Enhanced Link based Congestion Control (ELCC) in Peer-to-Peer (P2P) based Video on Demand (VoD) System

Arockia Xavier Annie R
Department of Computer Science and Engineering
Anna University, Chennai, India

Vanitha Sheba M
Department of Computer Science and Engineering
Anna University, Chennai, India

Yogesh P
Department of Information Science and Engineering
Anna University, Chennai, India

ABSTRACT
Video-on-Demand (VoD) services are effectively provided to the users when implemented using Peer-to-Peer (P2P) networks. As each peer node shares its resources, such as bandwidth, storage space, and computing power to their neighbors which is not prevalent in normal networks, the requests from the users are also processed using this property of sharing. The peer nodes upload and download content to and from neighboring peer nodes. This process called peering (or linking), occurs randomly between peer nodes. Every time the demand on the system increases, congestion in the system also increases and finally leads to resource suffocation. The random peering process generally followed, results in very high congestion rate. This paper tries to reintroduce the older pattern of handling congestion could in fact, overcome the user from distorted video due to loss or late arrival of the video packets when combined with new innovative simple idea such that the peers are provided with high level of motivation to serve/supply its buffered video content as well as relieve its duty of supplying to a different peer node which is not infected with congestion. Here, we provide the Enhanced Link based Congestion Control (ELCC) technique for reducing congestion onto the links between the peer nodes. This technique increases the provision of video streaming process to a greater extent. The peer buffers are handled to supply and relinquish data as per the user perceived video data streamed to it. From the results this has proven to be beneficial in general for VoD operations and especially for VCR (Video Cassette Recorder) operations such as fast-forward, pause, replay, backward, random seek etc., in particular.

General Terms
Multimedia communication, Video on Demand Systems, et. al.

Keywords
Peer-to-Peer (P2P), Video-on-Demand (VoD), video streaming, interactivity, Enhanced Link based Congestion Control (ELCC), congestion management, buffering.

1. INTRODUCTION
Video-on-Demand (VoD) services allow users to select and watch the video files on demand. Peer-to-Peer (P2P) network design is very efficient in providing such video on-demand services. P2P network achieves this service since it has the property of resource sharing among peer nodes. Peer nodes make a portion of their resources, such as processing power, network bandwidth, directly available to other participants in the network without the need for central servers. However, P2P streaming in VoD Systems faces many challenges. In VoD systems, the requests to view the video content may come from different users at the same time. VCR (Video Cassette Recorder) operations such as fast-forward, pause, replay, backward, random seek etc., in particular are highly delay sensitive in nature. Congestion occurs in the network due to simultaneous requests that come from different clients. This congestion heavily affects the performance of the network in terms of satisfying the requirements for VCR operations.

Many users may watch the same video content and it is necessary to provide all users with good quality video at the same time with minimum delay [2]. Hence it is necessary that the video streaming within the P2P network and at the user end is effective when congestion is reduced or no congestion is present between peer nodes. This method clearly stands out from the previous congestion control methods for P2P networks as described in [12], which restricts to external link congestion. Here we provide very less overhead incurring solution in redeeming the congested nodes and providing uninterrupted services for the media streams, the VCR request streams as well as the normal request streams.

In general, the peering between the nodes is random. In this paper, we use Enhanced Link based Congestion Control (ELCC) method which includes a technique called Buffering-progress based peering. ELCC helps the P2P network to achieve efficient bandwidth utilization and efficient video streaming. To alleviate resource suffocation problem and to provide the users with high quality videos in a decent interval of time, we motivate the peer nodes to upload more to the P2P network [3], [1]. This encouragement mechanism provided solves more problems such as delay, loss, and jitter etc., encountered during the user view time much efficiently than other existing methods. During VCR requests the download is speeded up with the requester providing upload content from its already filled buffer content. This adds to the congestion, which further is dealt by the ELCC technique that reduces the congestion between the links of the peer nodes where the congestion arises. The novel technique in this system followed for servicing peers is easy to implement and has high adopting ratio due to its convergence with the older Transmission Control Protocol (TCP) friendly congestion techniques involved in a way to a greater extent.

Since the peering process is based on the buffering status of the neighboring peer nodes, multiple requests at the same time is processed using the principle that only the peer nodes which have enough resource to satisfy the request involve in sharing process. These peer nodes are capable in terms of bandwidth, content and user information and act as suppliers to other peer nodes in need of the resources. This decreases the accumulation of unprocessed requests at the incapable peer nodes. To furthermore decrease resource suffocation problem and to provide the users with high quality videos in a decent interval of time, we motivate the peer nodes to upload more to the network by providing them with higher downloading and pre-fetching bandwidth for certain amount of upload.

The paper is organized as follows. In Section 2, we provide a short summary on the existing works done in P2P streaming solutions. The overview on the system design for effective video streams during congestion is established in Section 3 for the P2P networks that are both homogeneous and heterogeneous. The ELCC combined with motivation to upload in peers is elaborated in this section. In Section 4, to achieve continuous video streaming with the defined system, the implementation factors are put up. The performance of the ELCC under realistic network environment is studied through simulations in Section 5. Here, we demonstrate that the effectiveness for streaming video in P2P can increase even at high congestion rate by adapting to Buffering progress based peering that provides minimum delay bounds in highly variable network environments with a small peer upload bandwidth overhead. The paper is ends with conclusion and future work in Section 6.

2. RELATED WORKS

VoD system requires highly flexible streaming network which is supported by the P2P network streams with a higher range of sharing. The client-server model [6], [8] is sufficient for streaming data or video files across the network but it has its own advantages and drawbacks. The client-server model uses the cache and replay technique [6] in which the video file is downloaded completely and stored before the actual playback. This action requires the client to have a huge storage support. The response time is also high in these cases. There are chances for the server to crash when the workload crosses above a critical limit due to multiple requests for various different videos which leads to the destruction of the entire system. This creates scalability issues in such models.

In order to overcome the drawbacks of the client-server model the Peer-to-Peer network model [8] was developed. Earlier P2P model used peer network in the form of a chain [7]. In this model, for processing various requests the peers join the network and are linked using the chain structure. The peer that uploads the data is the parent and the one downloading from it is the child. When a new peer joins the network it is linked as the child to the peer just joined before it. For providing proper management of the resources in a P2P network model this structure works in a better way to identify nodes from the chain to which the newly arriving peers with their requests could be added to the structure as one of the child to an appropriate parent.

Providing high quality VoD [4] using P2P network is feasible using a combination of techniques like network coding, optimized resource allocation across the network and overlay topology management algorithms [8]. There are other P2P network models implemented, one of it is a Peer-to-Peer Multicast (P2Cast) [10] which groups the peer nodes according to the arrival time in the form of a multicast tree. The BitTorrent Assisted Streaming System (BASS) [5] model assumes that there is a streaming server, and all the peer nodes connect to the streaming server. This increases the bandwidth requirements at the server linearly with additional number of users. The Enhanced BitTorrent for supporting Streaming applications (BiTos) [9] is a method used to customize the BitTorrent protocol. This protocol takes in video files which are broken down into pieces. It uses the piece-selection mechanism. Although this mechanism is very efficient in providing peer nodes with rare pieces that are used in the tit-for-tat mechanism to share resource among the peers, the BiTos fails in providing time sensitive traffic. Though BiTos have its issues, this approach has provided an enhancement to the P2P video streaming concept.

Earlier and existing models in common does not take into account the neighbors peers' buffering status, and since the linking between the peer nodes for resource sharing occurs in a random fashion, there is a constant increase in the rate of congestion occurring in the peer network. Furthermore these models ignore the buffer status of the peer leads to inefficient peer bandwidth usage. These issues can be resolved using Buffering progress based peering technique that allows the peer nodes to get connected based on the buffering status of neighboring peers. This technique enhances the system resource utilization because request to server can be stopped at a time when many peer nodes start sharing full information. Motivating the peer nodes to upload more to the network is highly essential.

The early method used is tit-for-tat method [11]. In P2P streaming, this does not work since many peers cannot contribute uploading bandwidth greater than or equal to the playback rate. But here, we implement with the adaptive taxation scheme in which the rate of upload is kept constant and the download to playback ratio is varied. This provides proper taxation on peer nodes. Peers that have high ratio of download rate to that of upload are properly taken care by reducing the supply rate i.e. the download rate is kept at low constant so that these nodes are pursued to upload too. The peers contributing to the network are awarded with high bandwidth for downloading.

The scheme provided by ELCC has given interactivity operation in VoD system with more efficient streaming solution. This is achieved by tagging the nodes as congested and non congested ones and also retrieving the VCR-request related streams from the non-congested nodes. Here, we bring in two concrete plans to instantly relieve the congestion, first of all apart from relieving the congested node the streams at congestion point from the congested nodes are discarded for further transmission and secondly, avoid sending request to the congested nodes for the next consecutive counter time calculated for that peer node. This calculation is done with the number of requests reaching the congested node. Whenever the non congested nodes streaming content to neighbor peers, become congested they are tagged as congested nodes for a time period and streaming during this time is cared by other non congested nodes. This approach greatly reduces the interactive latency incurred and the normal playback rate is maintained which is brought under control through the LCC, combining with buffering progress of the linked peers.

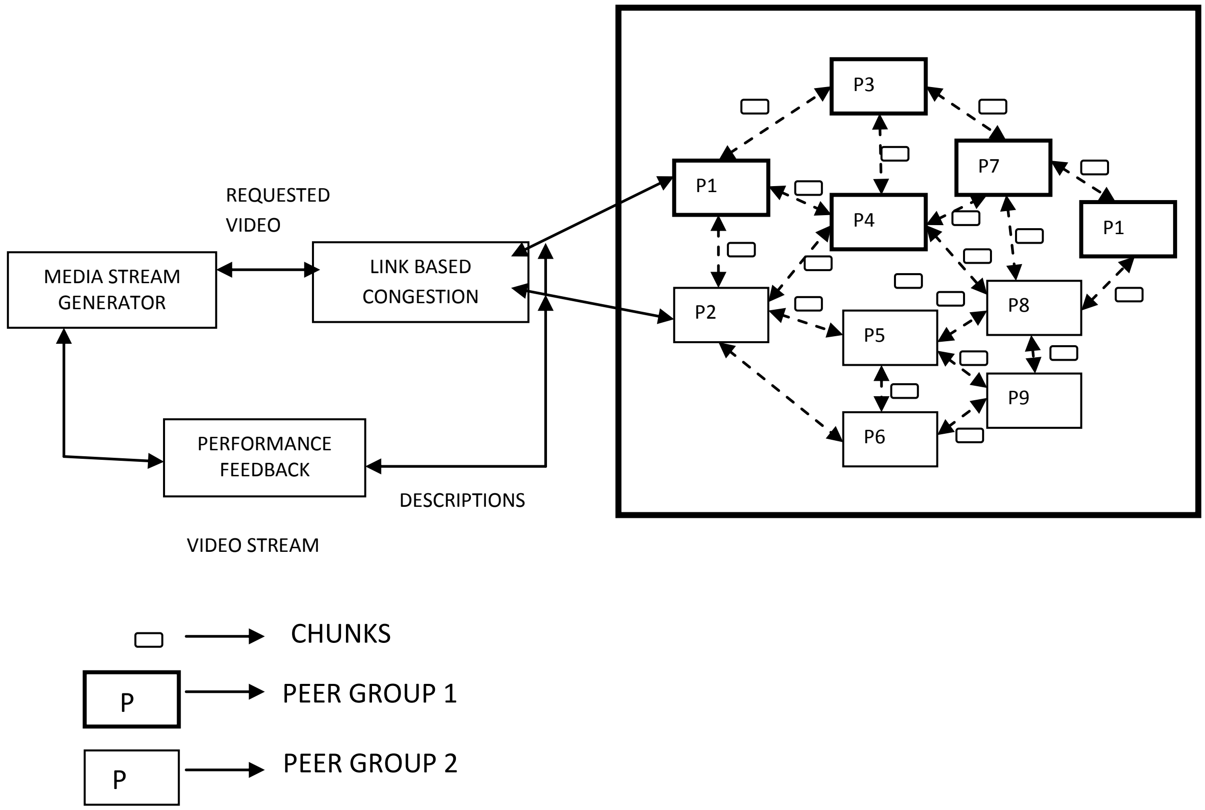

Fig 1: Overall System Design

3. SYSTEM DESIGN OVERVIEW

In this section we present the major design components involved in providing efficiency in streaming. The design and the working process of the modules are given in detail. The system design is as shown in Figure 1. Media streams are generated using the TCP based protocol called pullTCP protocol as it is used for extracting the streams when a request is made. The TCP agent provides the basic TCP connectivity setup for the simulation. This is consumed by the P2P network group for the generated on demand requests. The requests are processed by generating and transmitting the media streams. During the transmission progress, congestion may occur in the network causing loss of media packets, request-response delay, and improper packet transmission. In order to perform congestion control, we apply the ELCC algorithm onto the P2P network. ELCC does congestion detection, recovers the peer nodes from congestion and restores proper transmission of the media packets. As the result of applying ELCC the total number of media packets properly transmitted across the network is highly increased through retrieval operation. A congestion window is also implemented on the links between the networks. This window helps in evaluating the amount of congestion occurring in the network. Hence the variations in the size of the congestion window are also captured after applying ELCC.

3.1 Media Stream Generating Protocol

The video streams needed to generate for the requests are handled by this module. Here we use the media streaming protocol called the pullTCP protocol. Pull-based data exchange is simpler and more robust to peer churns. Each peer just request for the resource it needs from the neighbor. However such a design requires that each peer node knows what resource is available with the neighbor. The pullTCP protocol implements the pull-based data exchange method by generating a sequence of media streams on request from the peer nodes at a particular time. In the pullTcp protocol the receiver controls the full streaming process. The sender passively sends the packets on request. The media streaming protocol is invoked on request from the peer nodes. The protocol uses the congestion window to keep track of the congestion. The Media Streaming Protocol provides the input to the P2P network.

3.2 Peer-to-Peer (P2P) Network

The simulated P2P network as shown in Figure.1 is a combination of multiple networks. Each network is considered as a network consisting of multiple peer nodes. Inter network

communication takes place using the major links between the networks and intra network communication is provided by connecting all the peer nodes to a central tracker node. The links between each network is considered as bottle neck links. Congestion control is applied onto the bottle neck links.
Rather than taking a single network and performing congestion control within, the performance of the congestion control technique can be efficiently evaluated when many network are integrated together. That is the congestion control is being applied across many networks connected together.

3.3 Enhanced Link based Congestion Control (ELCC)

The Enhanced Link based Congestion Control (ELCC) technique consists of two major components, (i) Buffering Progress Based peering (BPB) and (ii) motivating the peer nodes component.
Buffering Progress Based peering: The first process involves the linking between peer nodes. The link is created based on the buffering progress of its neighboring peers. Few of the peer nodes which have higher buffering progress are selected as the suppliers and the other fraction of neighbors with lagging buffering progress are taken as the receivers. The receivers connect themselves to the suppliers and download the required video content. The receivers send requests to the supplying nodes or the senders to get the necessary resource.
In this paper, since the P2P network consists of many sub group underlay networks, each network acts as the sender in a particular time period. The underlay networks are formed when there is video request for the same video but a different segment of that video is requested. This happens during interactive requests wherein the user viewing pattern jumps from the sequential video segments to an incremented or decremented jump in the same video but to a different segment. The load distribution (distribution of the requesting load) is done in a sequential pattern where in each network is set to transmit packets to its neighboring networks. The neighboring network acts as the receiver and the congestion that occurs on the link between the two networks is detected and restored.
Peering process in this type of network model is defined as that one network links itself or exchanges packets only to the neighboring network with the proper resource. The congestion is probed in the link between the two networks and the detection and recovery of congestion is done thereby resuming the proper streaming to the receiver end from a possible different sender which is not termed as congested node.

3.3.1 Motivating peer nodes

The second process is the motivating process where the peer nodes are deemed to upload more to the network. Whenever a peer node uploads more to the network it gets a higher download rate while it starts downloading files from the network. This is managed with a counter to count the number of times the peer has provided/supplied to other peers. The subsequent incremented counter is meant for the server to decide on the greater speed provision in delivering to that particular peer.

3.4 Performance Feedback

This process helps out in altering the various factors affecting each peer sub network links for avoiding congestion. The performance of the system after implementing the Link-based Congestion Control is captured for each link based on the number of packets transmitted properly overcoming congestion and variation in the size of the congestion window as well as the data on congestion in the links. This information helps in appointing a node to either involve in the act of downloading alone for the counter period where it elapses to the total number of requests made to it before congestion occurred. For example, if a node involved in congestion has been identified as a congested node and which was due to the ‘n’ requests that have arrived at it, then it relieves itself from servicing to the next ‘n/2’ requests arriving at it for various video segments stored in its buffer. It also cuts off the streaming services for the previous ‘n/2’ requests that it has served until then at which the congestion might have occurred. Hence, in total the service is cut down for a total of ‘n’ requests at the identified congested node and moreover, further requests would be serviced in quantum time. This method of balancing the act of service at each congested node is unique in its approach which relaxes the congestion point to a great extent. The simulation results generated after implementing this technique is more encouraging and identifies that this is useful in interactive requests of video streaming.

4. IMPLEMENTATION

This section describes about how the P2P network is designed and the congestion control is implemented. NS2 simulator is used for designing the P2P network topology.

4.1 P2P network simulation

The first step of implementation in this paper is designing and simulating a P2P network. As shown in Fig.2, the P2P network consists of five networks each containing ten to fifteen peer nodes. The five networks are connected in a linear fashion. The links between the networks are formed as two simplex links with a bandwidth of 5Mbs and at the packet transfer rate of 20ms. With the two simplex links, one is used for upload and the other is used for download.
A drop-tail queue is integrated onto the links for maintaining the buffering progress of the peer nodes. The drop-tail queue has the ability to bring out global synchronization between nodes in the network. This concept of synchronization is necessary in a P2P network to provide efficient video streaming. Moreover the packets which are timed out due to congestion or those which exceed the threshold limit of the queue are dropped out. Hence, the number of packets lost by congestion can be easily computed. The central node of each network is set as the bottleneck nodes and the link between these nodes form the bottle neck links. The queue limit in the links is set to 100.

4.2 Distribution of request load

Once the P2P network is setup for simulation, the time slot for each network to exchange stream of TCP packets between each other has to be set. The start and stop time of the file transfer application is set. Load distribution is done both randomly and in a sequential manner. In random load distribution the mean peer inter arrival period is set as exponential value and the mean download period is set to be uniformly distributed. The start time and end time is generated randomly.

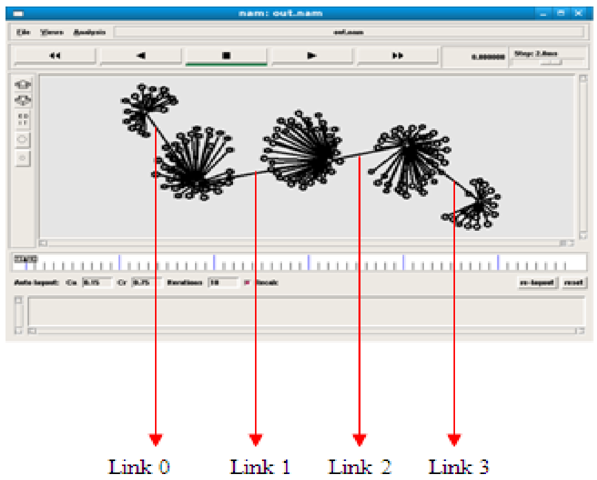

Fig 2: Simulated P2P network

In the sequential method, the network which should start first and then the remaining sequence of network download action is set. So when a network completes its download, the next network in the sequence will start the process.

4.3 Media Stream Generation

After distributing the load, the next step is processing the requests or in other words responding the receiver with the requested resource. The TCP media stream is generated on user request to the peer and is transmitted along the network.

4.3.1 Start and Stop media stream generation

When request for a media stream occurs, the protocol starts to generate the stream by setting the current time and then sending out stream as much as the request needs. The TCP TickTime is set to 0.0 sec based on which the Round to TCP Tick (RTT) time which is the time elapsed between transmission and acknowledgement is updated sequentially.

The generation of the media stream is stopped and restored when congestion occurs, and is completely stopped when the exchange of media streams between every network is completed. The instance of the pullTCP protocol is destroyed on completion.

4.3.2 Time out and Timer Reset

The timeout condition is activated when the sender does not receive the acknowledgement within the preset Round to TCP Tick (RTT) time. The timeout indicates congestion and hence the stream generation is stopped for half the requests that were serviced. The timer needs to reset each time a stream is generated and sent. For each transmission taking place from the sender to the receiver the timer is set to 0 but the counter is incremented.

4.4 Implementing and testing Enhanced Link Based Congestion Control Technique

As the exchange of the media stream starts between the networks, congestion starts to occur in the bottle neck links. The ELCC technique is applied to ease the congestion implemented and tested in this paper.

4.4.1 Congestion Window

A congestion window is implemented onto the links between the peer nodes. The congestion window updating is also done as a part of the pullTCP protocol as shown in ELCC pseudo code. The maximum and minimum size of the congestion window is set. As the streams of packets are generated and transmitted the congestion window gets filled up. The RTT is checked for each packet being transmitted. When the maximum threshold in the window is reached the extra packets are flushed out from on the right side of the window. If for the current request the amount of congestion is high then, this value is set as the new highest maximum of the requests made.

Detection of Congested link and Congested node: The link along which the download process is occurring currently is retrieved based on the node from which the request was last sent and stream transmitted. While downloading, congestion may occur at any node in the network and the congested node is retrieved and displayed. As shown in Figure.3, the links which are presently downloading are denoted with "1" and those that are static at that time are denoted by "0". And those links in which congestion occurs are denoted as "(1)". The congested nodes are appended to a list and once the congestion is recovered then those freed nodes are removed from the list.

```
(1)  0   0   0    53.69
CurrentDownloading_ 2 8 3 4
 0  (1)  1   0    57.55
CurrentDownloading_ 2 8 3 4
 0  (1)  1   0    60.12
CurrentDownloading_ 2 8 3 4
 0   1  (1)  0    62.99
CurrentDownloading_ 2 8 3 4
 0   1  (1)  0    63.14
CurrentDownloading_ 2 8 3 4
 0   1  (1)  0    65.29
CurrentDownloading_ 2 8 4
(1)  1   0   0    65.68
```

Fig 3: Sample Output on LCC

Pseudo code: ELCC

Step 1: Check for the scheduled requests on the links; Initialize the congestion window and the congestion node indicator.

Step 2: For each new request perform the following steps.

a) Set a capable peer node as sender.

b) Set the destination (receivers) list.

c) Start the pullTCP protocol.

Step 3: For each transmission on the link perform the following steps.

a) Monitor the amount of valid transmission.

b) Update the congestion window size.

Step 4: While the size of the congestion window reaches maximum limit perform the following steps.

a) Check if list of congested node is same as in previous transmission identified through the feedback. If same, then remove the links from the congested set and stabilize the node by removing half the streams it serviced identified from the current service to the older one. And further avoid the same number of services for the next incoming requests from all other nodes.

b) Else, if a new node has been appended to the sender's list then reset the value of congestion node indicator and start retransmission (go to step1).

4.4.2 Congestion Indicator and Recovery

Whenever congestion occurs and the media packets get timed out or lost, the congestion recovery process is invoked. The congestion node indicator say 'alpha' is used to recover from congestion at that node. Once congestion occurs, the on-recover process is invoked and the current media transmissions from that node to all other receiving nodes are stopped and the pullTCP is started afresh to retransmit the media packet. When the current congested node changes to a non congested one and congestion occurs in a new node then the congestion indicator is set and based on this change the on-recover function is activated. This function removes the links to the current congested node and the congestion indicator value is reset again as shown in Figure 4.

```
(1) 1  0  0   201.29
CurrentDownloading_ 3
on-recover 0
 1  1 (1) 1   252.58
CurrentDownloading_ 1
on-recover 3
 1 (1) 1  1   252.83
CurrentDownloading_ 1
on-recover 0
 1 (1) 1  1   256.47
CurrentDownloading_ 1
on-recover 3
 0  1 (1) 1   256.72
```

Fig 4: Sample Output for the on-recover process

4.4.3 Delay Probing

Since we have implemented the Buffering Progress Based (BPB) peering with the recovery process the amount of congestion is greatly reduced. In order to test the performance of this technique we manually insert delay slots into the network. The delay is probed in the links that are currently transmitting the media packets.

A delay probe program is created and its instance is used in the timer control program. Maximum and minimum delay slots are calculated and probed periodically into the network. Delay is probed for the packets that get timed out before reaching the receiver. On continuous probing the performance of the recovery process is tested from which the variations in the size of the congestion window and the number of packets transmitted properly overcoming the congestion is retrieved.

5. EVALUATION AND RESULTS

Simulation results are generated after applying the congestion control technique onto the peer network. Performance evaluation is done based on the number of packets transmitted on each link in the network. The graphs are plotted with a scale drawn using number of packets per second. There are four links (link 0, link 1, link 2, link 3) connecting the five networks of peer nodes. Two types of load distribution are applied on the network: random distribution and sequential distribution. The number of packets transmitted in both forms of distribution is retrieved and the comparison is done between them.

5.1 Congestion Detection and Recovery

As we can see in Figure 3, at time 53.69secs, link 0 is downloading from the network, and there occurs an aggregation in the link denoted by "(1)". But the time-out condition does not occur during the entire download operation and therefore the next link starts downloading in a sequential manner. But in the case of Figure 4, congestion occurs in link 0 and the packets gets timed-out, so the recovery process is invoked at 201.29 sec and an effort is made to resume downloading in the link or start a new download.

5.1.1 Simulation Results

Table 1 consists of the number of packets transmitted properly overcoming congestion. For example in the table, Link 0 represents the link between network 1 and 2. The attribute given as "BPB" is the number of packets transmitted on the links after applying Buffering Progress Based peering. The attribute "TCP" gives the number of packets transmitted on applying general TCP concept. Using these values a comparative graph is plotted. Table 2 consists of the time values and the size of the congestion window at that particular time. For example, in Table 2 the time is given in seconds, the congestion window size "CWND size" attribute gives the size of the congestion window implemented onto each link.

Table 1. Sample Packet Transfer Results in Link 0 and Link1.

TS	LINK 0		LINK 1		LINK 2		LINK 3	
	BPB	TCP	BPB	TCP	BPB	TCP	BPB	TCP
1	0	0	0	0	0	0	0	0
2	4.62	4.2	12.16	0	19.66	0	21.51	0
3	10.93	9.6	27.99	0	43.67	0	62.35	0
4	17.67	9.77	34.64	0	50.30	0	67.96	0
5	18.84	10.47	35.49	0	52.79	0	69.94	0
6	13.75	11.50	30.94	0	47.99	0	64.39	0
7	20.04	11.95	36.12	0	51.27	0	66.66	0
8	15.53	12.99	32.85	0	51.90	0	67.90	0
9	15.03	13.83	30.55	0	46.44	0	64.02	0
10	10.00	14.39	28.81	0	47.71	0	66.02	0

Table 2. Sample Congestion Window Size Variations in Link0, Link1, Link2 and Link 3

Domain 0		Domain 1		Domain 2		Domain 3	
Time	CWND size	Time	CWND Size	Time	CWND size	Time	CWND size
1.2	3	1.3	3	1.2	3	1.2	3
1.2	4	1.3	4	1.3	4	1.2	4
1.5	5	1.5	5	1.5	5	1.5	5
1.5	6	1.5	6	1.5	6	1.5	6
1.5	7	1.5	7	1.5	7	1.5	7
1.5	8	1.5	8	1.5	8	1.5	8
1.8	9	1.8	9	1.7	9	1.7	9
1.8	10	1.8	10	1.7	10	1.7	10
1.8	11	1.8	11	1.7	11	1.7	11
1.8	12	1.8	12	1.8	12	1.7	12

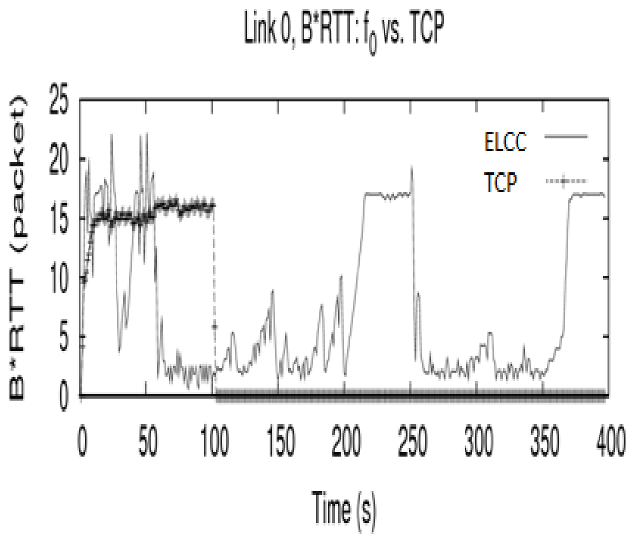

Fig 5: Packet Transfer rate in Link 0

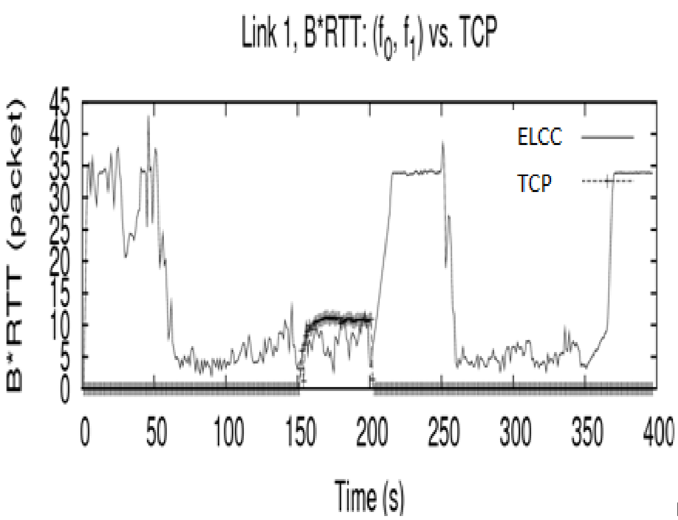

Fig 6: Packet Transfer rate in Link 1

5.2 Graphs Generated

The graphs are plotted based on the number of packets transmitted properly in the network at a particular time slot.

5.2.1 Link 0

The rate of the packets transferred properly across link 0 is shown in Figure 5, which is plotted using the values retrieved on transmission of media packets in link 0. In order to view the variation in the packet transmission in different links, the number of packets actually sent in each link is varied. In link 0, the total number of packets sent is 25, so the x-axis is called from 0-25 and the y-axis is scaled from 0-400. 'f0' denotes the.

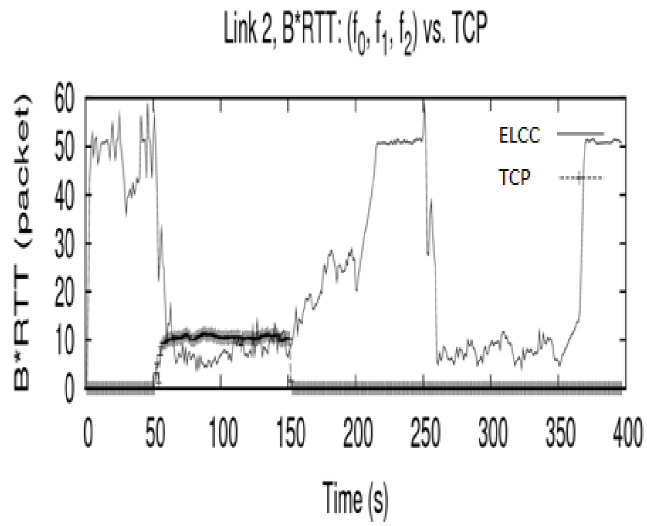

Fig 7: Packet Transfer rate in Link 2

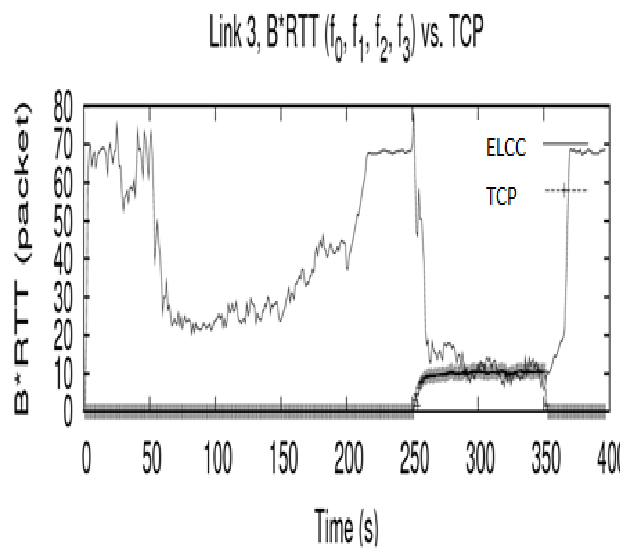

Fig 8: Packet Transfer rate in Link 3

file transfer path in link 0. "f0 vs TCP" denotes the comparison of both ELCC and TCP. We can clearly observe from the graph that, throughout the entire time period there is a minimum level transmission that is within the playback rate of packets across the network after applying BPB peering technique. Wherein, the random TCP transmission completely stops at time 100 second because random TCP does not apply recovery process as the ELCC does.

5.2.2 Link 1

The packet transfer rate across Link 1 is shown in Figure 6. In link 1 the number of packets transmitted across the network is 45 in 400 sec. On applying BPB the transmission starts at 0 sec and retains till 400th sec. But due to congestion in general TCP, the packet are completely lost until 150 sec and it stays only for few 50 sec within which, the playback rate has crossed over by 334 sec. 'f1' denotes the file transfer path for the group of nodes receiving the media content through link 1. In order to denote the path already traversed by the packets through the group of nodes, we represent the previous path as f0 and the present one as f1. Here, comparison of (f0, f1) with respect to general TCP with ELCC is done.

5.2.3 Link 2

The proper packet transfer rate across link 2 is shown in Figure 7. In link 2 the number of packets transmitted across the network is 60 in 400 sec. On applying ELCC the transmission starts at 0 sec and retains till 400th sec. But due to congestion in general TCP, the packet are completely lost until 50sec and it stays only for few 100 sec. again at time 150sec there is zero proper transmission on the network. 'f2'denotes the file transfer path for the group of nodes receiving the media content through link 2. In order to denote the path traversed by the packets through the group of nodes, we represent the previous path as f0 and f1 the present the current one as f2. So, a combined comparison of (f0, f1, f2) with respect to general TCP with ELCC is shown here.

5.2.4 Link 3

The proper packet transfer rate across link 3 is shown in Figure 8. In link 3 the number of packets transmitted across the network is 80 in 400 sec. It is the highest rate of all four links. Hence, the amount of congestion is also higher when compared to other links. On applying ELCC the transmission starts at 0sec and retains till 400th sec. But due to congestion in general TCP, the packet are completely lost until 250sec and the packet are transmitted properly only till 350sec, after which they are lost due to congestion.

In link 3 we do the complete comparison of the entire network. 'f3'denotes the file transfer path for the group of nodes receiving the media content through link 3. In order to denote the path traversed by the packets through the group of nodes, we represent the previous path as f0, f1, f2 and the present current one as f3. So a combined comparison of (f0, f1, f2, f3) vs general TCP is observed.

5.3 Discussion

From the graphs plotted, we can observe that the ELCC packet transfer rate is higher than that of TCP. Even with gradual variations the number of packets transmitted, ELCC values are always almost high than the general TCP. So, we can infer that Enhanced Link Based Congestion Control helps in reducing congestion to a greater extent and thus reducing packet loss, which is highly influential for time centric packet delivery systems such as media streams and in particular interactivity based streams.

6. CONCLUSION

This paper provides the Enhanced Link-Based Congestion Control technique which helps in reducing the congestion rate among the peer links, thereby adding to the efficiency of the video streaming process. The graphs generated clearly states the fact that the speed of video streaming increases after implementing the ELCC technique onto the P2P network. The incentives providing scheme for peer nodes which gives higher download bandwidth to the nodes which upload or share more to the network helps further in increasing the video streaming rate. We observe that LCC greatly enhance the server side load reduction and also provide the users with good quality video. The future work or extension of this paper could be focused on providing users with high quality video for live-streaming environment. Here all peer nodes should be synchronized and should contain almost the same content so that the user side reception is good. Achieving this during congestion could be a mastering effort in live video streaming environment.

7. REFERENCES

[1] Chao Liang Zhenghua Fu, Chai Wah and Yong Liu. 2010. Incentivized Peer-Assisted Streaming for On-Demand Services. IEEE Transactions on Parallel and Distributed Systems. Volume: 21, Issue: 9. doi: 10.1109/TPDS.2009.167. Volume: 21, Issue: 9 Page(s): 1354 – 1367.

[2] Y.Huang, T.Z.J. Fu, D.M. Chiu, J.C.S. Lui, and C. Huang. 2008. Challenges, Design and Analysis of a Large-Scale P2P-VoD System. Proceedings of the ACM SIGCOMM conference on Data communication. ISBN: 978-1-60558-175-0. doi:10.1145/1402958.1403001.

[3] Y. Chu, J. Chuang, and H. Zhang.2004. A Case for Taxation in Peer-to-Peer Streaming Broadcast. PINS '04 Proceedings of the ACM SIGCOMM workshop on Practice and theory of incentives in networked system. ISBN: 978-1-59593-654-7.doi: 10.1145/1016527.1016535.

[4] S.Annapureddy, S.Guha, C.Gkantsidis, D.Gunawardena, and D.Gunawardena. 2007. Is High Quality VoD Feasible Using P2P Swarming? PINS '04 Proceeding of ACM SIGCOMM workshop on Practice and theory of incentives in networked system. ISBN:1-58113-942-X doi: 10.1145/1242572.1242694

[5] C. Dana, D. Li, D. Harrison, and C.-N. Chuah . 2005. BASS: BitTorrent Assisted Streaming System for Video-on-Demand. IEEE 7th Multimedia Signal Processing. doi:10.1109/MMSP.2005.248586. Page: 1 - 4

[6] Bahjat Mohammad Khaleel Qazzaz. 2004. Admission Control and Media Delivery Subsystems for Video on Demand Proxy Server, Doctoral Thesis University of Autonoma of Barcelona (spain). http://www.tdr.cesca.es/TESIS_UAB/AVAILABLE/TDX.../bmkq1de1.pd

[7] Lei Zhang,Kwong–Tung Lo. 2004. A Peer-to-Peer Architecture for On-Demand Video Streaming on Internet. International Conference on Communications, Circuits and Systems. ICCCAS. ISBN: 0-7803-8647-7. doi: 10.1109/ICCCAS.2004.1346174. Volume 1. Page: 525.

[8] Siddhartha Annapur, Christos Gkantsidis, Pablo Rodriguez. 2006. Providing Video-on-Demand Using Peer-to-Peer Network. IPTV Workshop International World Wide Web Conference. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.86.9023&rep...

[9] A.Vlavianos, M. Iliofotou, and M. Faloutsos. 2006. BiToS Enhancing BitTorrent for Supporting Streaming Applications. Proceedings of INFOCOM. 25th IEEE International Conference on Computer Communications. doi:10.1109/INFOCOM.2006.43. Page: 1 – 6.

[10] Y. Guo, K. Suh, J. Kurose, and D. Towsley. 2007. P2Cast: Peer-to-Peer Patching Scheme for VoD Service. Journal on

Multimedia Tools and Appolications. Khuwer Academic Publishers Hingham, USA. Doi: 10.1007/s11042-006-0067-6. Volume.33. Issue 2. Page : 109-129.

[11] Z. Liu, Y. Shen, K.W. Ross, S.S. Panwar, and Y. Wang. 2008. Substream Trading Towards an Open P2P Live Streaming System. Conference on Network Protocols. ISBN: 978-1-4244-2506-8. doi:10.1109/ICNP.2008.4697028. Page:94.

[12] Zhi Chen, JingDong Xu, and GuoWei Huang. 2008. A Dynamic Hybrid Congestion Control Algorithm for Structured P2P Networks. In Proceedings of IEEE International Conference on Multimedia and Ubiquitous Engineering 2008. Page: 221-228.