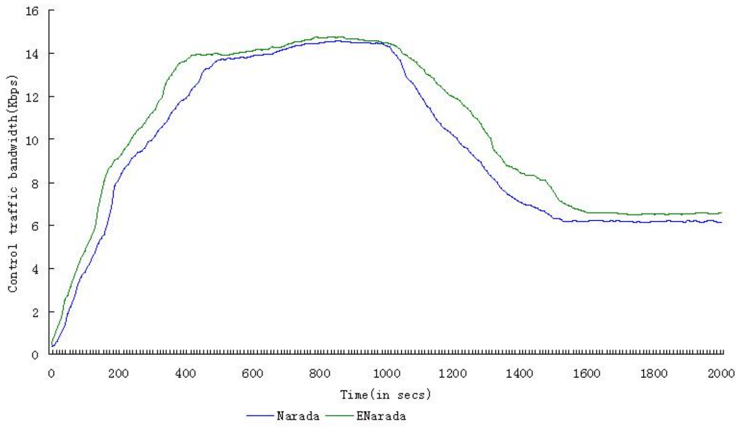

Figure 5. Control bandwidth required at end-host

To some degree, the optimization of Narada protocol may bring about protocol overhead. The overhead can be divided into two classes. One is the bandwidth consumed by the periodic exchange of control messages among multicast members. The other is the delay and bandwidth brought about by periodically detecting its neighbors. Here, we evaluate the feasibility of the new protocol by comparing the protocol overhead of Enarada and Narada. In Figure 5, the control bandwidth required at end-host access links is depicted. The x-axis represents time (in second), and the y-axis represents the average bandwidth consumed by the control messages received and sent by multicast members during 10 second intervals. According to the above figure, during the departure and join of multicast members the control overhead is lager than that of stable state. The reason is that when the multicast pseudo-tree is unstable, and especially there are many members which depart and join the overlay structure, Enarada can take corresponding mechanisms proposed to optimize the overlay structure which increases the control overhead. However, as for Narada, the average of bandwidth overhead is 9.506Kbps during [0，2000] seconds , and as for Enarada, the average bandwidth overhead is 10.322Kbps, so the rate of control overhead increase is 0.086. That is to say, the control overhead does not increase significantly than Narada. The simulation results show that our protocol can effectively solve the problem of single point of failure and has better fault tolerance and higher scalability.

6. CONCLUSIONS

In this Paper, an overlay multicast protocol is proposed to improve the transmission of real-time multimedia. The main step is to optimize the structure of multicast tree and make full use of each node to improve the transmission ability of data. Simulations show that it is desirable for our protocol to solve the problem of single point of failure. In addition, the protocol has better fault tolerance and higher scalability.

ACKNOWLEGEMENTS

This work was supported by the National Basic Research Program of China under No. 2003CB314807.

REFERENCES

[1] Chu, Uang-hus, Sanjay G. Rao, and Hui Zhang, “ A Case for End System Multicast,” ACM SIGMETRICS 2000, pp. 1-12.

[2] S. Banerjee, B. Bhattacharjee, and C. Kommareddy. Scalable application layer multicast. In Proceedings of ACM Sigcomm, August 2002.

[3] Y.-H. Chu, S. G. Rao, S. Seshan, and H. Zhang. Enabling Conferencing Applications on the Internet using an Overlay Multicast Architecture. In Proceedings of ACM SIGCOMM, August 2001.

[4] D. Pendarakis, S. Shi, D. Verma, and M. Waldvogel. ALMI: An Application Level Multicast Infrastructure. In Proceedingsof 3rd Usenix Symposium on Internet Technologies & Systems, March 2001.

[5] C. Jin, Q. Chen, and S. Jamin. Inet: Internet Topology Generator. Technical Report CSE-TR-433-00, EECS Department, University of Michigan, 2000.

[6] Paul Francis. Yoid: extending the internet multicast architecture.Technical report, NTT, April 2000.

[7] A. El-sayed, V. Roca, L. Mathy, “A Survey of Proposals for an Alternative Group Communication Service”, IEEE Network, pp.46-51, January/February 2003.

[8] M. Kwon, S. Fahmy, “Topology-Aware Overlay Networks for Group Communication”, ACM NOSSDAV’02, 2002.

[9] J. Jannotti, D. K. Gifford, and K. L Johnson, “Overcast: Reliable Multicasting With An Overlay Network. In USENIX Symposium on Operating System Design and Implementation, San Diego, CA, October 2000.

[10] B. Zhang, S. Jamin, and L. Zhang. Host multicast: A framework for delivering multicast to end users. In Proceedings of IEEE Infocom, June 2002.

[11] D. A. Helder, S. Jamin, End-host Multicast Communication using Switching-Trees Protocols, GP2PC’02

[12] K.W. Lee, S. Ha, J.R. Li, V. Bharghavan, An Application-level Multicast Architecture for Multimedia Communications, ACM Multimedia, 398-400, 2000.