International Journal of Computer and Internet Security.
ISSN 0974-2247 Volume 3, Number 1 (2011), pp. 1-8
© International Research Publication House
http://www.irphouse.com

Implementation of Searchtree in the Multicast Networks

Satisha[1] and Paramesh[2]

[1]Research Scholar in JJT University, India
[2]Professor in AIT, India
E-mail: satishsatyalal@gmail.com, drparameshaitrd@rediffmail.com

Abstract

The successful deployment of multicast in the Internet requires the availability of good network management solutions. Discovering multicast tree topologies is an important component of this task. Network managers can use topology information to monitor and debug potential multicast forwarding problems. In addition, the collected topology has several other uses, for example, in reliable multicast transport protocols, in multicast congestion control protocols, and in discovering network characteristics. We present a mechanism for discovering multicast tree topologies using the forwarding state in the network. We call our approach Search tree. First, we present the basic operation of Search tree. Then, we explore various issues related to its functionality. Next, we provide a detailed evaluation by comparing it to the currently available alternatives. Finally, we discuss a number of deployment issues. We believe that tracetree provides an efficient and scalable mechanism for discovering multicast tree topologies and therefore fills an important void in the area of multicast network management.

Keywords: Forwarding state, management, monitoring, multicast, routing, tree topology discovery.

Introduction

With the deployment of native multicast in commercial networks, multicast is getting closer to becoming a ubiquitous service in the Internet. Before multicast can be used as a revenue-generating service, its robust and flawless operation needs to be established in the interdomain [1]. This requires the availability of management tools to help network administrators configure and maintain multicast functionality within and between multicast-enabled domains.

Discovering multicast tree topologies is an important component of multicast network management [2]. Network managers can use the topology information as the basis of group monitoring or can use it to identify potential multicast forwarding problems that may occur due to routing protocol limitations, multicast network misconfigurations, or routing policy decisions. In addition, topology information has several other uses reliable multicast transport protocols [3], multicast congestioncontrol protocols [4], and discovering network characteristics [5]. Finally, end users can use topology information and traffic flow to monitor activity in a group, or, if there is a problem, where to direct an inquiry [6].

The organization of this paper is as follows. Section II gives a searchtree Scalability. Section III describes the Deployment Issue; Section IV describes the Software Analysis and design. In Section V, we present our conclusions.

Searchtree Scalability

Searchtree depends on each and every compliant on-tree router to send its response back to the querier. Basic scalability is provided by dividing topology discovery into rounds and discovering a controlled portion of the tree in each round. In addition to this mechanism, based on the characteristics of multicast forwarding trees, we propose a new response collection approach to further improve the scalability of searchtree. We call this approach non-relay response collection (nr-response).

The nr-response operates as follows: on receiving a request packet, each relay router first creates its response packet. Then, instead of sending this response directly to the querier, it appends it to the end of the request packet and forwards it to its downstream neighbor. On receiving a request packet, each branching router first creates its own response packet, and then appends it to the end of the accumulated information. At this point, the collected response information corresponds to the multicast path between this router and the previous compliant branching router on the multicast tree. In the next step, this router separates the accumulated response information from the request packet and sends it back to the querier. In the last step, it forwards a fresh request packet (a request packet having no response information appended) to its downstream neighbors. In addition, if a router has only one out-going interface but this interface is on a shared LAN segment and if this router has more than one multicast enabled

Neighbor on this shared LAN segment, then the router considers itself a branching router. In the case of leaf routers, they will perform similar steps as the branching routers (except for the request forwarding step).

One final modification related to nr-response is on the scope calculation of the request packets. As we have mentioned previously, searchtree uses a modified-TTL scoping mechanism for scalability and uses the duplication of IP TTL values (in the TTLtt field) to detect non-compliant routers. In the original tracetree mechanism, at each compliant router on the tree (whether it is a relay router or not), these values are computed or decremented. In nr-response, we require only the branching routers and the leaf routers to send responses back to the querier. Therefore, using the IP TTL value alone is not very helpful for controlling the number of responses. For this reason, we propose a slightly different TTL scoping mechanism for controlling the scope of request packets. That is, we use a new field TTLnr in the tracetree protocol header to indicate the number of responses expected to be received in this round from the network. In this situation, the (TTLIP; TTLtt) pair is used to detect noncompliant routers on the multicast tree.

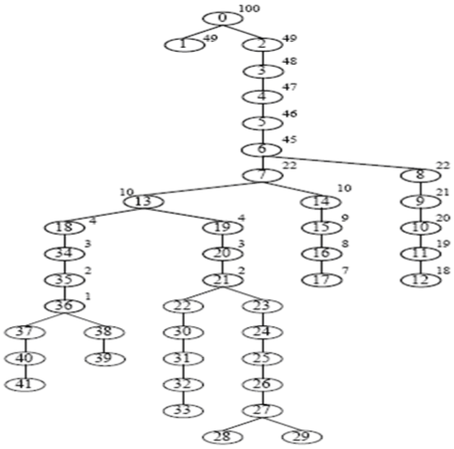

Figure 1

When a non-relay router receives the request packet, it uses the TTLnr value to send its response back to the querier and modifies this value for the request packets that it forwards on the tree. In addition, in order to prevent pre-mature scope expiration (due to IP TTL expiration in the network) each compliant router on the multicast tree adjusts TTLIP and TTLtt values according to TTLnr value.

As an example, consider the tree topology in Fig 4. According to nr-response, the querier will receive responses only from the root router (0), branching routers (nodes 6, 7, 13, 21, 27, and 36) and leaf routers (nodes 1, 12, 17, 28, 29, 33, 39, and 41). Therefore, the querier will learn the exact same topology information but will receive fewer responses (15 responses instead of 42 in this particular example). Thus, based on branching characteristics, we can reduce both the number of rounds and the overall discovery time.

Deployment Issues

In this section, we discuss potential tracetree deployment issues. One important issue is security in terms of using tracetree for launching denial-of-service attacks. This is possible if the tracetree functionality is accessible by any user. We presented mechanisms to make launching attacks more difficult and discussed how to reduce the effect of potential attacks. Ideally we expect these measures to provide sufficient assurance for the deployment of searchtree in the Internet. However, these mechanisms may not always be satisfactory for all users (ISPs). Considering this possibility, instead of completely turning off searchtree functionality in routers, concerned users can use a more controlled operation environment for tracetree. In this scenario, we use an agent based tracetree topology collection mechanism similar to the Multicast Consolidated Proxy Monitor (MCPM). Fig 2 shows the steps. In this approach, each domain allocates a well-known tracetree agent responsible for running all tracetree queries in the local domain.

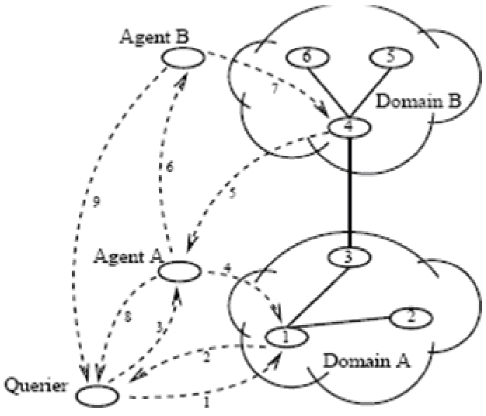

Figure 2

1. Querier sends a query packet to first hop router1.
2. First hop router1 responds with the address of the tracetree agent Agent A.
3. Querier sends its query to Agent A.
4. Agent A sends a query to first hop router 1 during topology discovery, on tree routers in domain send their responses back to the Agent A.
5. When router 4 receives a request from router 3 it sends the address of Agent B to request destination i.e. Agent A
6. Agent A forwards query to agent B.
7. Agent B sends query to router 4. During the topology discovery, on tree router in domain B sends their responses back to the Agent B.
8. Agent A sends the collected responses back to the querier.
9. Agent B sends the collected responses back to the querier.

All the routers are configured to accept searchtree query messages only from the local tracetree agent in their domain. Since tracetree is limited to supporting requests coming from a well-known agent site, secure communication primitives can be used to provide authenticated message exchange between the agent site and the routers.

Once a searchtree agent receives a query packet, it runs the query in the local domain, collects the responses, and sends them back to the querier. In cases where a tree topology spans multiple domains, tracetree agents in adjacent domains communicate query messages between each other so that a searchtree agent in each domain traces the portion of the multicast tree in its own domain and then sends a response back to the original querier.

In addition, in an agent-based deployment scenario, tracetree agents can cache the collected topology and use this information for subsequent queries. Moreover, agents can perform additional operations such as hiding the actual IP addresses of the routers in order to protect privacy of the internal network topology.

In summary, even though we prefer a native/standard deployment for search tree, we expect the agent-based deployment to provide a reasonably good assurance for ISPs to support this service in their networks.

Another important issue is the interaction of tracetree with the multicast routing protocols. searchtree uses existing multicast forwarding states in the routers. The multicast routing protocol deployed in the network may be using unidirectional or bidirectional trees and may be building source specific or shared trees.

Searchtree is insensitive to packet encapsulations used in some of the multicast routing protocols such as the Multicast Source Discovery Protocol (MSDP) and Protocol Independent Multicast-Sparse Mode (PIM-SM). In MSDP, when a new source starts sending to a multicast group, the Randezvous Point (RP) in the source domain uses MSDP Source Announcement (SA) messages to announce this new source to the RPs in remote domains. Later on, when the group receivers in these remote domains learn the existence of this new source, they use PIM-SM to establish a multicast forwarding path toward this new source. Therefore, searchtree cannot effectively return the actual multicast tree topology between this new source and the remote group receivers until the underlying forwarding tree is established.

Software Analysis and Design

Requirement analysis

Searchtree is simulated on a LAN network using TCP or IP sockets. This Querier function here is designed in the server. When the multiple clients need to communicate with each other, first the clients are analyzed by the querier, to determine whether the clients existing or not, for tracing multicast route in the network.

Searchtree is analyzed and designed to consider both complete statistics of the querier and clients' model to make decision about multicast route.

Function oriented Design

Function oriented Design relies on decomposing the system into a set of interacting functions with a centralized system state shared by these functions. Functions also maintain local state information, but only for the duration of their execution. This activity involves drawing and analyzing Request Flow Diagrams (RFD).

A Request Flow diagram describes how the output is derived from the input through a sequence of functional transformations. Request flow diagrams show functional transformations but do not suggest how these might be implemented.

A system described in this way might be implemented as a single program using functions to implement each transformation. The programming has to be done for the following events.

Client-Server Connection

The Client-Server Communication model is used to realize the working of the Traceetree mechanism in this project. Hence we are designing a server (querier) and many clients (sending (request/response) and receiving (request/response)) processes, socket mechanism is used for exchanging data between processes. These Processes can be either is on the same machine, or on different machines connected via a network. Once a socket connection is established, data can be sent in both directions until one of the endpoint closes the connection.

Generally, the process making the request is called client, and the process servicing the request is called the server. Here the server creates two sockets one for transmitting client and other for the receiving client.

We will briefly go over the steps in a typical client-server connection. The following table outlines these steps:

Table 1

Server	Client
1. Establish a listening socket and wait for connection from clients.	
	2.Create a client socket and attempt to connect to server
3.Accept the client’s connection attempt	
4.Send and receive data	4.Send and receive data
5.Close the connection	5.Close the connection

First, the server creates a listening socket, and waits for connection attempts from clients. The clients create a socket on its side, and attempts to connect with the server. The server then accepts the connection, and data exchange can begin. Once all data has been passed through the socket connection, either endpoint can close the connection.

Conclusion

In this work, we have proposed a mechanism, tracetree, for multicast tree topology discovery. It requires relatively little additional router support and relies only on forwarding state. We argued that the alternative approaches (SNMP and mtrace-based approaches) have requirements or limitations that significantly limit their use for topology discovery. A benefit of searchtree is that it provides tight control on the number of request messages that are forwarded throughout the tree. In this respect, we discussed a number of issues related to searchtree based topology discovery. In addition, we have evaluated tracetree by comparing it to the alternative approaches. We have shown that tracetree is comparable or superior to the alternative approaches in terms of topology discovery overhead and topology discovery time. In addition, searchtree can be used in both intra- and interdomain and it can tolerate the existence of noncompliant routers in the multicast tree. We believe that our technique provides a scalable and efficient way to discover a multicast tree's topology in real time while requiring marginal additional functionality in routers.

References

[1] C. Diot, B. Levine, B. Lyles, H. Kassem, and D. Balensiefen, "Deployment issues for the IP multicast service and architecture," IEEE Network, vol. 14, pp. 10–20, Jan./Feb. 2000.

[2] K. Sarac and K. Almeroth, "Supporting multicast deployment efforts: A survey of tools for multicast monitoring," J. High Speed Networking-Special Issue on Management of Multimedia Networking, Mar. 2001.

[3] S. Paul, K. K. Sabnani, J. C. Lin, and S. Bhattacharyya, "Reliable Multicast transport protocol (RMTP)," IEEE J. Select. Areas Commun., vol.15, pp. 407–421, Apr. 1997.

[4] S. Jagannathan, K. Almeroth, and A. Acharya, "Topology sesitive Congestion control for real-time multicast," inWorkshop Network Operating System Support for Digital Audio and Video (NOSSDAV), Chapel Hill, NC, June 2000.

[5] A. Adams, R. Bu, R. Caceres, N. Duffield, T. Friedman, J. Horowitz, F. Lo Presti, S. Moon, V. Paxson, and D. Towsley, "The use of end-to-end multicast measurements for characterizing internal network behavior," IEEE Commun. Mag., vol. 38, pp. 152–159, May 2000.

[6] A. Kanwar, K. Almeroth, S. Bhattacharyya, and M. Davy, "Enabling end-user network monitoring via the multicast consolidated proxy monitor," in Proc. SPIE ITCom Conf. Scalability Traffic Control in IP Networks,Denver, CO, Aug. 2001.

[7] S. Ratnasamy and S. McCanne, "Inference of multicast routing trees And bottleneck bandwidths using end-to-end measurements," in Proc. IEEE INFOCOM, New York, Mar. 1999, pp. 353–360.

[8] N. G. Duffield, J. Horowitz, and F. Lo Presti, "Adaptive multicast topology inference," in Proc. IEEE INFOCOM, Anchorage, AK, Apr. 2001, pp. 1636–1645.

[9] J. Case, K. McCloghrie, M. Rose, and S. Waldbusser, "Protocol Operations for version 2 of the simple network management protocol (SNMPv2)," Internet Engineering Task Force (IETF), RFC 1905, Jan. 1996.

[10] D. Makofske and K. Almeroth, "Real-time multicast tree visualization and monitoring," Software-Practice Experience, vol. 30, no. 9, pp. 1047–1065, July 2000.

[11] W. Fenner and S. Casner, "A 'traceroute' facility for IP multicast," Internet Engineering Task Force (IETF), draft-ietf-idmr-traceroute-ipm- *.txt, July 2000.

[12] HP OpenView network management solution. [Online]. Available: http://www.hpl.hp.com/ [13] P. Sharma, E. Perry, and R. Malpani, "IP

multicast operational network management: Design, challenges, and experiences," IEEE Network, vol.17, pp. 49–55, Mar.–Apr. 2003.

[13] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A transport protocol for real-time applications," Internet Engineering Task Force (IETF), RFC 1889, Jan. 1996.

[14] H. Holbrook and B. Cain, "Source-specific multicast for IP," Internet Engineering Task Force (IETF), draft-ietf-ssm-arch-*.txt, Nov. 2002.

[15] IP router alert option, Feb. 1997, RFC 2113.

[16] R. Chalmers and K. Almeroth, "On the topology of multicast trees," IEEE/ACM Trans. Networking, vol. 11, pp. 153–165, Feb. 2003.

[17] Y. Dalal and R. Metcalfe, "Reverse path forwarding of broadcast packets," Commun. ACM, vol. 21, no. 12, pp. 1040–1048, 1978.

[18] J. Pansiot and D. Grad, "On routes and multicast trees in the internet," ACM Comput. Commun. Rev., vol. 28, no. 1, Jan. 1998.

[19] K. Almeroth, "A long-term analysis of growth and usage patterns In the multicast backbone (MBone)," in Proc. IEEE INFOCOM, Tel Aviv, Israel, Mar. 2000, pp. 824–833.

[20] A. Ballardie, "Core based trees (CBT version 2) multicast routing," Internet Engineering Task Force (IETF), RFC 2189, Sept. 1997.

[21] D. Meyer and B. Fenner, "Multicast source discovery protocol (MSDP)," Internet Engineering Task Force (IETF), draft-ietf-mboned-msdp-*.txt, Nov. 2002.

[22] S. Deering, D. Estrin, D. Farinacci, V. Jacobson, G. Liu, and L. Wei, "PIM architecture for wide-area multicast routing," IEEE/ACM Trans. Networking, vol. 4, pp. 153–162, Apr. 1996.