Eliminating Redundant Link Traffic for Live Multimedia Data over Distributed System

K. Rajkumar[#1] & P. Swaminathan[*2]

Computer Science & Engineering, School of Computing,
SASTRA University, Tirumalaisamudram, Thanjavur-613401,Tamilnadu, India.
[1]rajkumar@cse.sastra.edu

* Computer Science & Engineering, School of Computing,
SASTRA University, Tirumalaisamudram, Thanjavur-613401,Tamilnadu, India.
[2]deanpsw@sastra.edu

***ABSTRACT*: Many internet applications need live multimedia data to be transmitted over the network. To incorporate the same, applications like video conferencing and e-learning need IP multicast or unicast. However in internet applications, lack of multicast services play a deterrent role in group communication. When live multimedia data is transferred to more than one receiver over the network, the more number of duplicate packets are transferred. Due to this duplicate packets, bandwidth and other resources are not properly utilized. When network connection is slow then clustering duplicate packets takes place in link. Due to this clustering, there is a chance for the data loss to occur. A CacheCast mechanism can be used on the router to eliminate such type of duplicate packets. Cache is placed on the router which acts independently for the sender and receiver. In single source multiple destinations, while transferring live multimedia data to more number of receivers, the duplicate packets may occur on the link. All duplicate packets are removed at the nearest router of sender and retrieved from the nearest router of receiver. This technique is used to achieve the better utilization of bandwidth. When send the live data to more than one receiver, bandwidth is efficiently used and the receiver's side original data has to be reconstructed without any discrepancies.**

Keywords: IP multicast, CacheCast, Multimedia data, Group Commuincation, Multicast Services, Video Conferencing, Bandwidth, Unicast, e-learning

I.INTRODUCTION

In the past few years live media streaming applications such as video conferencing, e-learning has become more and more popular among the internet applications. Many multimedia applications needs live multimedia data to be transferred over the distributed network (Yifeng He; Ling Guan, 2009). In internet, the multicast service does not support single source multiple destinations data transfers (Naixue Xiong et al.,2010). In internet the single source multiple destinations in terms of unicast or Application Layer Multicast (ALM) only (Srebrny, P et al., 2010). So, multicast service is not efficiently supported by the internet, all the multicast applications suffer a lot. Redundant links are used to connect the system in a distributed network. Both IP multicast (Yifeng He; Ling Guan, 2009) and Application Layer multicast creates more number of duplicate packets which cause traffic in the redundant link. This in turn is very difficult to deal with multicast congestion control (Matrawy, A.; Lambadaris, I, 2003). Delay and data loss may occur due to traffic. These duplicates can be detected and removed to reduce the traffic in the redundant link. In live streaming, packets are constructed and delivered to the receiver (Ling Chen, 2005). To reduce the traffic, duplicate packets should not be sent over the link. Thus these packets have to be identified first. Videos are sent over the network as a sequence of frames or packets (Hui Guo et al., 2009). More duplicates can be identified by comparing with previous frames or packets which is stored in the first session of the data transfer. This type of comparison technique is called CacheCast (Srebrny, P et al., 2010). It should contain small amount of memory to keep the packet or frame and placed on the router. The basic Cachcast mechanism explained through Fig. 1. In Fig1, P means payload, 0 means that the packet is original, 1 means that the packet is duplicate and A, B are receiver's address. One particular case is taken into consideration. Here a packet is transferred from sender, traverse a few nodes over a common route and is sent to different receivers. The first packet transfer as usual through all the intermediated nodes or hops. While transmitting the first packet or frame, each intermediate node or hop keep the packet's payload and records its output link in its own cache. When the second packet or fame enters the router, it compare with previous packet or frame's link and payload. If it is same as previous one, then simply discard the payload of that packet otherwise forward to next hop or node. When no duplicate is transferred over the link, traffic is reduced. The original video can be accessed from the nearest router's cache. Cache is placed over every router in the network, so

that duplicates can be removed on the entry and recovered at the exit (Srebrny, P et al., 2010, Hui Guo et al., 2009).

CacheCast is a caching mechanism for single source multiple destination data transfer. It makes the transmission more efficient in a multicast communication. It enables one-to-many data transfer in the internet with multicast technique. It needs to identify the frame or packets that are part of the single source multiple transfers as cacheable, add some more information to the header, and forward the frames or packet with the same payload. First, it identifies and marks the same packet or frame that carries the same payload as cacheable frames. It will not carry duplicate payloads. Whenever a router receives a packet or a frame, it compare with its cache data. If received packet is duplicated, then drop the payload and forward only the header of the packet to next router. It will repeat the same process until the packet enters in to the new channel.

CacheCast mechanism is not allowed to transmit duplicate packet while sending live multimedia data to more than one receiver through the same link or channel. Bandwidth is efficiently utilized during the process of transmissions of large volume of data.

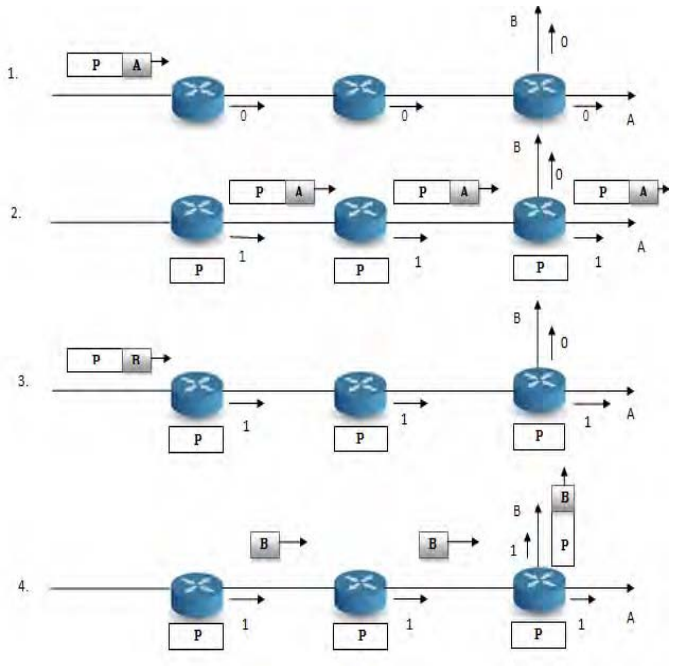

Fig. 1. Idea of CacheCast Mechanism

(P- payload of the packet or frame
A ,B - Receivers & header of the packet
0 – new packet transmission thru that channel
1 – packet transmitted already thru that channel)

II. PLACEMENT OF CACHE

Every packet or frame which is built in the sender side has to be checked for redundancy. So cache has to be placed at a place where every sending packet passes through. Cache can be placed on the machine by itself but it may reduce the system performance. It requires some space in the system. Separate maintenance has to be done for clearing the cache. It adds some complexity to the system. Cache can be placed on the router so that it is not disturbing the system. As the router decides where the packet should go, each packet to the receiver will pass through the nearest router. Thus every packet will be sent to the receiver through the router only. Placing cache on the router has more advantage over placing cache on the system. When the router fails there is no connection. If we allocate space for the Cache over the system, whenever router fails, every system memory, which is allocated for the cache goes waste. So the cache is needed to be placed on the router. The router decides which packet has to go on the link and which system it should go. Even if the cache placed on the router needs maintenance, it does not affect the system performance which generates the data to be sent. Every router in the multicast network is allocated with a cache as soon as the multicast network is established among the peers. Every packet is checked for redundancy before it is placed on the link. When less number of packets is sent, it travels faster than large number of packets. There shouldn't be any alteration in the data as the redundancy is removed. So, the original data is taken from the exit router which is nearer to the receiver. The duplicate packet is removed at the entry router which is nearer to the sender and it is recovered at the exit router. As the duplicate packets are removed there won't be any traffic which causes delay, data loss, and so on. But as the router holds the cache, the process of eliminating the

duplicate packet may take some amount of time but it is less when compared to the delay caused by the transfer of duplicate packet over the link.

III. CACHE ON SENDER SIDE

When sender starts sending the data the first frame has to be stored in the cache on the nearest router of the sender side. When the next set of data is ready to be sent, it is compared with the previously stored data. Then only the unique frame is transferred over the network. When live multimedia data is transferred it will be sent in the form of sequence of frames (Ching-Ying Lin; Verscheure, O.; Amini, L, 2005). So the first frame of video is stored in the cache. We can access the frames which forms the video. After extracting all the frames it has to be stored in an array. Next set of data is accessed and all the frames must be extracted from the video and it has to be stored in another array. As soon as the frames from the next upcoming video are extracted, it has to be stored in an array and compared. The array which is having the first video frames and the array which is having the next video frames have to be compared. There is a chance for the duplicate to arise at the same place in the multicast network. So sometimes duplicate also occurs in large amount. Thus at that time only very less amount data is sent over the link. When the duplicate frames are identified in the array then their indexes are recorded first, then the index and unique frames are sent over the link. Thus for every instance of time, more number of frames have to be sent by the sender is verified. The index of the array which is sent first will help in removing the duplicate frames. Then the unique frame is transferred and the video is built from the newly received frames. There won't be any discrepancy in the video as the frame which is removed on the side is correctly inserted in the array at the same place on the receiver side. The corrected data is sent over the link (Ching-Ying Lin; Verscheure, O.; Amini, L, 2005).

IV. CACHE ON RECEIVER SIDE

The initial frame which is stored in the sender side cache is also stored in the receiver side cache as it is going to be used for building the video which is to be sent to the receiver. When sender identifies the duplicate frame's indices in the array it is sent first. The same process of retrieving frames from the live video and storing in the array is done on the receiver side cache also. First the array index which has the duplicate frame is sent from the sender. Those frames are first deleted. Unique frame which is to be sent next is received in a separate area by the cache. After the deletion of frames from those indices it becomes empty. The received frame is placed in the cache accordingly as per the number of indices which the frame holds. When the frames are placed in their respective position then the array of frames from which the video to be sent is built and sent. The duplicate frames are not sent but the original frame to be sent through the link. Redundant link traffic is highly reduced by this process. Unique frames are received in a separate place by the cache without overwriting the existing video frames. Only after deleting the frame based on the indices, the received frames have to be placed in their respective indices. The video which is built from the frames should have the same frame rate as that of the original video which is sent over the network (Srebrny, P et al., 2010, Ching-Ying Lin; Verscheure, O.; Amini, L, 2005,). The frame rate at which the video is built is taken a special care as it may affect the data sent over the network. The actual data is retrieved from the exit router which is placed near the receiver system. No duplicate frame is sent over the network but the actual data which is sent is received as the same actual data at the receiver end.

V. CACHE MAINTENANCE

A fixed size memory of space is allocated for the cache on the router. It should be enough to store the frame or packet, receive the next set of frames, compare and fix the duplicate on the sender side cache. The cache on the receiver side should be able to store the initial frame, receive unique frame and satisfies other memory needs. As the cache is placed on the router it has to be monitored to maintain it. The next sequence of data must be retrieved with same size as that of the initial frame. No more extra memory is needed for storing the next data. No memory over-flow occurs when the next data is retrieved properly from the sequence. When the length of the video sequence retrieved is same as the original video then both the videos should have the same number of frames. Then there won't be any over-flow in the cache memory. Monitoring of cache is done only for checking the proper functioning of the cache. Very less amount of memory is needed to be allocated for the router. Router should keep track of the cache memory of the sender and receiver for which it holds the cache. The stored data on the router can be cleared when a peer goes out of the group (Hui Guo et al., 2009). The next stored data must be cleared subsequently as the other data arrives. When unique frames from the video are sent, the stored video has to be cleared as the next video sequence arrives in the link. The memory space which receives unique frames should be same as the memory which store initial packet. Sometimes the unique frame may be less. Sometimes the memory goes waste, when only less number of unique frames arrives. Router has to hold the memory for sending and receiving. Holding cache may add some extra burden to the router. But it works smoothly when no problem occurs with storing or retrieving of data from the cache. Even if some problem occurs it should not affect the router in any way (Tay, Y.C.; Hweehwa Pang, 2000).

VI. EXPERIMENTATION

Fig. 2. Finding duplication

Fig. 3. Making video based on received Frame

This is implemented through JAVA. Initially we can access all the frames from source and forward towards destination. Each and every frame is compared with the subsequent frame as per the caching principle (Fig. 2). After the duplicates are found those frames are removed and only unique frames are sent over the link. If the receiver successfully received the frames, then they created a video file. While creating video Width, Height and Frame rate should not change. so that there won't be any discrepancy in the original data. Sender and Receiver do not know that these duplicate frames are removed in the transmission. As no duplicate is transferred there won't be any traffic over the link. Due to this, Bandwidth and other resources can be efficiently utilized. The data which is sent from the sender is delivered as such, without any change to the receiver. But no duplicate data can be transferred over the link which reduces traffic over the link and carries all the Frames (Fig. 3).

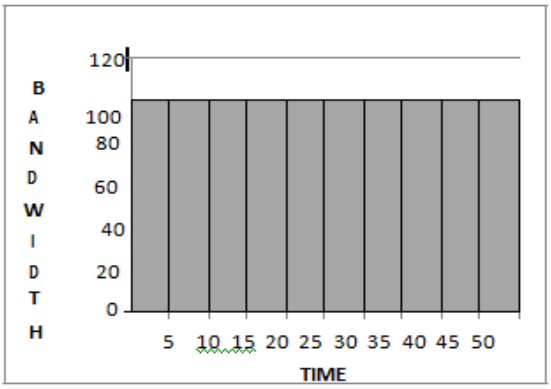

Fig. 4. Without Caching Principle

If no cache is placed on the router, then the duplicate is not removed in the data sequence. Every packet is sent over the link which utilizes full bandwidth. The traffic over the redundant link will be high. Due to this traffic delay, data loss may occur (Fig. 4).

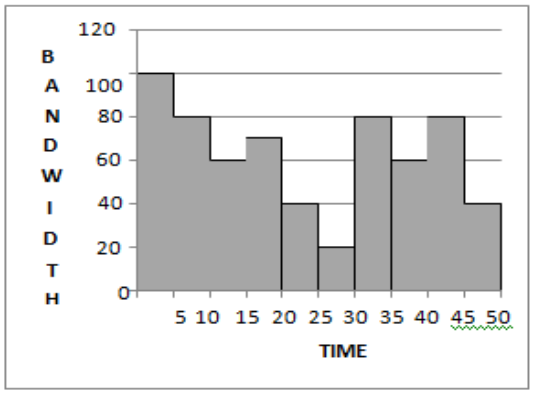

Fig. 5 With Caching Principle

If cache is placed on the router, then the duplicates in the data sequence are removed. So, perfectly utilize the bandwidth while sending the large volume of data (Fig. 5).

VII. CONCLUSION

In this paper, duplicate packets are not sent over the link. Bandwidth is efficiently utilized when large number of packets is sent. Utilization of bandwidth may be varied dynamically while sending the large volume of data like live multimedia data as shown in Fig. 2. Even a small length video has large number of frames. There's high probability for the occurrence of duplicate frames in the video. As there is no duplicate in sending and receiving, there won't be any delay. Cache is placed on the router so it doesn't affect the system performance in anyway. Router is the main thing which decides the receiver that needs to receive the data packets. When router fails in any situation, then there is no interconnection in the network. When cache is placed on the system and when router fails in it, then memory space allocated for cache goes waste. Router is the correct one to place the cache. The performance of the network increases as no duplicate is sent. Duplicate is identified by performing comparisons. Time taken for comparing is far less than delay caused by the redundant link traffic. There is no centralized controller, so maintenance is not done by the sender at a particular instance. Randomly it should be monitored based on the performance.

REFERRENCES

[1] Yifeng He; Ling Guan, Optimal resource allocation for video communication over distributed systems. IEEE Conference Publications, Page(s): 1414 – 1423, 2009.

[2] Naixue Xiong; Xiaohua Jia; Yang, L.T.; Vasilakos, A.V.; Yingshu Li; Yi Pan, A Distributed Efficient Flow Control Scheme for Multirate Multicast Networks. Parallel and Distributed Systems, IEEE Transactions on Volume: 21,Page(s):1254-1266, 2010.

[3] Srebrny, P.; Plagemann, T.; Goebel, V.; Mauthe, A. CacheCast: Eliminating Redundant Link Traffic for Single Source Multiple Destination Transfers, Distributed Computing Systems , Page(s): 209 – 220, 2010.

[4] Hui Guo; Kwok-Tung Lo; Yi Qian; Jiang Li. Peer-to-Peer Live Video Distribution under Heterogeneous Bandwidth Constraints, Parallel and Distributed Systems, IEEE Transactions on Volume: 20, Issue: 2, Page(s): 233 – 245, 2009.

[5] Ching-Ying Lin; Verscheure, O.; Amini, L. Semantic Routing and Filtering for Large-Scale Video Streams Monitoring, IEEE Conference Publications Page(s): 1408 – 1411, 2005.

[6] Ling Chen. A conference control protocol for small scale video conferencing system, Advanced Communication Technology, Page(s): 532 – 537, 2005.

[7] Tay, Y.C.; Hweehwa Pang. Load sharing in distributed multimedia-on-demand systems, Knowledge and Data Engineering, IEEE Transactions on Volume: 12 , Issue: 3, Page(s): 410 – 428, 2000.

[8] Matrawy, A.; Lambadaris, I. A survey of congestion control schemes for multicast video applications, Communications Surveys & Tutorials, IEEE Volume: 5 , Issue: 2, Page(s): 22 – 31, 2003.