EFFICIENT VIDEO STREAMING TECHNIQUE FOR ONLINE CLASSES DURING COVID PANDEMIC

S. Christalin Nelson and J. Dhiviya Rose

School of Computer Science, University of Petroleum and Energy Studies (UPES), Bidholi, Dehradun, INDIA 248007

Email: {christalinnelson@gmail.com, dhiviyanelson@gmail.com}

Abstract: The internet is composed of heterogeneous network devices. These devices are verily constrained owing to their heterogeneity and due to the same type of multimedia data streamed in common with multicast group members as defined by the distribution tree. The proposed system discussed in this paper discuss an efficient streaming algorithm for multicasting video over a network of heterogeneous devices. In multicasting, same data is sent to all group members and in a heterogeneous environment all members or devices may not be able to reproduce the data well. So we need to find a network device which will gather the information about the display capabilities of the devices like video quality it is able to reproduce at the moment. The video quality is obtained from each receiver and augmented with routing table. This paper presents the implementation of the proposed system to improve video stream and the results obtained after testing

Keywords: Multicasting, J-Sim, Router

1. Introduction

The internet is composed of heterogeneous network devices. These devices are verily constrained owing to their heterogeneity and due to the same type of multimedia data streamed in common with multicast group members as defined by the distribution tree. The objective is to develop a streaming algorithm that makes the video stream sent by the source compatible to the diverse network devices. It also includes identification of a network device (Router) that is capable of querying the members about their configuration and related details based on which the streaming algorithm converts the actual video sent by source to the format compatible with members of the multicast group.

Video multicasting is having a numerous applications in the field of business, education and entertainment. In the multicasting we have the advantage of less bandwidth requirement since same data is sent in common to all in a multicast group. Network processing in the intermediate routers are also less since for many receivers router need to process only one packet. When the same data has to be received by a group of recipient's multicasting is the best option. In this project we are particularly focusing on improving the video quality according to the existing network conditions like network bandwidth, CPU capacity of the receiver. In the field of education also video multicasting can be made use of mainly in the case of distance learning seminars or classes can be taken using video multicasting, since in this application many receivers will be receiving the video same time at different locations. Multicasting learning

video contents will provide bandwidth benefits and moreover will minimize the cost of such applications. Learners from various geographic locations and who are part of the learning multicast group will receive these videos in the format as posted. Only one data needs to be sent in common for all the receivers in the group so the router processing also will be very less. Our project proposes a solution for maintaining the quality of video in spite of the heterogeneity of receivers.

2. Literature survey

Multicasting is a one to many technique in which source will send data in common to all members in a multicast group. Special addressing schemes are used for multicasting. In IPv4 [9] class D address space is used which ranges from 224.0.0.0 to 239.255.255.255. In IPv6 [10] four fields are there in the multicast address. Format Prefix represented by 1111 1111, Flag which can have two values zero and one, if set to zero then the group address is a permanent multicast address, set to one when not a permanently assigned address. Scope represents the scope of the packet, whether it is local or global. Group ID represents the multicast group address.

8 bits	4 bits	4 bits	112 bits
11111111	Flags	Scope	Group ID

Figure 2.1: Multicast address format for IPv6

Multicast group management protocol IGMP [11, 12] is

used to maintain and establish multicast groups. IP hosts

multicast their group membership to their immediate neighboring multicast routers using IGMP. Three types of messages are used to exchange membership information, Membership Query, Membership Report and Leave Group messages. All IGMP messages to the host follows same format.

8 bits	8 bits	16 bits
Type	Max. Response time	Checksum
Group Address		

Figure 2.2: IGMP message format

The multicast router further uses the membership query message in order to know whether members of any multicast group are present in the attached network. The multicast group members respond to the membership query with the membership report message. Moreover, when a group member or host intends to leave a certain multicast group it should provide the concerned multicast router with a leave group message. A distribution tree is maintained in a multicast group with multicast group members as its nodes. Protocol Independent Multicast [13, 14] is used for maintaining the tree. Most commonly there exists two types of distribution trees, namely: the source distribution tree and the shared distribution tree respectively. In source distribution tree a specific tree is formed for a particular source to multicast data in a group. In shared tree same tree will be used by different sources who are multicasting data to the group. PIM works in two modes of operation, dense and sparse mode. PIM dense mode is used when sources and receivers are very close to each other and when there is a frequent flow of multicast data.

In this mode source will forward the data to the first hop router which will multicast the data downstream to other routers, from where they will be forwarding to the multicast group members. A prune message is sent to the upstream router by the concerned downstream router if it finds that there does not exist any multicast hosts. After this, the upstream router stops sending any further messages in this path intended to the downstream router. When a host in the multicast router wants to join the multicast group router will send join messages upstream to the multicast router. PIM sparse mode is used when senders and receivers are not close to each other and when the multicast data is infrequent, this mode makes use of shared tree.

1.1 Multicast Routing Protocol

Routing information is exchanged between the existing multicast routers using the Distance Vector Multicast Routing Protocol, DVMRP [15]. It is an Interior Gateway Protocol (IGP) which can be used only inside an Autonomous System. DVMRP can be used only to route

multicast datagrams. Significant messages used in DVMRP are DVMRP Probe, Report, Prune and Graft. DVMRP Probe messages are used to identify neighbors, alert neighbors about the router's presence and to detect neighbor loss sent in every 30s. A list of neighbors are also exchanged to allow neighbors to know they are identified. DVMRP Report message is used to find neighbors to send data back to the sender. Prune and Graft messages are sent to join and leave a multicast group. This protocol is based on Routing Information Protocol (RIP). Metric used in this protocol is hop count.

2.2 Active Network

Active Network [1] is a network in which intermediate nodes are active, ie. they have a data processing capability. A legacy router can be seen as a store and forward system where an active router can be one with programmable interface attached where the programmable element is present. An active node can have a number of active components present like Network Address Translator, Web Cache or Firewall. So we need to control the order in which these components are processed, there comes the need for service composition.

Active node [1] will be doing the legacy functions like packet forwarding, queuing etc and also an additional function which can be programmed. As we have already seen an active node can have several active components like NAT, Web Cache or Firewall. Our objective is to have an active multicasting component which can scale the video according to the capability of receivers. Several works have been previously done which was intended to provide certain functions like network management, multicasting of multimedia traffic.

An active node with network management [7] capability was developed for managing the network and monitoring. The notes executes the messages which are actually the programs which increase the performance as the programs to near the decision points. Smart packets use the processing power of the active networks for network management. Architecture consists of a packet format which will be encapsulated into the network data delivery service. A high level language called Sprocket and Spanner an assembly language is developed for programming of smart packets. The code is executed after the node verification takes place. Another active component which was previously for making the node active was for multicasting the multimedia traffic [8]. The router plugin implements the active router architecture for multicasting video distribution. Video scaling is done and it is evaluated by considering the load situation in the downstream link after the active router plugin software is installed in the router. The specific header is added between IP header and UDP header with the details of the plugin is used for identification during forwarding the packets. If the referenced plugin is not found in the system it will be downloaded from the code server.

2.3 Extensible Routers

Kernel programming has been in use for making the router platform extensible. A number of software programmable routers have been developed based on this a few to mention are CROSS/Linux software programmable router, Java based OS for software router, Click modular router and Router plugins.

CROSS Programmable Router: CROSS/Linux [5] is a software programmable router platform that combines the resource management capabilities of CROSS and modular configurability of Click. It integrates a remote code downloading mechanism and a multi-hop signaling protocol. This router is dynamically extensible and configurable and also provides predictable processing of network flows. Resource management in CROSS is based on resource allocation. This capability of CROSS is used for scheduling the Click elements with per-flow QoS allocation.

Applications for which it was developed are intelligent video scaling in response network congestion and router throttling as a defense mechanism against DoS attack. Router throttling is a resource management based defense mechanism it is to protect server from excessive service request arrivals over a global network. An installed throttle will limit the forwarding of packets to the server when it exceeds an average value. Real-time video streaming used with video scaling environments preferably apply wavelet encoding. This encoding would incorporate progressive and base enhancement layers. Moreover, the basic and vital information will be housed mainly within the lower layers when

compared to the enhancement layers. In the time of congestion, the router would tend to drop the enhancement layer packets which will help in achieving graceful degradation under constrained network bandwidth.

Janos [4]: Janos is a Java-oriented operating system preferably developed to manage resources with the components including (i) Active Application (AA), (ii) Execution Environment, and (iii) the NodeOS. The hardware layer was taken care by the NodeOS and Execution Environment works on NodeOS which provides an interface calls for a network programmer. Janos operating system provides NodeOS and Execution Environment and it's modelled to avoid application interferences. A Janos node is managed by an administrator. This administrator should be capable to query the state of a node, control the access to node, and provision resources. Janos is capable of control the resources used by the active AAs. It is also important that termination of AAs are closely monitored by Janos, so that it can proceed with retrieval or reclaiming of the provisioned resources. It also provides resource management to a single node. The resources in which Janos focuses for controlling are CPU, memory and outgoing bandwidth. Janos works to control the CPU usage of the particular domains.

The primary focus of Janos is to permit independence in usage of individual components amidst the main components collaboratively working together in an optimized manner. The infrastructure is derived from the popular contributions including (i) The Kaffe virtual machine, (ii) OSKit library. Kaffe is an open source virtual machine based on Java. The major design components in Janos are Moab and Janos virtual machine. Moab is built upon the OSKit. When considering the Janos node, the Janos virtual machine plays a crucial part. The design of JanosVM is followed from Alta OS and Kaffe OS. The Java runtime environment offers a platform for system interaction with malevolent active applications.

The Click Modular Router: Programmable routers can be flexibly built and configured based on the architectures proposed with Click Modular Router [2]. Packet handling components or elements are assembled to form the Click router. Basic functions of a router that could be implemented by each component include scheduling, classification of packets and queuing. A graph is configured by appropriately assembling these components. The edge of this graph would then be equipped to support flow of packets. Elements are implemented as C++ objects. Two specific features of Click are push and pull processing. The connection could be thought of having an end on both upstream and downstream sides. A pull call is initiated by the downstream side whereas the upstream side is capable to initiate a push call. Both these calls (push and pull) together aid in construction of a suitable connection to proceed with transfer of packets.

Thereby the configuration of a Click router appears to be a digraph with its vertices otherwise identified as components or elements. Each component would symbolize an entity of router processing. The vital qualities of a component would include a configuration string, component class, input ports and output ports. Each component has a class that determines its behavior that is Component class. The connection end points between two or more components are considered as ports. A component could have certain number of ports to handle input and output. In order to initialize per-component state, few component classes provision extra arguments. The configuration string for instance contains these arguments.

The components can be realized as objects in OOP language. Every single component class relates to a component subclass in implementation with C++. Configurations in Click router are implemented using two vital constructs, namely (i) Declaration – which insists on the creation of an element or component, and (ii) Connections – portrays the connection between the components, in simple textual language. These configurations fragments incorporate specific extensions relevant for routers. A few vital extensions include (i) policies to schedule and packet dropping, (ii) differentiated services, and (iii) major inclusions for queuing support. Implementation of a scheduler in Click could be done as a

single component. Click implements scheduler components like round-robin scheduler (roundrobinshed) and priority scheduler (priosched). A simple policy of configuring a maximum length queue is adopted for packet dropping beyond the threshold. The edge (border) and core routers manage the collective streams of traffic with the support of the mechanisms proposed by the differentiated services. Click performs well despite of its modularity.

3. Java based router elements for video multicast

The proposed system is designed to develop a plug and play of software components that will ease the way of usage. The J-Socket class of objects were used with the java components deployed earlier on the routers for simulation[6] makes the system easy for creating the reusable components which pave the ways for its application diversity in the area of multicasting. Thus the proposed design can help the connectivity for various devices like palmtops, desktops and laptops by only changing the router's software module and not disturbing the router.

Most of the compatible routers which are implemented in C/C++ are efficient but they are highly dependable on the OS kernel. The intricate dependency make the design inefficient for the plug and play. In the proposed system if the same is designed using the OOPS Java platform independent language overcome the stumbling block.

3.1 Integrating Router J-Sim Modules

Extensible internetworking framework (EIF) is the combination of component based architecture ACA and packet-based network modeling framework. J-Sim is one of the Extensible internetworking framework (EIF). In ACA the components are organized as classes and the ports as the interfaces by which it can communicate outside the class. Each components are constructed with the dependence only during the initial configuration and later maintains its autonomy during the execution. The J-Sim follows a two layer architecture comprising of the core service layer (CSL) and upper protocol layer (UPL). The routing is taken care by CSL and application and transport by UPL.

The first phase of the design is to develop a module to establish communication between the components in the real network. J-Socket following the specification of ACA is used for establishing the communication. If a components is sending the data to the other by the connecting ports it said to be active. J-Sockets has been designed with specific input and output ports. The ports are linked with the packet dispatcher of CSL for the data transmission. This making the CSL running in different machine to communicate vis J-Socket without any kernel dependencies. The simulator used J-Sim follows a specific message format to gain efficiency. This problem is been rectified in the proposed system by adopting a packet conversion methods in the J-

Figure 3.1: Organization of JSocket component

Sockets. Fig.3.1 shows the organization of J-Socket module. Process() manages the input and _start() helps in creating and managing the output. The data that are transmitted is received and interpreted by the packet conversion module built in J-Socket.

3.2 General Multicast System (GMS)

The proposed system uses the single transmission via a common link which can be received by a set of receivers. When it's performed the challenge is meet the needs of the varying customers with various configuration. To reduce the amortize cost scalability has been adopted. This has paved the way shared transmission cost and reduced congestion.

3.3 Leveraging of J-Sim

The wavelet video streaming is performed using GMS. The DVMRP and IGMPv2 is been modified and used to manage the quality of streaming. The group member's information is maintained by IGMP and the routing table by the DVMRP. They act beneath the packet conversion system in J-Sockets. The program helps to negotiate the quality of streaming also. For example, system supports discrete wavelet transform (DWT) encoded video, which consists of a base layer of low-frequency video information, and progressive enhancement layers of higher frequency information. The 33 progressive layers encoded video supports therefore supports 33 video quality levels. The best quality level is level one, consisting of all 33 layers. The lowest quality level of 33 has the base layer only.

In the multicast data path of the forwarding pane the scaling of packets is performed using gmc. The recipient's details and the various down streamed quality is managed by gmc. Each video packet is managed with a 7bit tag that decided on the quality of the video to be down sliced.

3.4 Adaptation Mechanism

Adaptation mechanism is of prime importance to meet the dynamic conditions of the network in relevance to the quality level of video expected by the learner.

Adaptation should also consider the availability of resources locally apart from the present conditions prevailing in network at any instant of time. When considering a learner in the wireless network for instance, the congestion in network may result in large number of packets being dropped. Thus the quality of video rendered to the user would be severely degraded. The power backup at the learners end would be another major factor that would impact delivery of high video quality for an extended longer period. GMS operates with appropriate mechanisms that provide feedback, using which adaptation could be rolled-out on time as required.

```

1 algorithm_Quality_decrease(datarate) {
2 i=1
3 while (i <= max_lvl) {
4 if (dratetab[i] < datarate) then
5 return i;
6 }
7 i=i+1;
8 }
9 return max_lvl;
10 }

(Assume: c_lvl := Current Level, n_lvl := New Level
after adaptation)
1 algorithm_Quality_adapt(datarate) {
2 diff = datarate - dratetab[c_lvl];
3 Check if (diff < 0) then{
4 Check if (|diff| < threshold * dratetab[c_lvl]) then{
5 Check if (recovr_Dlay++ < maxDlay) then {
6 n_lvl = c_lvl;
7 } else {
8 n_lvl = ideal_lvl;
9 recovr_Dlay = 0;
10 }
11 } else {
12 n_lvl = algorithm_Quality_decrease(datarate);
13 }
14 } else {
15 Check if (recovr_Dlay ++ < maxDlay) then {
16 n_lvl = c_lvl;
17 } else {
18 n_lvl = ideal_lvl;
19 recovr_Dlay = 0;
20 }
21 }
22 return n_lvl;
23 }

```

CROSS/Linux control plane provides the needed support in realizing the above said adaptation with GMS.

In every end-systems, a basic client thread is invoked to monitor the resources of the system in real time. This would provide insight into the power availability, bandwidth, and CPU load. Any critical situation identified as a change in resource requirement way beyond the threshold at the learners side, would prompt a new level of video quality to be demanded from the network elements directly connected. The daemon in the router would appropriately trigger an update in its GMS table based on the demanded new video quality. The request gets propagated to the concerned route through the routers with DVMRP. As example quality levels may be considered proportional to the permissible data rates of different video profiles. This is fetched from a specialized table maintained for access by the clients. This table is used by the end system to loop-up and select the appropriate video quality level as per the data rate requirements. If there is a need for reduction of video quality the following algorithm would prove beneficial for eventual quality degradation.

4. Implementation and Results

Network Simulators provides an inexpensive method for network designers to test new protocols or changes to existing protocols in a controlled environment. Since network simulator ns support the addition of new function research is going on to extend ns to be used in active networks. J-Sim is an application development environment based on ACA. Scripting language used is TCL. It supports object oriented approach; each router function can be implemented as components. JavaSim(J-Sim) is one of the simple network simulator that works on OOPS concepts. In this the components are represented as the base class and the ports as the interfaces. The components of the J-Sim works in a realistic and autonomous ways. Contracts are the main programming components of J-Sim which are independently designed and tested before deployment. The individuality maintained in each component design make the module flexible like a plug and play device, even can be plugged into system during the time of execution. As J-Sim is fully coded in JavaTM with the addition of the autonomous component makes it reusable and platform neutral. The script interface in J-Sim and dual interface are the highlights that helps in programmers, as the scripts can be written using languages like Perl, TCL and even python and can be joined with TCL/Java.

A level-3 hierarchical network is created.net01, net02 and net03 has same structure as that of net00.n0, n1, n2 and n3 are router nodes which run IGMP and DVMRP for multicasting. Addressing scheme used is CIDR. Net10, net11 and net12 are level-2 networks.net00, net01, net02

and net03 are level-1 networks as shown in Figure 4.1.

Figure 4.1: Network scenario created.

Figure 4.2: Simulated Results

5. Conclusion

In multicasting of video to a number of heterogeneous receiver's main problem is the heterogeneity of receivers, since in a multicast group same data is sent in common to the receivers. The solution proposed for this to request a video quality from each receiver and augment this quality level value to the routing table. When the network conditions or CPU capacity is changed the device can request for a change in video quality which will result in the updating of routing table. So the problem we faced in multicasting of video to heterogeneous receivers is solved using this approach. During this COVID pandemic this technique if implemented will reduce the stress the school kids face in poor video stream because of the heterogeneous devices used.

References

- [1] S. Schmid, component-based active router architecture, Ph.D.Thesis, Lancaster University, UK, November 2002
- [2] E. Kohler et al., "The Click Modular Router," ACM Trans. Comp. Sys., vol. 18, no. 3, Aug. 2000, pp. 263–97.
- [3] D. Descaper et al., "Router Plugins: A Software Architecture for Next Generation Routers," Proc. ACM SIGCOMM, Vancouver, Canada, Sept. 1998.
- [4] P. Tullmann, M. Hibler, and J. Lepreau, "Janos: A Java-Oriented OS for Active Networks," IEEE JSAC, Mar. 2001.
- [5] P. Gopalan et al., "Application Performance on the CROSS/Linux Software-Programmable Router," CS TR-01-019, Purdue Univ., Nov. 2001.
- [6] Y. Dong et al. "Composition of Java-Based Router Elements and Its Application to Generalized Video Multicast", IEEE Network Nov/Dec 2004.
- [7] B. Schwartz et al., Smart packets: applying active networks to network management, ACM Transactions on Computer Systems 18 (1) (2000) 67–88.
- [8] R. Keller et al., "An Active Router Architecture for Multicast Video Distribution", Proc. IEEE INFOCOM, Mar. 2000.
- [9] http://uw713doc.sco.com/en/NET_tcpip/iproutingD.mrouted_descrip.html
- [10] <http://technet.microsoft.com/en-us/library/cc781068.aspx>
- [11] <http://www.networksorcery.com/enp/protocol/igmp.htm>
- [12] <http://www.faqs.org/ftp/rfc/pdf/rfc2236.txt.pdf>
- [13] <http://www.networksorcery.com/enp/protocol/pim.htm>
- [14] <http://www.faqs.org/ftp/rfc/pdf/rfc2362.txt.pdf>
- [15] <http://www.javvin.com/protocol/DVMRP.html>
- [16] <http://www.javvin.com/protocol/MBGP.htm>
- [17] http://www.j-sim.org/drcl.inet/inet_tutorial.html
- [18] <http://www.j-sim.org/drcl.inet/inet2.html>
- [19] <http://en.wikipedia.org/wiki/Videoconferencing>
- [20] http://en.wikipedia.org/wiki/Video_on_demand