© 2016 IJSRSET | Volume 2 | Issue 3 | Print ISSN : 2395-1990 | Online ISSN : 2394-4099
Themed Section: Engineering and Technology

Comparative Analysis of CODEL and LEDBAT to Overcome Bufferbloat Problem

Jyoti Chawla, Shailja Kumari
CSA Department, Chaudhary Devi Lal University, Sirsa, Haryana, India

ABSTRACT

Bufferbloat is a problem in a packet-switched network which can occur due to increase in the size of buffer with increase in internet traffic. This creates high latency in network which ultimately degrades the performance of the network. The usage of oversized buffer in the networks results in increasing the queuing delay. Researcher has made two types of suggestions to solve the bufferbloat problem. One is End to End (E2E) congestion control and second is deployment of Active Queue Management (AQM) techniques. This research paper focuses on comparative analysis of Active Queue Management Technique CoDel (Controlled Delay) and Congestion Control Technique LEDBAT (Low Extra Delay Background Transport) for solving bufferbloat problem. This paper compares CoDel with LEDBAT using performance metrics such as end to end delay; queuing delay and queue loss using network simulator-2 (ns-2) and the graphs are drawn using X-graph. The simulation results revealed that LEDBAT has efficiently overcome bufferbloat problem as compared to CODEL by having less end to end delay, less queuing delay and no queue loss.

Keywords : Bufferbloat, Active queue management, Controlled Delay, Low Extra Delay Background Transport, Network Simulator-2

I. INTRODUCTION

Bufferbloat is a problem in a packet-switched network where excessive buffering of packets inside the network causes high latency and jitter, which results in diminishing the overall throughput of the network [1].When a router or switch is configured to use excessively large buffers, even very high-speed networks can become practically unusable for many interactive applications like Voice over IP (VoIP), online gaming, and even web surfing. Buffering is used throughout high performance network systems to handle delays in the system. In general, buffer size will need to be scaled proportionally to the amount of data "in flight" at any time. For very high performance applications that are not sensitive to network delays, it is possible to interpose large end to end buffering delays by putting in intermediate data storage points in an end to end system, and then to use automated and scheduled non-real-time data transfers to get the data to their final endpoints. Some equipment manufactures placed overly large buffers in some of their models. In such equipment, bufferbloat occurs when a network link becomes congested, causing packets to become queued in buffers for too long. In a first-in first-out queuing system, overly large buffers result in longer queues and higher latency, but do not improve network throughput and may even reduce goodput to zero in extreme cases. When the bufferbloat phenomenon is present and the network is under load, even normal web page loads can take many seconds to complete, or simple DNS queries can fail due to timeouts. To solve this bufferbloat problem there are two ways: One is End to End (E2E) congestion control technique such as Low Extra Delay Background Transport and second is deployment of Active Queue Management (AQM) technique such as Controlled Delay.

II. METHODS AND MATERIAL

2. Bufferbloat Solutions

2.1 Congestion control techniques: Congestion Control techniques are those that can either prevent congestion, before it happens, or remove congestion, after it has happened.

2.1.1 LEDBAT (Low Extra Delay Background Transport):

LEDBAT is a way to transfer data on the internet quickly without clogging the network. It was invented by Stanislav Shalunov and is used by Apple for software updates and by BitTorrent for most of its transfers. LEDBAT is estimated to carry 13–20% of Internet traffic. LEDBAT is described in [2] as a windowed protocol, which is governed by a linear controller designed to infer the occurrence of network congestion earlier than TCP. Its congestion control algorithm is based on the estimation of one-way delay: queuing delay is calculated as the difference between the instantaneous delay and a base delay, taken as the minimum delay over the previous observations. Whenever a growing one-way delay is detected by the sender, it infers that queue is building up and reacts by decreasing its sending rate. In this way, LEDBAT reacts earlier than TCP because TCP has to wait for a packet loss event to detect congestion. As TCP congestion control needs losses to back off i.e under a DropTail FIFO queuing discipline, TCP necessarily fills the buffer. As uplink devices of low-capacity home access networks can buffer up to hundreds of milliseconds, this may lead to poor performance of interactive applications e.g., slow Web browsing and bad gaming/VoIP quality [3].

To avoid this drawback, LEDBAT implements a distributed congestion control mechanism whose main design goals are:

1. When no other traffic is present saturate the bottleneck, but quickly yield to TCP and other UDP real-time traffic sharing the same bottleneck queue.
2. When no other traffic is present keep the delay low and add little to the queuing delays induced by TCP traffic.
3. Operate well in DropTail FIFO networks, but use explicit congestion notification (e.g., ECN) where available.

To saturate the bottleneck it is necessary that queue builds up, otherwise, when the queue is empty, at least sometimes no data is being transmitted and the link is underexploited. At the same time, the queuing delay needs to be as low as possible in order to operate friendly toward interactive applications. LEDBAT is therefore designed to introduce a nonzero target queuing delay. To achieve this goal, LEDBAT follows a simple strategy. First of all, it exploits the ongoing data transfer to measure the one-way delay, from which it derives an estimate of the queuing delay on the forward path. Second, it employs a linear controller to modulate the congestion window, and consequently the sending rate, according to the measured delay.

2.2 Active Queue Management Algorithms: It refers to algorithms which are used to control the amount of data stored in network node buffers. AQM algorithms have been widely considered in the recent years to monitor queue sizes and limited congestion in routers.AQM mechanisms were designed to stay away from congestion by proactively informing the TCP sender about congestion such as dropping or marking a packet.

2.2.1 CODEL (Controlled Delay)

In network routing, CoDel (Controlled Delay) is a scheduling algorithm for the network scheduler developed by Van Jacobson and Kathleen Nichols to solve the bufferbloat problem in the internet by limiting the packet queue delay that happened in the network links (routers).CoDel tries to enhance overall performance of the network by reducing the delay and packet loss with high link utilization and throughput. An implementation of CoDel was written by Michael D. Taht and Eric Dumazet for the Linux kernel and dual licensed under the GNU General Public License and the 3-clause BSD license. Some important characteristics of CoDel are [4] :

1. It is parameterless i.e no knobs/handles are required for operators, users, or implementers to adjust.
2. It treats good queue (a queue that exhibits no bufferbloat i.e. it quickly drains as packets are transmitted) and bad queue (a queue that is filled up at the same rate as packets are transmitted, so the queue never empties) differently - that is, it keeps the delays low while permitting bursts of traffic.
3. It controls delay, while insensitive to round-trip delays, link rates, and traffic loads.

CoDel has two important parameters, target and interval. These parameters are needed to be configured wisely to get better performance. CoDel can be considered as delay-based AQM because it uses packet-sojourn time instead of arrival rate or queue length in its congestion indicator [5]. Packet-sojourn time is that time which packet spends in the queue and can be calculated by adding a timestamp to each arrival packet to the queue which contains arrival time for that packet. When the packet is about to leave the queue, the packet-sojourn time can be calculated by subtracting the leaving time with the time that recorded in the timestamp (arrival time) for each packet independently. If the packet sojourn time is larger than a pre-defined target, the algorithm will set timer for dropping packet at dequeueing (leaving the queue). This dropping will happened only when the packet sojourn time is larger than the target and the packets at the queue is less than one MTU's (Maximum Transmission Unit's) of bytes. The time that indicated the next dropping event will be update periodically according to the equation below:

Next_Drop_Time+=Interval/(Sqrt) Count The count represents the total number of dropped packet since the first drop event. Whereas, interval is the minimum value of sliding window that entered the queue and CoDel algorithm has to experience that by time. The dropping action will be stopped when the packet sojourn time goes below the target value.

3. Simulation Setup

We consider the network topology as shown in Fig. 1.The network topology consists of two senders, two receivers and two routers. Duplex link is created between all the nodes. In CODEL,Target 5ms and Interval 100ms is set. Four TCP Agents RENO are used in CODEL whereas four Linux Agents LEDBAT are used with DropTail technique in case of LEDBAT.Four FTPs are used in both CODEL and LEDBAT.

Consider Bandwidth between nodes and routers=15Mbps, Propagation Delay between nodes and routers=15ms, Propagation Delay between routers=25ms, Round Trip Time =100ms and Packetsize=1200 bytes, Buffer Size=600. We consider three cases of Bandwidth between the routers i.e 1.25Mbps,1.5Mbps and 2Mbps to perform the comparison between CODEL and LEDBAT.

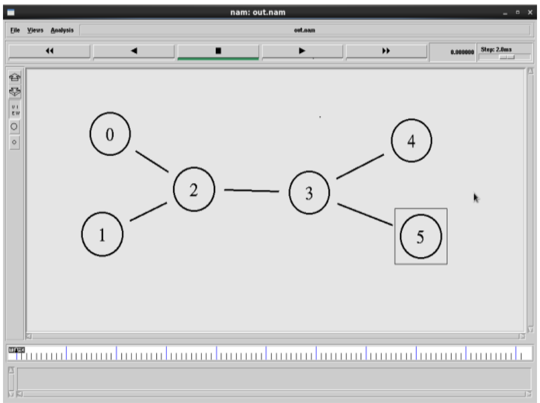

Figure 1. Simulation Topology

4. Performance Metrics

Performance metrics which are used to evaluate the performance of CODEL and LEDBAT are:

1. Queuing Delay: It is the time a job waits in a queue until it can be executed. The maximum queuing delay is proportional to buffer size. The longer the line of packets waiting to be transmitted, the longer the average waiting time is.
2. Queue Loss : It indicates the packets lost from the queue.
3. End-to-end delay: It refers to the time taken for a packet to be transmitted across a network from source to destination.

III. RESULTS AND DISCUSSION

i) Average Queuing Delay Analysis Of Controlled Delay and Low Extra Delay Background Transport on Bandwidth=1.25Mbps,1.5Mbps and 2Mbps

Average Queuing Delay Comparison Between CODEL and LEDBAT		
Bandwidth	**CODEL_EndToEndDelay**	**LEDBAT_EndToEndDelay**
1.25Mbps	0.069549	0.059129
1.5Mbps	0.067051	0.058535
2Mbps	0.064195	0.057793

Table1. Average Queuing Delay Comparison Between CODEL and LEDBAT

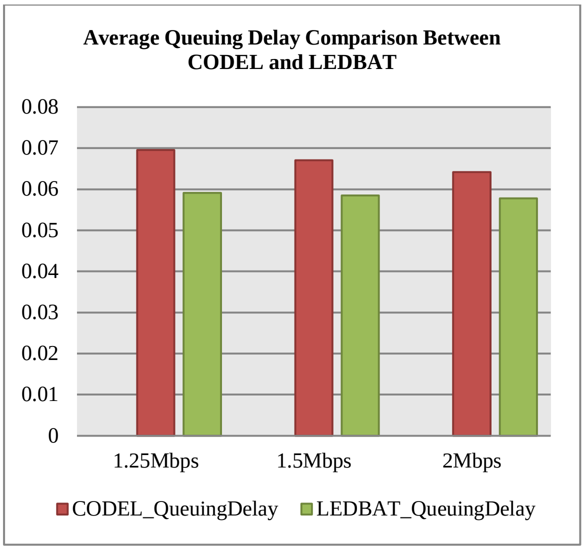

Figure 1. Average Queuing Delay Comparison Between CODEL and LEDBAT

It is observed from Table 1 and Fig.2 that average queuing delay is less in LEDBAT as compared to CODEL.

ii) Queue Loss Analysis of Controlled Delay and Low Extra Delay Background Transport on bandwidth=1.25Mbps, 1.5Mbps and 2Mbps

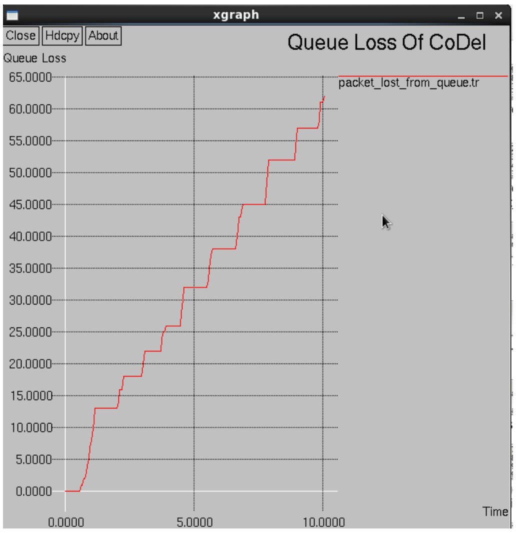

Figure 2. Queue Loss of CODEL on Bandwidth=1.25Mbps

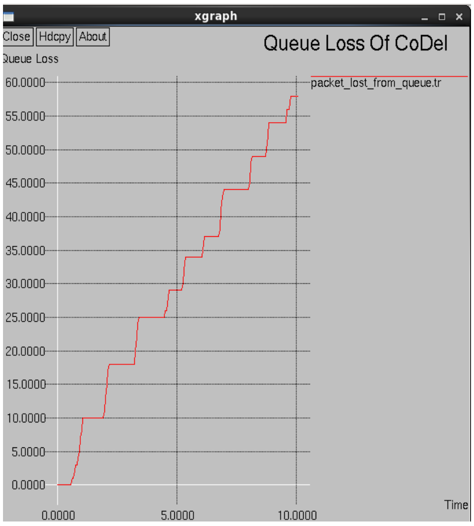

Figure 3. Queue Loss of CODEL on Bandwidth=1.5Mbps

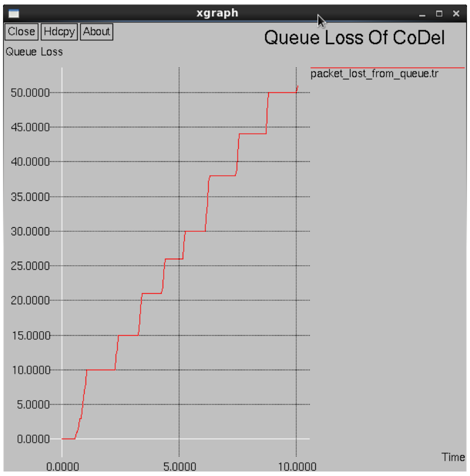

Figure 4. Queue Loss of CODEL on Bandwidth=2Mbps

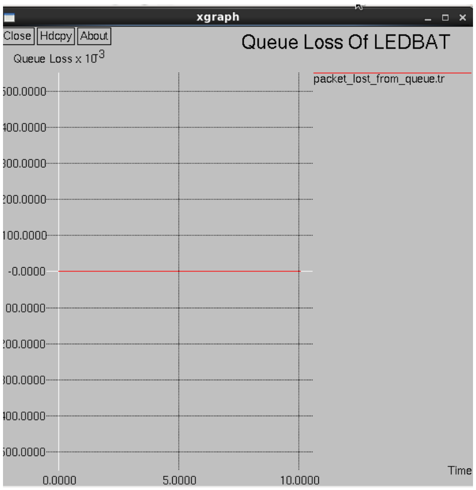

Figure 5. Queue Loss of LEDBAT on Bandwidth=1.25Mbps,1.5Mbps and 2Mbps

After the analysis of all the graphs of queue loss of CODEL and LEDBAT (Fig.3 To Fig.6) on

Bandwidth=1.25Mbps,1.5Mbps and 2Mbps, it is observed that there is no queue loss (i.e no packets are lost from the queue) in LEDBAT whereas there are queue losses at different instant of time in CODEL.

iii) End To End Delay Analysis of Controlled Delay and Low Extra Delay Background Transport on bandwidth=1.25Mbps,1.5Mbps and 2Mbps

Average End To End Delay Comparison Between CODEL and LEDBAT from Node0 To Node 4		
Bandwidth	**CODEL_EndToEndDelay**	**LEDBAT_EndToEndDelay**
1.25Mbps	0.0274994	0.0209188
1.5Mbps	0.0272404	0.0205487
2Mbps	0.0238498	0.0200865
Average End To End Delay Comparison Between CODEL and LEDBAT from Node1 To Node 5		
Bandwidth	**CODEL_EndToEndDelay**	**LEDBAT_EndToEndDelay**
1.25Mbps	0.0286832	0.0209584
1.5Mbps	0.0252424	0.0205813
2Mbps	0.0246837	0.0201107

Table 2 Average End To End Delay Comparison Between CODEL and LEDBAT from Node0 to Node4 and Node1 to Node5

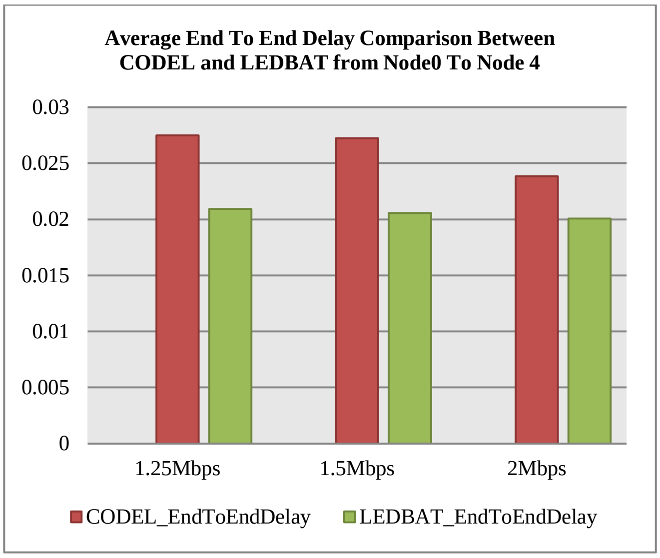

Figure 7. Average End To End Delay Comparison Between CODEL and LEDBAT from Node0 To Node 4

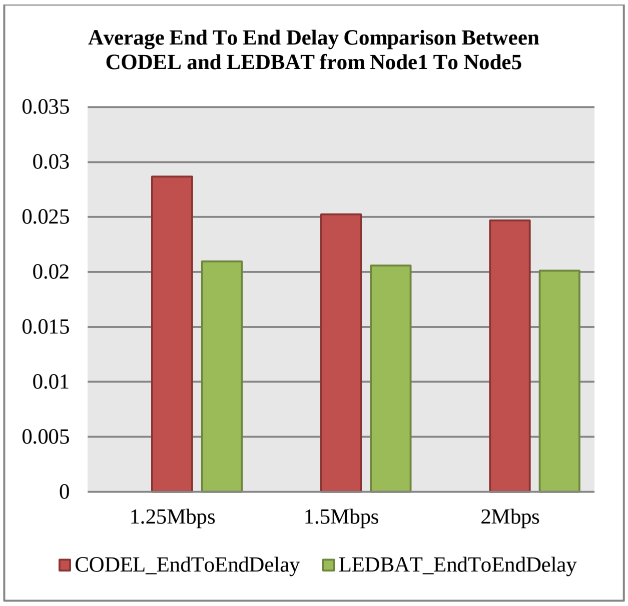

Figure 8 Average End To End Delay Comparison Between CODEL and LEDBAT from Node1 To Node 5

It is observed from Table 2 and Fig.7 and Fig.8 that end to end delay is less in LEDBAT as compared to CODEL.

IV. CONCLUSION

We have compared the performance of Active Queue Management Technique **Controlled Delay** and Congestion Control Technique **Low Extra Delay Background Transport** using performance factors like queuing delay, queue loss and end to end delay with ns2 simulator. After analysis of all the graphs, it is observed that the average queuing delay is less in LEDBAT as compared to CODEL. There are no packets lost from the queue in LEDBAT whereas packets are lost from the queue in CODEL. It is also observed that average end to end delay is less in LEDBAT as compared to CODEL. It is concluded that in terms of reducing average end to end delay,reducing queuing delay and no queue loss, LEDBAT is better as compared to CODEL. Therefore LEDBAT reduces the problem of bufferbloat better than CODEL.

V. REFERENCES

[1] Hashim Ali,Samiullah Khan,and Muhammad Quaid, "Comparative Analysis of Controlled Delay (CoDel) with Deficit Round Robin (DRR) to Overcome Bufferbloat Problem in Wired Network" ,International Journal of Current Engineering and

Technology,E-ISSN 2277 – 4106, P-ISSN 2347 – 5161, Vol.5, No.5 (Oct 2015)

[2] S. Shalunov, "Low extra delay background transport (ledbat)," IETF Draft, Mar. 2009.

[3] Dario Rossi, "LEDBAT: The New BitTorrent Congestion Control Protocol", Computer Communications and Networks (ICCCN),2010 Proceedings of 19th International Conference,Page(s):1-6

[4] Tanvi Sharma, "Controlling Queue Delay (CoDel) to counter the Bufferbloat Problem in Internet", International Journal of Current Engineering and Technology,E-ISSN 2277 – 4106, P-ISSN 2347 – 5161, Vol.4, No.3 (June 2014)

[5] Atheer Flayh Hassan, Suhaidi Hassan, and Suki Arif, "Fairness in Active Queue Management Algorithms:A Review", The 4th International Conference on Internet Applications, Protocols and Services (NETAPPS2015).

[6] Shakeel Ahmad, Adli Mustafa, Bashir Ahmad, Arjamand Bano And ALSammarraieHosam,"Comparative Study Of Congestion Control Techniques In High Speed Networks",International Journal of Computer Science and Information Security,Vol.6, No. 2, 2009

[7] Anita Yadav,Pratap Singh Patwal,"Analyzing the performance of RED, RARED and CoDel in Active Queue Management", International Journal of Advanced Research in Computer Science & Technology, Vol. 2, Issue 4,2014

[8] B. Subramani, Dr. T. Karthikeyan, "A Review on Congestion Control", International Journal of Advanced Research in Computer and Communication Engineering, ISSN (Print) : 2319-5940,ISSN (Online) : 2278-1021,Vol. 3, Issue 1, January 2014

[9] Jens Schwardmann, David Wagner, and Mirja Kuhlewind, "Evaluation of ARED, CoDel and PIE",Institute of Communication Networks and Computer Engineering (IKR) University of Stuttgart, Germany,2014.