International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-2 Issue-2, May 2013

Performance Analysis of Adaptive Queuing Techniques for Streaming Real-Time Video

Matilda.S, B.Palaniappan, Thambidurai.T

Abstract— Wireless communication today is a mixture of real time traffic whic is expeced to ocuppy over 70% by 2016 [1]. The challenge lies in integrating the age old wired network with the new-born 4G networks to provide better user experience. Excessive delay is observed in the Base stations and WiMax networks due to difference in available bandwidth between the fixed network and the wireless link. The main reason attributed to the failure in streaming of real-time video, is lack of proper buffer design, which has resulted in bufferbloat across the Internet. This increases the delay across the network eventually leading to packet loss. If QOS is configured correctly in the network, time sensitive packets get priority and playout is smooth. Many techniques such as Active Queue Management and Controlled Delay tend to reduce the effects of bufferbloat. The random drop of packets and static value of queue size adopted in these methods do not support real-time traffic. In this paper Adaptive Controlled delay algorithm has been proposed to overcome the drawbacks of the existing methods and provide a better Quality of Service for real-time video.

Index Terms— Real-time video, Bufferbloat, Controlled Delay, Adaptive Controlled Delay.

I. INTRODUCTION

Video chat is becoming popular as it allows the users to hear the voice of the other person and see them at the same time. Google+ Hangouts VSee, Gmail Video Chat Skype are popular applications which use H.264-SVC. All these applications sounds good on paper but when in use, they result in a very poor experience with lots of stuttery video and massive memory leaks. The quality of video is measured in terms of smooth-motion video, lip-sync maintenance, low bit rate and systems resources requirements. If QOS is configured correctly in the network, time sensitive packets get priority and playout is smooth. This is being done by the ISP and is set only for users in the business class.

One of the reasons attributed to this failure, is lack of proper buffer design, which has resulted in bufferbloat across the Internet. This increases the delay across the network eventually leading to packet loss and reduced throughput. Buffer design is a stochastic, non-linear problem with an integer decision vector. It is a hard combinatorial optimization problem, which is made more difficult by the fact that it is not obtainable in a closed form to interrelate diagonally opposite metrics like delay, packet loss, throughput and link utilization.

There is no single point solution to overcome this problem. Studies conducted on the Internet show that bufferbloat is more pronounced in networks with bandwidth less than 8 Mbps, which includes most of the broadband services.

II. LITERATURE SURVEY

Most prior studies on buffer sizing have focused on TCP traffic performance, and ignore the performance implications for real-time traffic. This is because, it is widely accepted that TCP constitutes nearly 85-90% of the traffic, while real-time (UDP) accounts for about 5-10% [2], [3].

Those in [4] - [6], have proposed the use of adaptive strategies wherein routers adapt their buffer size on-the-fly depending upon the changing network conditions, so as to achieve a desired loss rate and link utilisation. However, these studies rely on the existence of a monotonic relationship between packet loss and buffer size, i.e., that loss rate decreases as buffer size increases.

Dumbbell topology is commonly used to analyse the performance of various congestion control algorithms, including TCP. Such a topology captures packet queuing effects at the bottleneck link, which is the dominant factor in end-to-end loss and delay variation for each flow, while abstracting the rest of the flow path by an access link on either side of the bottleneck link. Buffers have been designed only for the bottleneck link while studies show that this is not sufficient. A bloated buffer could make things even worse.

Studies related to real-time video utilize video traces or online movies. These cannot be categorized as real-time video as they fail to satisfy the basic characteristics shown by real-time video.

Researchers from Georgia Tech [7] and the University of Illinois at Urbana-Champaign [8] have focused on per-flow TCP throughput and shown that the ratio of output-to-input link capacity at the router plays a fundamental role in sizing buffers. If the output link has higher capacity than each input, losses fall exponentially with buffer size and small buffers are sufficient. If the output link has lower capacity than each input, losses follow a power-law reduction and significant buffering is needed. Other studies have considered factors such as application layer performance [9], [10] and fairness [11] influencing buffer sizing. Real-time experiments to determine correlation between buffer sizing and packet loss [12] shows that UDP loss increases as buffers get larger. This anomaly has serious implications, since it suggests that in this region of buffer sizes larger buffers marginally benefit TCP throughput, but have a detrimental impact of UDP loss.

Revised Manuscript Received on 30 May 2013.
* Correspondence Author
Prof.Matilda.S, Department of Information Technology, IFET College of Engineering, Villupuram, India.
Prof. B.Palaniappan, Department of Computer Science, Annamalai University, Chidambaram, India.
Dr. Thambidurai.P, Principal, Perunthalaivar Kamarajar Institute of Engineering & Technology, Karaikal, India.

III. BUFFERBLOAT IN THE INTERNET

Bufferbloat problem is defined as "buffering of packets causing high latency and jitter, which ultimately leads to reduction in network throughput" [13]. The misguided desire to avoid packet loss has led to larger buffers being deployed in the hosts, routers, and switches. Buffers were introduced into cellular networks to absorb the bursty traffic over the variable wireless channel. Link layer retransmission also requires large buffers in the routers and base stations to store the unacknowledged packets. Middleboxes configured for deep packet inspection [14] relies on a large buffer. It is not easy to remove the extra buffers in cellular networks because the operations have to be done by these commercial carriers

A. Existing Solutions to Mitigate Bufferbloat

The first solution offered was Active queue management (AQM), AQM attempts to keep the queues at the bottleneck from growing too large by monitoring the growth of the packet queue. If the queue grows beyond the permitted limit it signals the sender's TCP to slow down by dropping or marking packets [15]. One flavor of AQM is the Random Early Detection (RED), which is based on the fact that, packet loss is not a problem in TCP but is an essential mechanism for functioning in the face of congestion. RED determines that the queue length and randomly selects packets for discard. Dynamic Right Sizing (DRS) is another solution, which allocates just enough buffer so that the throughput of the TCP is never limited by the receive window size but only constrained by network congestion. DRS dynamically adjusts the receive buffer size to suit the connection's demand [16]. DRS increases the receive window size only when it might potentially limit the congestion window growth but never decreases it.

Dynamic receive window adjustment (DRWA) algorithm tends to improve TCP performance over bufferbloated cellular networks. The aim of DRWA is to adaptively set the receive window to a proper size in different environment. DRWA is built on top of DRS. Instead of a unidirectional adjustment where the advertised window is non-decreasing, DRWA implements a bidirectional adjustment algorithm to rein TCP but at the same time ensures full utilization of the link. According to real-world tests performed, DRWA reduces the delay by 25 _ 49% in general cases and increase TCP throughput by up to 51% in some specific scenarios. Since DRWA requires modification at the client side (e.g., smart phones) only and is fully compatible with existing TCP protocol and is immediately deployable

The CoDel mechanism proposed by Kathleen Nichols and Van Jacobson, is designed to provide a "no-knobs" approach to queue management to overcome bufferbloat [17]. They emphasize that bufferbloat occurs mostly in edge routers and defeats the built-in TCP congestion avoidance mechanism, which relies on dropped packets to find the ideal send rate for a given end-to-end link [18].

One of the key insights in the design of CoDel is time taken by a packet to make its way through the queue and reach the destination CoDel defines a maximum acceptable queuing delay, called 'target' and a period over which this delay is measured called 'interval'. If a packet's time in the queue exceeds the target value, then the queue is deemed to be too long. But an overly-long queue is not, in itself, a problem, if empties out at the rate at which it is built. CoDel monitors the time spent by the packets in the queue over each interval and checks if it falls below target at least once. If that does not happen, CoDel starts dropping packets [17]. Dropped packets are a signal to the sender that it needs to slow down. If the queue time remains consistently above target, CoDel will drop progressively more packets. This reduces the incoming rate of the packets and the queue lengths at reasonable values, on a CoDel-managed node. The authors have found that that a target of 5ms and an interval of 100ms work well in any setting [17]. The use of time values instead of packet or byte counts makes the algorithm function independently of the speed of the links.

B. Drawbacks of the Existing Solutions

AQM technique does not support streaming of real-time video. Another disadvantage is that, AQM configuration requires implementation of a patch in all the nodes in the Internet. In practice AQM is not enabled in routers and is unavailable in many devices. Flow control is preserved by DRS but the receive window and the receive buffer size undergoes dynamic adjustments which is unidirectional. Further studies revealed that DRS, is a suboptimal ad-hoc solution to mitigate bufferbloat. It merely mitigates bufferbloat problem only in some scenarios. CoDel compares favorably to the common RED in simulation. RED requires careful tuning of parameters to work well, while CoDel works within a broad range of bandwidths without the need to change any settings. CoDel always kicks in, when standing queues exceed five milliseconds [16]. CoDel has implementation advantages over other AQMs as it does nearly all of its work at the dequeue stage (when packets are transmitted). CoDel requires adding a timestamp to each individual packet as it is received, but even if the network hardware cannot do this, timing information can be directly obtained from the CPU register in modern CPUs. But assuming a uniform target value of 5 ms irrespective of the incoming and outgoing bandwidth does not work well. This also does not fare well with higher number of hops and varying RTT which is typical in video streaming. Codel drops packets at random and has not been tested on real-world data [18]. The probability of losing a TCP packet is nine times higher than a UDP packet as the ration of TCP: UDP is in the order of 9:1. If hybrid transport layer protocols are used for streaming real-time video, this results in the loss of the vital packets carrying I frames. The end result is a poor quality video.

IV. ANALYSIS OF THE EXISTING INTERNET

To overcome all the perceptions and doubts, a number of networks across the globe were studied. Netalysr [19], Pingplotter [20] and Wireshark [21] are the analysis tools. Netalysr returns the summary of the network condition, of which the Network Access Link Properties were used for analysis. It returns the RTT value, Loss, TCP connection setup latency, Network bandwidth on the Uplink and Downlink, Network buffer measurements on the uplink and downlink. Pingplotter was used to identify the bottleneck link, delay jitter and packet loss at each node.

A. Study Methodology

The three tools were run simultaneously to obtain a complete picture of the Internet and the traffic through the network. Online movies were viewed and real-time chat application using google chat was also activated.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

The number of clients in google chat was increased from 1 to a maximum of 9. Other background traffic, like browsing and file download using µ torrent was also included. Results of 137 network paths across the world were obtained using Netalysr and consolidated for study.

B. Observation

The servers are located at Berkely and any client in US could be reached within 16 hops. The RTT value ranged from 126 to 164 ms. All other countries recorded more than 20 hops with RTT of 260 to 340 ms for wired connections and 260 to 700 ms for wireless connections. The TCP connection setup time was close to the value of RTT in most cases. It was observed that network performance was excellent in leased line services and broadband services above 8 Mbps. Bandwidth below 1 Mbps could not support real-time video. The size of the MTU was 1474 bytes. Surprisingly, Wireshark results revealed that only TCP packets of this size traversed the network. The maximum size of UDP packet was 187 bytes. Table 1 shows the sample recording of Network Access Link Properties of 4 networks. 92% of the networks showed symptoms of bufferbloat, while 14% networks recorded a RTT of more than 600ms. Fig. 1 shows the correlation between the bandwidth and end-to-end buffer size in ms observed in the ideal cases.

To determine the correlation between the Bandwidth and buffer size, regression equation was determined. The values obtained are:

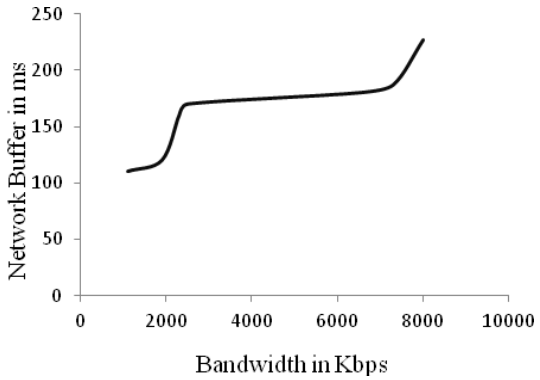

Fig 1. Recording of buffer size for ideal cases

$$\gamma = 0.84066$$
$$\sigma_x = 2559.7852$$
$$\sigma_y = 31.994$$
$$y = 0.0105x + 124.09$$
$$y = 127.5334 \times 1.000026^x$$

The above values depict that there is a definite correlation between the observed values of Bandwidth and buffer size.

TABLE I Network Access Link Properties of Sample Networks

Network Access Link Properties	Mountain View, California, US	BSNL Broadband Pondicherry, India	Airtel Broadband Pondicherry, India	Reliance Leased line, Villupuram, India
Connection Type	Wireless	Wired	Wireless	Wired
RTT	107 ms	260 ms	700ms	270 ms
Loss	0%	2.0%	12%	0%
TCP connection setup latency	101 ms	270 ms	1100 ms	280 ms
Network bandwidth: Upload	1.1 Mbit/s	220 Kbit/s	860 Kbit/s	8.0 Mbit/s
Network bandwidth: Download	240 Kbit/s	2.3 Mbit/s	1.9 Mbit/s	7.3 Mbit/s
Network buffer measurements: Uplink	190 ms	990 ms	750 ms	227 ms
Network buffer measurements: Downlink	3300 ms	160 ms	84 ms	187 ms
FeedBack	Downlink buffering is quite high and real-time applications such as games or audio chat can work quite poorly	Uplink buffering is quite high and real-time applications such as games or audio chat can work quite poorly	Uplink buffering is quite high and real-time applications such as games or audio chat can work quite poorly	This level of buffer may serve well for maximizing speed while minimizing the impact of large transfers on other traffic

V. ADAPTIVE CONTROLLED MECHANISM

The fact that buffer sizing has to vary with the bandwidth supported by the outgoing link is the basis of this mechanism. It does not tend to vary the size of the buffer, but varies the queuing delay each packet experiences in the buffer. Adaptive CoDel (ACoDel) redefines the 'target' and 'interval' to suit the current network parameters. These values are based on the value of RTT, which varies with each segment.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

A. Setting-up of Initial Values

As per economical design of buffer, buffer at any time should hold the number of packets required for immediate consumption by the application. For example, in an application where every tenth packet is counted or if only the tenth packet has the required information, it is sufficient to design a buffer with a capacity to hold 10 packets. For H.264 encoded video stream this value is equal to one GoP. Hybrid Transport Layer protocol approach [22] is used to stream the real-time video. I frames are transmitted using TCP while the less priority Band P frames are transmitted using UDP.

Equation 1 and 2 gives the initial values :

$$\text{Window Size} = 1 \text{ GoP} \quad (1)$$

$$\text{Initial Delay} \cong \frac{2 \times 1GoP}{B_{WeakestLink}} \quad (2)$$

1 GoP is the size of the total frames constituting one GoP of the encoded video stream and $B_{\text{Weakest Link}}$ is the Bandwidth of the weakest link along the path. The initial delay is approximated to the time taken for 1 GoP to arrive at the receiver and acknowledgement to reach the sender provided the bandwidth of the weakest link is greater than or equal to the playback speed of video

B. RTT Estimation

One method of estimating RTT for calculations is to find the average of all the RTT's for those segments which have been acknowledged. This is implemented in estimating the timeout period for the retransmission timer and is given by equations 4 and 5.

$$\text{ARTT(K+1)} = \frac{1}{K+1} \sum_{i=1}^{i=K+1} RTT(i) \quad (3)$$

This expression can be rewritten as

$$\text{ARTT(K+1)} = \frac{K}{K+1} ARTT(K) + \frac{1}{K+1} RTT(K+1) \quad (4)$$

Where RTT(i) is the roundtrip time observed for the i^{th} transmitted segment and ARTT(K) is the average round trip time for the first K segments. Equation 4 simplifies the calculation involved by avoiding the entire summation each time and gives equal weightage to each instance of RTT. For long time transmissions early values of RTT might not reflect the exact picture of the current network environment. Greater weightage to more recent instances is likely to reflect the future behavior much better. RFC 793 suggests predicting the next value on the basis of a time series of the past values using exponential averaging. The expression for Smoothed RTT according to RFC 793 is given by equation 5.

$$\text{SRTT(K+1)} = \alpha \times SRTT(K) + (1-\alpha) \times RTT(K+1) \quad (5)$$

Where $0 < \alpha < 1$. The advantage of using a small value of α is that the average will quickly reflect any rapid changes.

C. Estimation of 'target' and 'interval'

If the buffer size is proportional to the bandwidth of the outgoing link, an optimization can be achieved between delay and packet loss. Challenge lies in determining the proportionality constant. Based on trials the value of 'Interval' and 'Target' are given by the following expressions 6 and 7.

$$\text{Interval} = \text{SRTT} \quad (6)$$

$$\text{Target} = SRTT \times \frac{BW_{out}}{\sum BW} \quad (7)$$

Where, SRTT is the smoothened RTT, BW_{out} is the Bandwidth of the outgoing link and $\sum BW$ is the sum of all the bandwidths in the path.

D. ACodel Algorithm

ACodel can be as two different algorithms being implemented simultaneously to keep the utilization high and delay low. One is time-based and the other is backlog based. When the buffer is full, the backlog based algorithm is implemented. The algorithm then starts dropping the next few packets, till it drops a TCP packets. This results in an ICMP error message, inducing the sender to reduce the window size.
When the delay of a packet exceeds the target value during a specific Interval, only UDP packets are dropped.

```
// Set the initial Values/: win_size,, initial_delay, interval,
target/
/Checking if Queue is full/

   isfull (queue *q)
{
If ((q−>rear is equals to MAXSIZE −1 AND q−>front is
equals to 0) OR (q−>front is equals to q−>rear + 1 AND
q−>front is not equals to 0)) return true;
Else return false;
}

/Time- based algorithm/

If (! isfull (queue))
{
   For Target (0,target)
   If  delay < Interval
   Insert (queue *q, pkt)
{
Update q−>rear by (q−>rear + 1) % MAXSIZE;
Assign pkt  into q−>arr [q−>rear];
}
Else
   If  delay ≥ interval
   Drop  udp pkt
     Print UDP OVERFLOW and return
   }
   /Backlog based algorithm/
   {
   If (isfull (queue))
   drop packet upto next TCP packet
  Print TCP OVERFLOW and return
   }
```

VI. PERFORMANCE EVALUATION

The network from Mountain View, California, US to Berkely was simulated using Opnet Modeler with WiMax in the tail end. Six simulations were run in each case using Opnet, in order to obtain the 95% confidence intervals. Simulation time was 120 seconds.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

WiMax interface was set as a bottleneck link. GoogleChat application served as real-time video traffic and was imported into the network using SITL gateway. The TCP frame size was set to 1500 bytes and UDP was set to 200 bytes to portray a realistic scenario. Background traffic included file download, HTTP, Email and Voice applications. Simulations were run for AQM (RED), CoDel and ACoDel.

The queue sizes estimated by ACoDel are given in Fig.2.

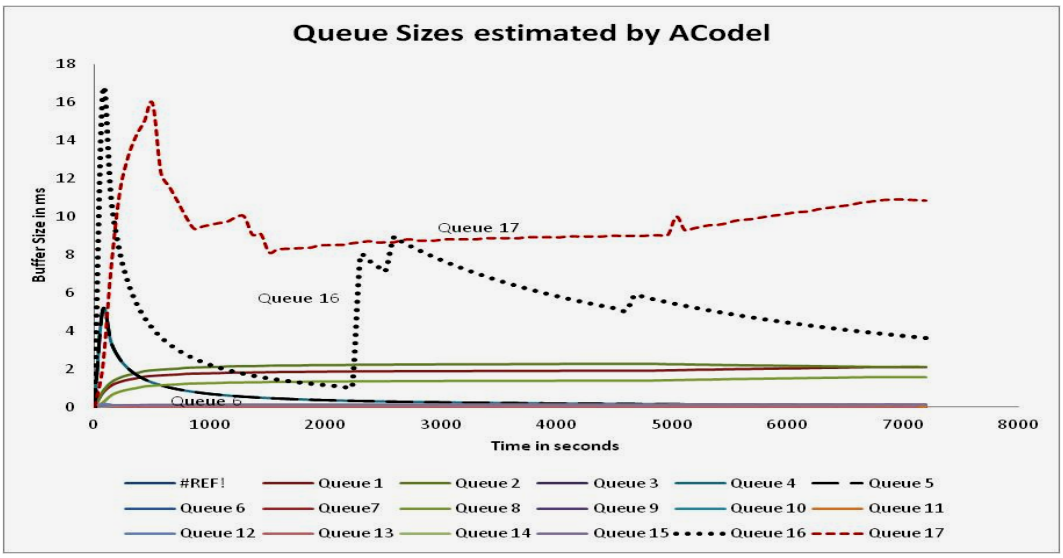

Fig. 2 Queue Sizes Estimated byACoDel

TABLE II. Average Queue Size

Queue No.	Average Queue Size in ms
1	1.83
2	2.11
3	0.8
4	0.42
5	0.42
6	0.23
7	0.76
8	1.48
9	0. 13
10	1.37
11	0.47
12	1.35
13	2.67
14	1.27
15	1.28
16	4.7
17	9.35

The average value of Queue size evaluated by ACodel algorithm is given in Table 2. This is against the constant value of 5 ms adopted by CoDel. The average end-to-end queuing delay is 30.51 ms for 16 hops. This is less compared to the queuing delay of 80 ms estimated theoretically in CoDel. Heavy variations were observed in Queue 6, 16 and 17. The end-to-end delay pattern observed is shown in figure 3.

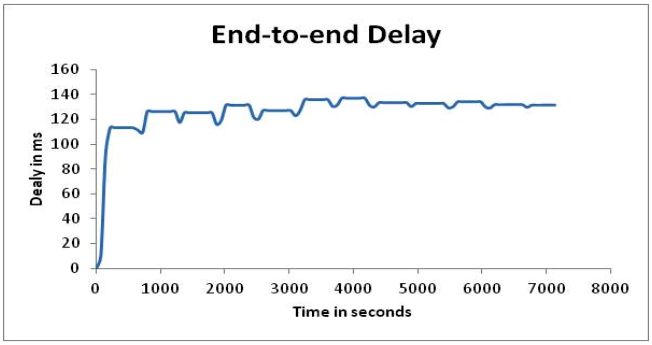

Fig. 3 End-to-end Delay pattern in ACoDel

The average end-to end delay. packet loss and throughput for AQM, CoDel and ACoDel are shown in table 3

TABLE III Comparison of QoS Parameters

Algorithm	End-to-end Delay in ms	Packet Loss in %	Throughput Mbps
AQM	299.9	5.45	1.2
CoDel	155.1	1.86	1.56
ACoDel	129.2	1.36	1.79

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

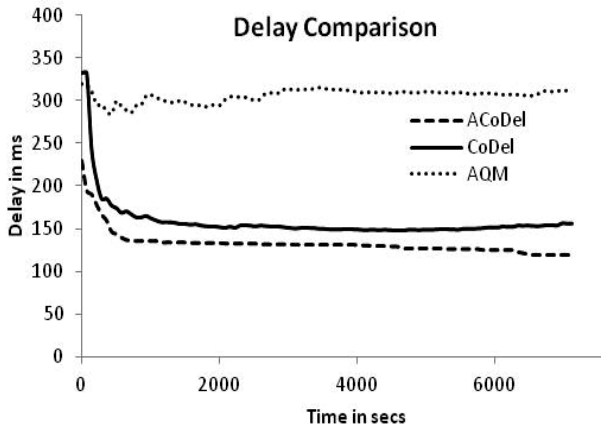

Fig. 4 Comparison of End-to-end Delay

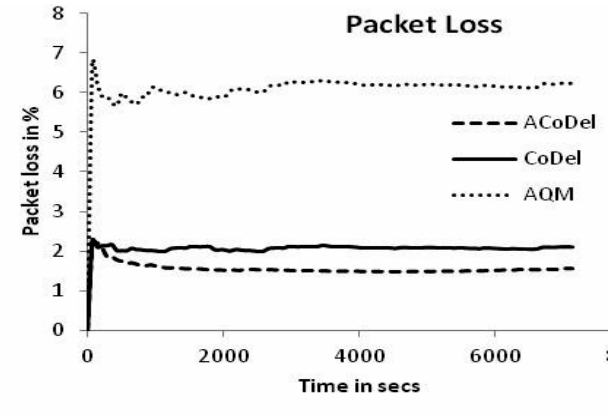

Fig. 5 Comparison of Packet Loss

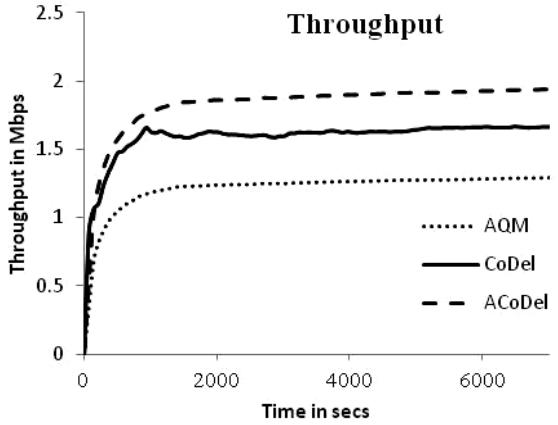

Fig. 6 Comparison of Throughput

VII. CONCLUSION

In this paper, the end-to-end performance of real-time video in a real -time network with ACoDel, CoDel and AQM are compared. ACoDel is based on the concept that less buffer memory keeps queues are kept short. Results prove that ACoDel distributes the buffer along the path based on the capacity of the next outgoing link. It is quite useful in reducing the queuing delays and increasing throughput. Thus, RED, CoDel or most probably any AQM scheme with static parameters is not enough to improve QoS. ACoDel in conjunction with hybrid transport layer protocol approach will provide excellent QoS for real-time video. This will provide a better user experience in the Internet.

REFERENCES

1. Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update- 2011-2016. White Paper, February 2012
2. [2] Y. Zhang and D. Loguinov, "ABS: Adaptive buffer sizing for heterogeneous networks," *Proc. IEEE International Workshop on Quality of Service (IWQoS)*, Enschede, The Netherlands, Jun 2008
3. [3] R. Stanojevic, R. Shorten and C. Kellet, "Adaptive tuning of drop-tail buffers for reducing queueing delays," *IEEE Communications Letters*, vol.10, no. 7, pp. 570-572, Jul 2006
4. Du Li; Qiu Zhen-Yu; Guo Yong-le, "An Improved Queue Management Algorithm in DiffServ Networks," *Information and Computing Science, 2009. ICIC '09.* vol.1, no., pp.123,126.
5. Y. Zhang and D. Loguinov., "ABS: Adaptive buffer sizing for Heterogeneous Networks" *Proceedings of IEEE International Workshop on Quality of Service (IWQoS)*, Enschede, The Netherlands, Jun 2008.
6. R. Stanojevic, R. Shorten and C. Kellet., " Adaptive tuning of drop-tail buffers for reducing Queueing Delays" *IEEE Communications Letters*, vol. 10, no. 7, pp. 570-572, Jul 2006.
7. R.S.Prasad, C. Dovrolis and M. Thottan.,"Router buffer sizing revisited:The role of the output/input capacity ratio,"*Proceedings of the ACM Conference CoNEXT*, New York, USA, Dec 2007.
8. A. Lakshmikantha, R. Srikant and C. Beck, "Impact of file arrivals and departures on buffer sizing in core routers," Proc. IEEE INFOCOM,Phoenix, Arizona, USA, Apr. 2008.
9. A. Dhamdhere and C. Dovrolis, "Open issues in router buffer sizing,"ACM SIGCOMM Computer Communications Review, vol. 36, no. 1, pp.87-92, Jan 2006.
10. G.Vu-Brugier, R. S. Stanojevic, D.J.Leith and R.N.Shorten, "A critique of recently proposed buffer-sizing strategies," ACM SIGCOMM Computer Communications Review, vol. 37, no. 1, pp. 43-47, Jan. 2007.
11. M. Wang and Y. Ganjali, "The effects of fairness in buffer sizing," Proc.IFIP NETWORKING, Atlanta, USA, May 2007.
12. Arun Vishwanath, Vijay Sivaraman and George N. Rouskasz, "Considerations for Sizing Buffers in Optical Packet Switched Networks ," Citeseer, 2009.
13. Jim Gettys, Kathleen Nichols, "Bufferbloat: Dark Buffers in the Internet," in *ACMQueue*, Vol. 9 No. 11 – November 2011, pp. 15–64.
14. Haiqing Jiang, Zeyu Liu, Yaogong Wang, Kyunghan Lee and Injong Rhee, "Understanding Bufferbloat in Cellular Networks," CellNet '12 Proceedings of the 2012 ACM SIGCOMM workshop on Cellular networks: operations, challenges, and future design, pp 1-6.
15. Lakkakorpi, J.; Sayenko, A.; Karhula, J.; Alanen, O.; Moilanen, J., "Active Queue Management for Reducing Downlink Delays in WiMAX," *Vehicular Technology Conference, 2007. VTC-2007 Fall. 2007 IEEE 66th* , vol., no., pp.326,330, Sept. 30 2007-Oct. 3 2007.
16. W.-c. Feng, M. Fisk, M. K. Gardner, and E. Weigle. "Dynamic Right-Sizing: An Automated, Lightweight, and Scalable Technique for Enhancing Grid Performance" Proceedings of the 7th IFIP/IEEE International Workshop on Protocols for High Speed Networks (PIHSN), pages 69–83, 2002.
17. Kathleen Nichols and Van Jackobson, "Controlling Queue Delay" Communications of the ACM, Volume 55, Issue 7, pages 42-50, 2012
18. Richard Chirgwin, "Researchers propose solution to Bufferbloat" http://www.theregister.co.uk/2012/05/09/bufferbloat_and_codel, , 9th May 2012.
19. netalyzr.icsi.berkeley.edu
20. www.pingplotter.com
21. www.wireshark.org
22. Matilda, S., Palaniappan, B., Cross Layered Hybrid Transport Layer Protocol Approach To Enhance Network Utilisation For Video Traffic. ICTACT Journal on Communication Technology. Volume1 Issue 1, March 2010, pp. 54-60.

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)