Reliable & Scalable Multicast Communication in Real Time Collaborative Systems

Jayesh M Patel A/L Maganlal [1] and Shamsul Sahibuddin [2]
[1,2] Collaborative Working & Distributed Applications Group,
Faculty of Computer Science and Information System,
Universiti Teknologi Malaysia, 81310 Skudai, Johor, Malaysia.
Tel.: +60-7-5503570, Fax.: +60-7-5565044
[1] calicatel@maxis.net.my
[2] shamsul@fsksm.utm.my

Abstract: *The world wide web (WWW) already accounts for more Internet network traffic than any other application, including email and simple file transfer. It is also a collaborative technology in a weak sense of the word - it allows people to share information. Synchronous collaboration is where an interactive activity is simultaneous and in real-time. Computer based real time collaborative systems like shared whiteboards, collaborative editor etc. are only beginning to emerge recently. These applications involving more than two users exchanging information, require multicast communication. Multicast communication is a transmission mode that is now supported by a variety of local and wide area networks. Multicasting enables multiparty communication across a wide area to sparsely distributed groups by minimizing the network load. Multicasting itself is one of the key technologies in the next generation of the Internet. This paper describes the technical issues from the aspect of multicast communication and its reliability in synchronous collaborative application.*

1. Introduction

Point to point communication models and protocols such as TCP, were designed merely for the use in client-server applications. As the demand for collaborative applications increases, communication involving multi point and proper network bandwidth utilization arises as an issue. In IPv4, however, multicasting was introduced as an extension of the basic specification; hence, IPv4, nodes do not necessarily support multicasting. On the other hand, specifications of IPv6 require that all IPv6 nodes support multicasting. Recently, with the explosive growth of Internet, Internet Protocol Version Six (IPv6) has been introduced to overwrite the existing version of Internet Protocol (IP) and overcome the shortage of IP addresses. This paper will focus on the multicast service with emphasis on IPv6 and its reliability issue in synchronous collaborative applications. The final part of this paper describes the current stage of the ongoing research.

2. Reliable Multicast

Different multicast applications require many different levels of reliability and ordering guarantees in the face of transient network failures such as dropped packets[1]. Although multicasting has an enormous potential to reduce application bandwidth requirements, its lack of reliability remains a problem and how multicasting system handles lost packets becomes an issue[9]. A synchronous collaborative application requires reliable packet delivery to ensure data consistency. Data must be sent error-free to ensure that the images and text send are correct and latency requirements remain low, as extreme delay greatly reduces the utility of a real-time data application in a collaborative environment[11]. Types of multicast application can be categorized as:

i. Push content - video streams, audio streams, caching servers
ii. Multimedia collaboration - audio and video conferencing
iii. Resource discovery
iv. Remote sensing

Most applications running above Transport Control Protocol/Internet Protocol (TCP/IP) use TCP as the Transport layer for the rich services it provides. TCP, however, provides only point-to-point (unicast) services. Thus all multicast applications must run on top of User Datagram Protocol (UDP)[11]. UDP on the other hand is unreliable and if UDP detects an error in the received packet, it silently drops it[5]. Lost packet has to be retransmitted again and bandwidth utilization increases.

In addition to delay and data loss intolerance there are other reliable multicast application requirements. For example, some applications require verification that all receivers have received all data. However, delay and data loss intolerance are the most significant. Multicasting is a best effort service without guarantees about delivery or correctness. It is left to the multicast transport protocol to implement reliability. Although multicasting has the advantage of saving bandwidth, yet it has drawback problems in reliability[9]. More than 20 protocols for reliable multicasting have been proposed. These protocols must contend with issues such as sources being overwhelmed by feedback from receivers, lost packet recovery, the ordering of packets and the distribution of the receiving group. As a result, it's unlikely that a single protocol will be the solution[9].

Another reliability issue is how the multicasting system handles lost packets. The packets can be recovered by having the sender retransmit the lost information by multicast to the group, but the disadvantage of this method is that all receivers, whether or not they lost the packets, receive the retransmitted data[9].

2.2 Multicast Transport Protocol

Normal TCP is not possible, because it is designed to be one to one protocol. TCP can keep state information only about one connection, whereas in multicast connection there is in principle state information for each sender-receiver pair. One multicast transport connection can involve hundreds or even thousands of receivers. Different receivers may lose different packets and packets can arrive in different order. Reliable transport protocol must handle all these situations, so that every receiver receives consistent information. Therefore, designing reliable, scalable and efficient multicast transport protocol is a really demanding project.

The Internet Engineering Steering Group (IESG) recognizes the importance of reliable multicast transports, yet they still consider existing reliable multicast transports such as MTP, MTP2, RMTP and others as experimental and subject to research, not for wide deployment[12]. Although they are still subject to refinement, a number of documents have attempted to describe the requirements for reliable multicast transport implementations. The short list is: scalability, congestion control, error recovery, and robustness[7][12][13]. Current focus of the reliable multicast transport protocol is on solving the needs of a particular application and not as a general solution.

2.3 Scalability Issue

Many group applications require a full reliable data transmission to a large number of receivers while still exploiting the advantages of multicast communication, such as bandwidth saving and maintaining a suitable performance in heterogeneous environment. How well can multicast transport protocol handle large number of receivers is the scalability issue. Data transfer is reliable if all receivers will get an error free copy of the transmitted data. Error control algorithms are necessary to ensure reliable data transmission[13]. Since multicast sessions involve multiple sources of feedback, this method would lead to feedback implosions at the source[7].

3. IPv6

IPv6 plays a more important role by increasing network protocol functionalities and its performance, which in turn enables development and deployment of new applications over the Internet. In order to facilitate the creation of applications that tap into IPv6 features, the deployment of global IPv6 are very important. Successful deployment of IPv6 depends on the reliability, functionality, security and availability of its networks and equipment.

3.1 IPv6 Multicasting

IPv6 multicasting is quite like IPv4 multicasting in that it is possible to send one packet to multiple receivers by sending to a special address. This address specifies a multicast group, which a receiver must join so that they can receive the packets. In IPv4 multicast groups have the most significant byte set to 224, given addresses of the form 224.x.x.x. In IPv6 multicast groups have the left most 8 bits set, giving a prefix of ff00/8. The next 3 bits are not used currently and must be unset, while the next bit specifies whether the group is transitory or permanent[4]. Systems engaged in a teleconference, for example, can use a transient group address[5].

3.2 IPv6 Multicast Addressing Scheme

Address space in IPv6 is 128 bits and from this address space a fraction of 1/256 is allocated to multicast addresses. Binary prefix of multicast addresses in IPv6 is 1111 1111. The rest 120 bits are divided to three fields: 4 bits for flags, 4 bits for scope and remaining 112 bits for group ID[10]. Mapping of IPv6 multicast addresses to ether net MAC addresses is similar to IPv4 mapping, but low order 32 bits of group address are mapped to MAC address instead of low order 23 bits in IPv4 specification[2].

4. Synchronous Collaborative Systems

Basic support for synchronous work is provided by application sharing systems like Shared X and Xshare or audio / video conferencing such as NetMeeting and NetShow. These systems allow the sharing of standard applications between workstations. The application window is identically displayed on multiple workstation screens while maintaining the audio / video streaming. Some challenges faced by developers in developing synchronous systems are[14]:

i. Minimum response time
ii. Session control
iii. Communication mechanism
iv. Flexible edits

Other than that most synchronous collaborative system have centralized architecture, in a sense that the synchronization process is done by a main server. This centralized architecture suffer from bottleneck where the performance of the whole system is affected. Bottleneck is an important issue in the design of a centralized collaborative application since it influences the performance[3].

5. Current Implementation

This research is currently into the development stage where a prototype will be developed. The prototype will have the function of a collaborative system such as audio conferencing, collaborative text editing and shared whiteboard. The prototype will be developed to function on both IPv4 and IPv6. Since native IPv6 based network is not yet available, our current study shows how transition mechanism can be used to migrate from IPv4 to IPv6. Based on the study done, three major mechanism have been suggested[8]:

i. Dual Stack IP
Has the ability to support both IPv4 and IPv6 on the same machine. These nodes have the ability to send and receive both IPv4 and IPv6 packets. The IETF has drafted 2 dual-stack transition tools: Dual Stack and DSTM. As shown in Figure 1, packets can use native IPv4 addresses between IPv4 clients, native IPv6 or IPv4-compatible-IPv6 addresses between IPv6 or IPv4 clients and IPv4-mapped-IPv6 addresses between IPv4 clients[6].

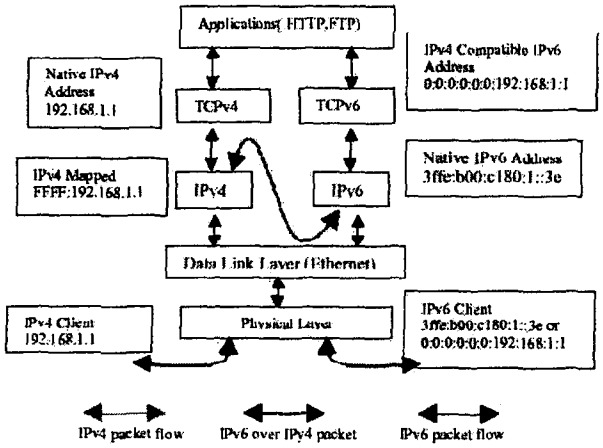
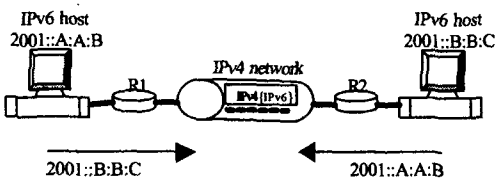


Figure 1: General Protocol Layers for Dual IP Stacks

ii. Tunneling (IPv4-over-IPv6)
Tunneling allows the use of IPv4 networks to carry the IPv6 traffic. This can be done without changing any IPv4 infrastructure. As shown in Figure 2, a border router can encapsulate IPv6 packets to the IPv4 packets and sending it through the IPv4 network, which is transparent to the network. The routing node which sits on the other side of the border will decapsulate the packets and route it to the appropriate destination.



R1: encapsulates IPv6 packet into IPv4
R2: decapsulates IPv4 packets into IPv6

Figure 2: Tunneling mechanism interconnecting IPv6 host via IPv4 network

IPv6 over IPv4 tunneling in complete IPv4 topology involves four main configurations: host-to-host, router-to-router, host-to-router and router-to-host. Other tunneling tools that has been drafted by IETF are 6to4 and ISATAP. ISATAP is a tunneling method that embeds IPv4 address into the 64 bits Interface ID in IPv6 address.

iii. Translation
Translates IPv6 protocol, application etc to IPv4 and vice versa. IETF has drafted several translation tools such as:

NAT-PT: Does address and header translation. Device resides at boundaries between IPv4 and IPv6 network.

SIIT: Function same as NAT-PT but does not have any address translation (Stateless).

BIS: Translation method. Sits between Network Card driver and network protocol.

BIA: Host based IPv4/IPv6. Translation between IPv6 API's and IPv4 API's.

5.1 Implementation Using Windows Socket

IPv6 can be implemented on Windows based application by adding IPv6 capability to windows socket. This can be achieved by changing the data structures. Avoid any hard-coded IPv4 addresses in the application. To port existing code base from IPv4 to IPv4- and IPv6-interoperability, acquire a file called Checkv4.exe from Microsoft IPv6 Technology Preview for Windows 2000. Run this file to check the applications code. This utility is used to alert the usage of *sockaddr* or *sockaddr_in* structures in your code. Replace this with the new *SOCKADDR_STORAGE* structure. Typically many application developers use *sockaddr* structure to store protocol-independent addresses, or the *sockaddr_in* structure for IP addresses. Neither both of these are large enough to hold IPv6 addresses, and therefore both are insufficient if your application is to be IPv6-compatible.

6. Future Work and Summary

In collaborative work, communication is an important factor, especially in synchronous system where real-time applications are concern. Multicast communications are better than unicast communication as unicast communication does not scale well with the number of recipients and increases network load by the reciprocal of the group size. Multicasting reduces the transmission overhead on the sender and, it can reduce the overhead on the network and the time taken for all destinations to receive the information. IPv6 on the other hand has been designed to enable high-performance, scalable internetworks to remain viable well into the next century. Nevertheless, the switch will have to come and most of the next generation applications will be IPv6 native.

This research will continue into the next stage where the testing of the complete prototype will be done and results of the analysis would be studied and interpreted. The next step of this research will look

into the types of testing that should be done on the prototype, what kind of tools should be used for the testing, what should be measured during the testing, what results should be used as a comparison and what are the expected results.

7. References

[1] Callahan, J., Montgomery, T. and Whetten, B., "*High Performance, Reliable Multicasting: Foundations for Future Internet Groupware Applications*", NASA/West Virginia University Software IV & V Facility, 1996.

[2] Crawford, M., "*RFC 1972: A Method for the Transmission of IPv6 Packets Over Ethernet Networks*", 1996.

[3] Dewan, P., "*Architectures for Collaborative Applications*", In Beaudouin-Lafon, M. (Ed.), Computer Supported Cooperative Work, Trends in Software Series 7, John Wiley and Sons Ltd., 1999.

[4] Elphick, D., "*IPv6 Multicasting*", Southhampton University, 2000.
http://www.ecs.soton.ac.uk/~dreoor/multicast/report/ipv6_multicasting_background.html

[5] Forouzan, B. A., "*TCP/IP Protocol Suite*", McGraw Hill, 2000.

[6] Gilligan, R. and Nordmark, E., "*Transition Mechanisms for IPv6 Hosts and Routers*", RFC 1933, April 1996.

[7] Hoffmann, M., "*Scalable Multicast Communication in the Internet*", University of Karlsruhe, 1996.

[8] Karuppiah, E. K., Kurup, G. and Yamazaki, T., "*Application Performance Analysis in Transition Mechanism from IPv4 to IPv6*", Research & Business Development Dept, NTTMSC, 2001.

[9] Kosiur, D., "*(multi)casting a Reliable Net*", PC Week, ZDNet, 1998.
http://www.zdnet.com/zdnn/content/pcwk/1514/302986.html

[10] Luoma, J., "*Tik-110.551 Internetworking Seminar: Multicasting in the Internet*", AvantComp Oy, Matti Tella, 1997.
http://www.tml.hut.fi/opinnot/Tik-110.551/1997/multi.htm

[11] Miller, C. K., "*Multicast Networking and Applications*", Addison Wesley Longman Inc., 1998.

[12] Quinn, B., "*Reliable IP Multicast*", Stardust.com Multicast Technical Resource Center, 1999.
http://www.stardust.com/multicast/whitepapers/ReliableIP_01.htm

[13] Rezende, J. F. and Fdida, S., "*Scalability Issues for Reliable Multicast Protocols*", University Prerre et Marie Curie, 1999.

[14] Yang, Y., Chengzheng, S., Yanchun Z. and Xiaohua J., "*Real-Time Cooperative Editing on the Internet*", IEEE Internet Computing, 2000.