TCP for Balanced Video Streaming in Wireless Mesh Networks

Rajeswari.G[1] and Srimathi.K [2]

[1] AP (Sr.G), Department of Computer Science and Engineering, Surya Group of Institutions,Vikravandi, Tamilnadu, India

[1]rajilaxman.1980@gmail.com

[2]PG Scholar, Department of Computer Science and Engineering, Surya Group of Institutions,Vikravandi, Tamilnadu, India

[2]srimathikrishnan2706@gmail.com

ABSTRACT

TCP is a basic communication protocol that can be used as a protocol for a private network. Over the years, video has been considered as a important media in the domain of entertainment. At early days, video was sent in analog form. Later it was digitalized. Now the research is about TCP for video transmission which is areliable service. This paper focus on TCP for video streaming.

Keywords — TCP, round trip time, variants, video streaming, balanced flow.

1. INTRODUCTION

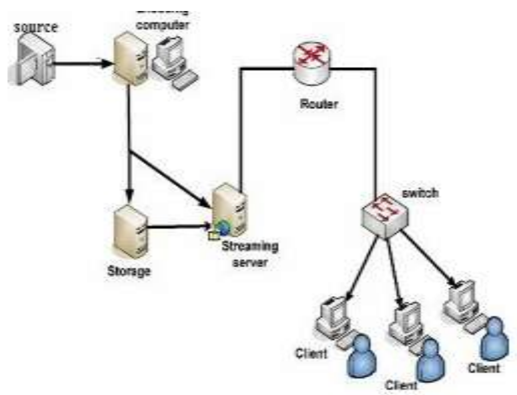

Figure 1. TCP Video Streaming Architecture

Figure 1 shows the architecture of TCP video streaming from single source to multiple client. Video is sent in the form of frames. When the streaming is not live, it can be stored into a storage buffer before transmission. TCP is suitable only for delayed transmission. From the storage buffer the frames are sent to the client. The clients may be of varying bandwidth through routing service. Router is supposed to choose a route through which it can be transmitted. It can choose depending upon the traffic at the adjacent routes.

2. SMART STREAMING OVER TCP PROTOCOL

Streaming means a delivery or transmission of particular data to the client which requested for a data from a authenticated source. Streaming is different from downloading. A client can play the media before it is completely loaded to the buffer storage. By default the video transmission uses TCP. When a video is transmitted, it is fetched by the buffer space. The video plays out from the buffer space as long as the buffer is not empty.

TCP can handle congestion. When the same link is about to suffer from the traffic, TCP can handle. UDP is used when the video is sensitive. Because the delay occurs in TCP at least by milliseconds. Hence TCP is preferred only when the video is not sensitive.

The rate of sending and receiving changes the performance percentage. Moreover the round trip time defines the sending percentage with respect to the receiving client's bandwidth variance.

3. TCP VARIANTS [3]

3.1 TCP TAHOE: A Tahoe refers to the TCP congestion control algorithm which was suggested by Van Jacobson in his paper. TCP is based on a principle of conservation of packets, i.e. if the connection is running at the available bandwidth capacity then a packet is not injected into the network unless a packet is taken out as well. It implements this principle by using the acknowledgements to clock outgoing packets because an acknowledgement means that a packet was taken off the wire by the receiver. It also maintains a congestion window CWD to reflect the network capacity. It suggests that whenever a TCP connection starts or re-starts after a packet loss it should go through a procedure called slow-start. Reason for this procedure is that an initial burst might overwhelm the network and the connection might never get started. The congestion window size is multiplicatively increased that is it becomes double for each transmission until it encounters congestion. Slow start suggests that the sender set the congestion window to 1 end then for each ACK received it increase the CWD by 1. So in the first round trip time (RTT) we send 1 packet, in the second we send 2 and in the third we send 4. Thus we increase exponentially until we lose a packet which is a sign of congestion. When we encounter congestion we decrease our sending rate and we reduce congestion window to one, and start over again. The important thing is that Tahoe detects packet losses by timeouts. Sender is notified that congestion has occurred based on the packet loss.

3.2 TCP Reno:

This RENO retains the basic principle of Tahoe, such as slow starts and the coarse grain retransmit timer. However it adds some intelligence over it so that lost packets are detected earlier and the pipeline is not emptied every time a packet is lost. Reno requires that we receive immediate acknowledgement whenever a segment is received. The logic behind this is that whenever we receive a duplicate acknowledgment, then this duplicate acknowledgment could have been received if the next segment in sequence expected, has been delayed in the network and the segments reached there out of order or else that the packet is lost. If we receive a number of duplicate acknowledgements then it means that sufficient time have passed and even if the segment had taken a longer path, it should have gotten to the receiver by now. There is a very high probability that it was lost. So Reno suggests fast Re-transmit. Whenever we receive 3 duplicate ACK‘s we take it as a sign that the segment was lost, so we re-transmit the segment without waiting for timeout. Thus we manage to re-transmit the segment with the pipe almost full. Another modification that RENO makes is in that after a packet loss, it does not reduce the congestion window to 1. Since this empties the pipe. It enters into an algorithm which we call Fast-Recovery.

3.3 TCP New Reno:

New RENO is a slight modification over TCP-RENO. It is able to detect multiple packet losses and thus is much more efficient that RENO in the event of multiple packet losses. Like R NO, New-RENO also enters into fast retransmit when it receives multiple duplicate packets, however it differs from RENO in that it doesn‘t exit fast recovery until all the data which was out standing at the time it entered fast recovery is acknowledged. The fast recovery phase proceeds as in Reno, however when a fresh ACK is received then there are two cases If it ACK‘s all the segments which were outstanding when we entered fast recovery then it exits fast recovery and sets CWD to threshold value and continues congestion avoidance like Tahoe. If the ACK is a partial ACK then it deduces that the next segment in line was lost and it re-transmits that segment and sets the number of duplicate ACKS received to zero. It exits Fast recovery when all the data in the window is acknowledged.

3.4 TCP Vegas:

Vegas is a TCP implementation which is a modification of Reno. It builds on the fact that proactive measure to encounter congestion is much more efficient than reactive ones. It tried to get around the problem of coa se grain timeouts by suggesting an algorithm which checks for timeouts at a very efficient schedule. Also it overcomes the problem of requiring enough duplicate acknowledgements to detect a packet loss, and it also suggests a modified slow start algorithm which prevents it from congesting the network. It does not depend solely on packet loss as a sign of congestion. It detects congestion before the packet losses occur. However it still retains the other mechanism of Reno and Tahoe, and a packet loss can still be detected by the coarse grain timeout of the other mechanisms fail.

4. TCP FOR BALANCED VIDEO STREAMING

Smart streaming is an orthogonal strategy that tries to use pre-fetching during less busy times to reduce the load at peak hours. Although smart streaming serves as a centralized resource allocation strategy at the server, it can be implemented in a distributed manner as well. Instead of the client making one request for all video segments and the server deciding how and when to send the segments, smart streaming can be implemented based on the existing HTTP streaming protocol - having the client side request for each segment. Based on this information, together with the knowledge of whether the requested segment belongs to the browsing or viewing phase, the server can implement BB. To be more accurate, it would also be helpful for the client side to include the round-trip time (RTT) i the request, so that the server can better take the delay into account.

4.1 Architecture

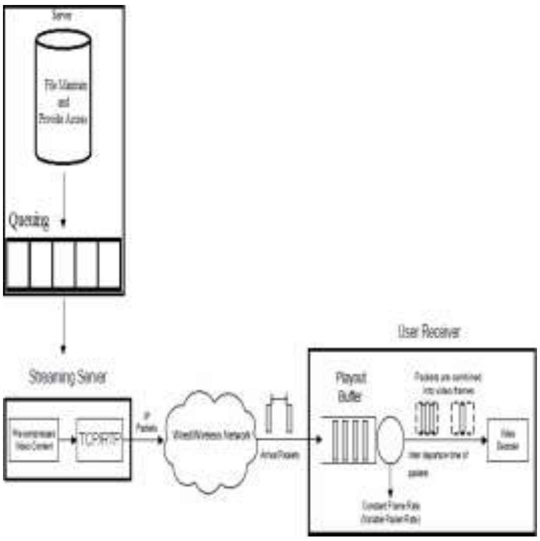

Figure 2. TCP architecture for video streaming

4.2 Modules

- **Buffer state estimation**

Initially the buffer space is estimated by the bandwidth of the client system. Higher the bandwidth, more the number of frames transmitted from the server.as long as the buffer is having frames the video will play with no lag. The client with minimum bandwidth can also receive or load video with no delay with the help of varying round trip time.

- **The client control mechanism**

On receiving the arrived packets; RTT is calculated using the information in data header. It evaluates the sending rate based on the calculated RTT. Detect the buffer changes and calculate the bounds. If the condition matches, send the evaluation rate, and the warning bounds back to the video server.

- **The video server control mechanism**

The video server receives the ACK packets from multi-client. On receiving the ACK it checks the part that was missing from the video. Since the video are sensitive it will result in enormous amount of changes when even a single part of the data is lost and hence it checks. The particular frame that was lost is resent over the network.

- **Smart Streaming**

The overall quality of service can be improved by smart streaming stratergy. The bandwidth wastage is minimized by early departure and excess loading of the video data. The streaming of multi videos with single server application is achieved by switching port number to corresponding system.

5. COMPARISON OF TCP WITH UDP

UDP is unreliable and non-congestion control protocol. UDP cannot handle error correction mechanism. Though the above mechanism have drawbacks, it is not supported with TCP. Forward error correction is necessary when it comes to video streaming and hence TCP can be preferred over UDP.

The advantages of TCP over UDP are as follows:

- TCP provides selective load transmission
- It is adaptable to bandwidth in nature.
- It can be implemented over applications since the firewall uses HTTP.
- Client side buffer, early congestion control and selective acknowledgment is supported with advanced TCP.

6. CONCLUSION

This paper proposed a scheme for video streaming with TCP protocol. Although the TCP protocol uses the required parameters there exist many issues, which are needed to be addressed properly. The research should be extended to video slider and to be tailored to live streaming and minimized delay.

REFERENCES

[1] U.Rahamathunnisa, Dr.R.Saravanan, "A SURVEY ON VIDEO STREAMING OVER MULTIMEDIA NETWORKS USING TCP", Journal Of Theoretical And Applied Information Technology, 20th July 2013. Vol. 53 No.2 , 2005

[2] Tom Jacob Thomas, M. Balasubramani ," Intelligent End-To-End Congestion Control In Wireless Mesh Networks" . IJARCSSE, Volume 4, Issue 4, April 2014

[3] http://Ijcttjournal.Org/Volume4/Issue-8/IJCTT-V4I8P204.Pdf

[4] Ah-Lot Charles Chan, Baochun Li, Edmond Poon, Jilei Liu And Roy Leung (2001),'MP-DSR: A Qos-Aware Multi-Path Dynamic Source Routing Protocol For Wireless Ad-Hoc Networks', Local Computer Networks, 2001. Proceedings. LCN 2001. 26th Annual IEEE Conference, Pp.132-141.

[5] Acharya, P.A.K (2010),' Gateway-Aware Routing For Wireless Mesh Networks', Mobile Adhoc And Sensor Systems (MASS), 2010 IEEE 7th International Conference, San Francisco, CA, Pp. 564 - 569

[6] Adriana Hava, Yacine Ghamri-Doudane (2015),' Increasing User Perceived Quality By Selective Load Balancing Of Video Traffic In Wireless Networks', IEEE Transactions On Broadcasting, Vol. 61, No. 2, Pp.238-250.

[7] A. Hava, G.-M. Muntean, Y. Ghamri-Doudane, And J. Murphy (2013), 'A New Load Balancing Mechanism For Improved Video Delivery Over Wireless Mesh Networks', Proc. IEEE 14th Int. Conf. High Perform. Switch. Routing (HPSR), Taipei, Taiwan, Pp. 136–141.

[8] Hengjun Wang And Na Wang (2010), 'A Security Architecture For Wireless Mesh Network', 2010 International Conference On Challenges In Environmental Science And Computer Engineering,Wuhan, China, Vol.1, Pp.263-266.

[9] Hung Quoc Vo, Choong Seon Hong (2008),' Hop-Count Based Congestion-Aware Multi-Path Routing In Wireless Mesh Network', Information Networking, 2008. ICOIN 2008. International Conference,Busan,Pp.1 - 5

[10] Jyoti Gupta, Paramjeet Kaur Bedi And Nitin Gupta (2011), 'Fault Tolerant Wireless Mesh Network: An Approach', International Journal Of Computer Applications , Vol. 23, No.3, Pp.43-46.

