ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering

ICACTRP 2017

International Conference on Advances in Computational Techniques and Research Practices

Noida Institute of Engineering & Technology, Greater Noida

Vol. 6, Special Issue 2, February 2017

An Evaluation of Queuing Delay in the Latency of Internet Network Architecture

Rahul Kumar Sharma[1], Pragya Kamal[2], Mayank Deep khare[3], Amrendra Singh Yadav[4]

Assistant Professor, Department of Computer Science & Engineering, NIET, Greater Noida, India[1,3,4]

Assistant Professor, Department of Computer Science & Engineering, BIT, Gorkhpur, U.P. India[2]

Abstract: This paper is the analysis of various factors which influence to the speed and performance of the communication network. Online games, bank transaction etc are required to be minimum time to perform because if it takes a lot of time then the player or user would be irritated and never want to use this service again. So we describe here about latency, which occur due to the queuing delay in database in the network architecture. We give elaborate to queue and queuing delay in the term of buffer, MAC buffering, scheduling, Queue management also.

Keywords: MAC Buffering, FQ, RR, DRR.

I. INTRODUCTION

Delays from packet queuing at devices along the end-toend path, in general, contributes the largest delays to the flight latency. This latency originates from contention for either the switching fabric or the output interface. Provisioning sufficient resources will always reduce (or eliminate) contention, but at the expense of decreased link utilization. Input and/or output buffering is needed to ensure high utilization with bursty network traffic, but inevitably leads to building queues. The overall effect of queuing delay is complex, with buffering often present in each device and each network layer or sub-layer. In this section the term buffer will refer to the resources available to queue packets, and the term queue will refer to the amount of buffer space being used.

Managing queues for quality of service metrics, including latency, was a very active area of research until Dense Wave Division Multiplexing (DWDM) made core network over provisioning a cost-effective approach to support latency sensitive traffic [1]. More recently, latency and network buffering issues have again received attention through the efforts of Gettys, who coined the term buffer bloat to describe the over-abundance of buffering resources in several parts of typical Internet paths. Large queues can induce high latency at any congested point on the end-to-end path.

II. EFFORT OF REDUCE QUEUING DELAY

Currently this is 18 mainly an issue at the edge of the network, but the problem will increasingly affect the core as network access speeds increase. Efforts to reduce queuing delays along the path can be divided into seven approaches:

1) Flow and circuit scheduling,
2) Reducing MAC buffering,
3) Smaller network buffers,
4) Packet scheduling,
5) Traffic shaping and policing,
6) Queue management, and
7) Transport-based queue control.

1) Flow and circuit scheduling

A network device can avoid queuing delays by directly connecting its inputs and output ports (as in e.g. optical switches used to handle the high rates in the core of the Internet or in data centres). There are many types of optical switching [2], but two main categories: Circuit switched (wavelength, fibre or time slot) and connectionless (packet and burst). The former requires the a priori set up of an all optical path from ingress to egress, resulting in less statistical multiplexing. However, after a path has been established, there is no S[IOL] delay and SF has also potentially been reduced. For data travelling along this path, delay will be the speed of light in the fiber times the distance. If such a path is not available, data may have to wait for a path to be created, or may have to be routed via another egress, resulting in a temporary increase in latency and jitter. Since only small optical buffers are currently feasible, designs for optical burst and packet switches have almostno buffering delay. Currently, optical burst switching is the most practical of the connectionless optical switching techniques. In burst switching, packets destined for the same egress are collected in a burst buffer at the ingress and sent in a group. This reduces or removes the need for buffering in the network, but can increase the overall end-to-end latency due to the additional ingress buffering. Optical packet switches are still an area of active research, and developments may help improve latency compared to burst switching because they do not require the extra ingress buffering of optical burst

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering

ICACTRP 2017

International Conference on Advances in Computational Techniques and Research Practices

Noida Institute of Engineering & Technology, Greater Noida

Vol. 6, Special Issue 2, February 2017

switching. Both burst and packet switching may involve tuning wavelength converters, configuring Micro-Electro-Mechanical- System (MEMS) switches, and/or wavelength selective switches. Depending on the architecture, switching delays of the order of 100–300 ns may be achievable resulting in these being no slower than electrical switches.

2) Reducing MAC buffering

Buffering at or below the MAC layer is present for a range of reasons, including: traffic differentiation; header compression; capacity request/medium access, FEC encoding/interleaving, transmission burst formation; handover and ARQ. While systems are typically designed for common use-cases, a large number of independently maintained buffers can add significant amounts of latency in ways that may not be immediately obvious [3]: e.g. when traffic patterns change, the radio resource becomes congested, or components of the system are upgraded revealing buffering in a different part of the network stack.

Network devices have moved from a position where under buffering was common to where MAC buffer bloat can now significantly increase latency, with latencies of many seconds not uncommon in an un-optimized system.

Delays can also result as a side effect of other link protocols. For example, some Ethernet switches implement a type of back pressure flow control using IEEE 802.3X PAUSE frames [4]. If the input queue of a downstream switch is full, a switch can send a PAUSE frame to causes upstream devices to cease transmission for a specified time. This mechanism can avoid loss during transient congestion, but sustained overload can result in a cascading effect that causes the buffers of switches along a path to be filled – dramatically increasing the path latency. Anghel et al. show that use of PAUSE frames in data centres can improve flow completion times, but that care is needed in setting the thresholds in switches and ensuring that there is end-system support. Priority Flow

Control (PFC) is an enhancement that can reduce delay for latency sensitive flows by allowing the PAUSE to specify a particular class of traffic in IEEE 802.1Qbb.

In general, unnecessary buffering below the IP level needs to be eliminated to improve latency. Where possible, packets should be buffered in the IP layer using Active Queue Management (AQM) methods [3]. This can require a redesign of the architecture to enable coordination between protocol entities, avoiding the pitfalls of direct implementation of a large number of independent layers and construction of individually buffered "pipes/streams" across the lower layers.

3) Smaller network buffers

The most effective means of reducing queuing delay is to reduce the size of buffers in each device along the end-to end path, this limits the maximum queue size. An early buffer dimensioning rule-of-thumb [5] recommended that buffers should be sized proportional to the line rate times the RTT

(B = RTT _ C), the Bandwidth Delay Product (BDP), but this is now known to be excessive.

Appenzeller et al. investigated whether BDP sized buffers are required for high utilization in core routers, and showed that core router buffers can take advantage of a high degree of statistical multiplexing and reduce BDP sized buffers by a factor of

p

n, B = RTpT_C

n , where n is the number of concurrent flows on the link.

Further reductions are possible

if the full utilization constraint is relaxed, though Dhamdhere and Dovrolis [6] raised concerns of higher loss rates and thus lower TCP throughput. The work by Appenzeller et al. and an update spawned a number of studies and proposals. Vishwanath et al. surveyed and critiqued much of this work and conducted experiments with mixed TCP and UDP traffic. They concluded that small buffers make all-optical-routers more feasible. As well as reducing latency, Havary-Nassab et al. showed that small buffers make the network more robust against Denial-of-Service (DoS) attacks.

Although work on reducing buffer sizes in the core network is necessary and important for the future, most current congestion is closer to the network edges, where there is not

a high degree of statistical multiplexing. Chandra divides congestion into packet-level and burst-level congestion.

Packet-level congestion only requires small buffers, however congestion due to traffic burstiness and correlations requires much larger buffers. For this reason, small buffers at lightly multiplexed network edges require traffic to be smoothed or paced to avoid burst-level congestion and allow smaller buffers.

Optimizing buffer sizes for various scenarios is still an area of research. A trade-off will remain between latency, utilization and packet loss—with latency expected to become more critical.

4) Packet scheduling

Packet scheduling can also impact latency. A scheduling mechanism allows a network device or end point to decide which buffered packet is sent when multiple packets are queued. Internet hosts and network devices have by default used first-in-first-out (FIFO) scheduling, which sends the oldest queued packet first. This can cause head-of-line blocking when flows share a transmission link, resulting in all flows sharing an increased latency. There are, however, a wide variety of queue scheduling mechanisms and hybrid combinations of mechanisms that can either seek to ensure a fair distribution of capacity between traffic belonging to a traffic class/flow (class/flow isolation), or to prioritize traffic in one class before another. These methods can reduce latency for latency-

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering

ICACTRP 2017

International Conference on Advances in Computational Techniques and Research Practices

Noida Institute of Engineering & Technology, Greater Noida

Vol. 6, Special Issue 2, February 2017

sensitive flows. This section does not seek to explore all scheduling methods, but will highlight some key proposals.

a) Class based: Some scheduling mechanisms rely on classifying traffic into one of a set of traffic classes each associated with a "treatment aggregate". Packets requiring the same treatment can be placed in a common queue (or at least be assigned the same priority). A policy apportions the buffer space between different treatment aggregates and a policy determines the scheduling of queued packets. Different classes of traffic may receive a different quality reflecting their latency and other requirements, so scheduling with this knowledge can have positive impacts on reducing latency for latency-sensitive flows. A router or host-based model implements scheduling in individual routers/hosts without reference to other devices along the network path . This is easy to deploy at any device expected to be a potential bottleneck (e.g., a home router), although it does not itself provide any end-to-end quality of service (QoS).

A more sophisticated model aligns the policies and classes used by the routers across a domain, resulting in two basic network QoS models: The differentiated services model [7] aligns the policies configured in routers using the management plane to ensure consistent treatment of packets marked with a specific Differentiated Services Code Point (DSCP,) in the IP packet header (i.e. devices schedule based only on treatment aggregates). In contrast, the integrated services model uses a protocol to signal the resource requirements for each identified flow along the network path, allowing policies to be set up and modified in real-time to reflect the needs of each flow. Both models can provide the information needed to control the delay of traffic, providing that delay sensitive traffic can be classified. The integrated services model is best suited to controlled environments, e.g. to control latency across an enterprise domain to support telepresence or other latency-sensitive applications.

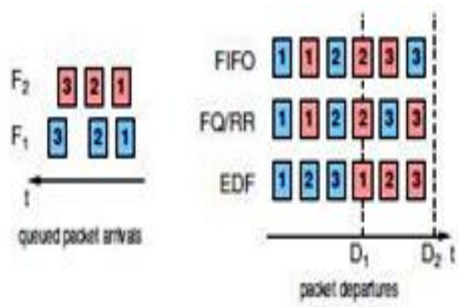

Fig1:-Sharing of available capacity by two flows, illustrating the different between FIFO, FQ/RR and EDF scheduling . Flow F1 has deadline D1 and flow F2 had deadline D2

b) Flow based: Flow based queuing allows a scheduler to lessen the impact that flows have on each other and to discriminate based on flow characteristics. A key example of this is Fair Queuing (FQ), proposed by Nagle and Demers et al., which aims to improve fairness among competing flows. This ensures that queuing delays introduced by one (possibly misbehaving) flow do not adversely affect other flows, achieving fairness. FQ has been adapted and extended in many different ways including: weighted FQ, and practical approximations such as Round Robin (RR) scheduling and Deficit RR (DRR) . In practical implementations, there may be a limit to the number of queues that can be implemented, hence stochastic fair queuing (SFQ McKenney [8]) and similar methods have been proposed to eliminate the need for a separate queue for each traffic flow.

Per-flow classification requires a flow classifier to discriminate packets based on the flows to which they belong and deduce their required treatment. In the router or host-based model this function is performed at each router and requires visibility of transport protocol headers (e.g. protocol and port numbers), whereas in the Differentiated or Integrated models visibility is required at the edge of the QoS domain. When tunnels are used, all tunnel traffic is generally classified as a single flow (an exception could be the use of the IPv6 Flow Label to identify sub flows). This can add latency by assigning all traffic that uses a VPN tunnel to the same queue, besides the obvious processing cost of encryption and decryption.

c) Latency specific: Some schedulers schedule packets to achieve a low or defined latency. The simplest is Last-In-First-Out (LIFO), which minimizes the delay of most packets—new packets are sent with a minimum delay at the expense of packets already queued. Unfortunately, this method also maximizes the delay variance and reorders packets within a flow.

Deadline-based schemes attempt to bound the latency of a queue, e.g. Earliest Deadline First, where jobs (or packets) are scheduled in the order of their deadline. for two flows with different deadlines; both flows can meet their deadlines if the flow with the earliest deadline is scheduled first. Unfortunately, these methods fail to provide good performance under overload.

Shortest Queue First (SQF) is a flow/class based scheduler that serves packets from the flow/class with the smallest queue first. It has been proposed for reducing latency in home access gateways. Carofiglio and Muscariello [9] show that the SQF discipline has desirable properties for flows that send less than their fair share, such as thin latency-sensitive flows and short flows, at the expense of bulk throughput-sensitive flows.

d) Hierarchical scheduling: In many networks it is normal to create a hierarchy of scheduler treatments, in which some classes of traffic are given preferential or worse treatment by the scheduler, to achieve different treatments for the traffic. For example the Expedited Forwarding (RFC 3246) differentiated services class assigns a treatment that offers low loss and low latency. Class-based

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering

ICACTRP 2017

International Conference on Advances in Computational Techniques and Research Practices

Noida Institute of Engineering & Technology, Greater Noida

Vol. 6, Special Issue 2, February 2017

queuing (Floyd and Jacobson [10]), hierarchical packet fair queuing [11] and 802.11e QoS enhancements for WLANs provide such methods. Any priority-based algorithm needs to correctly classify the traffic in a way that guarantees the required treatment. This typically requires a policing (or traffic-conditioning) function to prevent misuse. In the integrated and differentiated services model this conditioning may be provided at the domain edge.

5) Traffic shaping and policing

Traffic shapers smooth traffic passing through them using a buffer to limit peak transmission rates and the length of time these peak rates can be maintained. While shaping can help prevent congestion—and therefore delay further along the path—, it does so at the expense of delay introduced at the shaper. The foundational traffic shaping algorithms are the leaky bucket algorithm and the related token bucket algorithm. Traffic shapers are used extensively in the Internet, though often to reduce ISP costs rather than to reduce delays along the path.

Traffic policers drop packets that exceed a specified peak transmission rate, peak burst transmission time, and/or average transmission rate. Policing was first proposed by Guillemin et al. [12] for ATM networks, but is still an effective tool for managing QoS, especially latency, in the Internet. Briscoe et al. [13] propose a policing mechanism that could be used either to police the sending rate of individual sources (e.g. TCP) or, more significantly, to ensure that all the sources behind the traffic entering a network react sufficiently to congestion, as a combined effect, without constraining any flow individually. This developed into the IETF Congestion Exposure work to enable a number of mechanisms, especially congestion-based policing, to discourage and remove heavy sources of congestion and the latency they cause.

6) Queue management

Network devices can monitor the size of queues and take appropriate action as the queue latency builds; this is known as queue management. Techniques such as drop tail and drop front are said to be passive. In contrast, Active Queue Management (AQM) techniques manage queues to achieve certain queue loss and latency characteristics by proactively marking or dropping packets; which signal to endpoints to change their transmission rate. AQM mechanisms generally work in combination with scheduling, traffic shaping, and transport-layer congestion control. Adams [14] provides an extensive survey of techniques from Random Early Detection (RED Floyd and Jacobson [15]), introduced in 1993, through to the year 2011. This section looks at more recent contributions, with a specific focus on latency.

a) PIE and CoDel: Two current proposals aim to minimize the average queuing delay: Proportional Integral controller Enhanced (PIE Pan et al. [16]) and Controlled Delay (CoDel Nichols and Jacobson [17]), and more recently, a per-flow queuing version of CoDel called FQ-CoDel.

The PIE algorithm uses a classic Proportional Integral controller to manage a queue so that the average queuing delay is kept close to a configurable target delay, with a current default value of 20 ms. PIE does this by using an estimate of the current queuing delay to adjust the random ingress packet drop or marking probability. The algorithm selftunes its parameters to adapt quickly to changes in traffic. PIE tolerates bursts of packets up to a configurable maximum, with a current default value of 100 ms. CoDel attempts to distinguish between two types of queues, which the authors refer to as good queues and bad queues— that is, queues that simply buffer bursty arrivals, and those just creating excess delay. Although the default target delay is 5 ms, it allows temporary buffering of bursts which can induce delays orders of magnitude larger than the target delay. Packets are dropped or marked at deterministic intervals at the head of a queue. Dropping/marking at the queue head decreases the time for the transport protocol to detect congestion. Combining this with the flow isolation of a fair queuing scheduler avoids packet drops for lower-rate flows.

Both schemes attempt to keep configuration parameters to a minimum, auto tune, and control average queue latency to approach a target value. Both exhibit high latency during transient congestion episodes. Use of smaller buffers would prevent this, but this is an area that requires further research.

b) DCTCP and HULL: Data Centre TCP (DCTCP Alizadeh et al. [18]) is illustrated in Fig. 2. It uses an AQM method that has been designed to keep queuing delay and delay variance very low, even for small numbers of flows including a single flow. The method appears deceptively simple; it merely marks the ECN field of all packets whenever the queue exceeds a short threshold.

The AQM for DCTCP signals even brief excursions of the queue, in contrast to other AQMs that hold back from signalling until the queue has persisted for some time. Even though existing switches often only implement RED, they can avoid introducing signalling delay by simply setting their smoothing parameter to zero. High bandwidth Ultra-Low Latency replaces the AQM algorithm in DCTCP. It aims to keep the real queue extremely short by signalling ECN when a virtual queue exceeds a threshold. A virtual queue is a token bucket- like counter that fills at the real packet arrival rate, but drains slightly more slowly than the real line. A growing range of commercial equipment natively supports virtual queues, often using two hardware leaky buckets.

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering

ICACTRP 2017

International Conference on Advances in Computational Techniques and Research Practices

Noida Institute of Engineering & Technology, Greater Noida

Vol. 6, Special Issue 2, February 2017

7) Transport-based queue control

A number of transport layer mechanisms have been proposed to support low queuing delays along the end-to-end

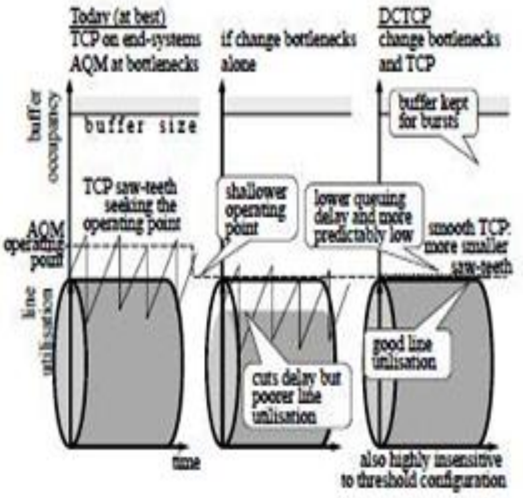

Fig 2:- How Data Centre TCP (DCTCP) reduces delay without lowing utilization

path. Two key elements are: burstiness reduction and early detection of congestion.

a) Coupled congestion control: When multiple flows that originate from the same endpoint traverse a common bottleneck, they compete for network capacity, causing more queue growth than a single flow would. Detecting which flows share a common bottleneck and coupling their congestion control can significantly reduce latency, as shown with an SCTP-based prototype. Solutions in this space are planned outcomes of the IETF RMCAT working group.

b) Burstiness reduction: A TCP session that always has data to send typically results in paced transmission, since the sender effectively paces the rate at which new data segments may be sent at, to the rate at which it receives ACK packets for old segments that have left the network. However, this is not always the case. TCP's window-based congestion control together with bottleneck queuing can result in very bursty traffic. In some implementations, the TCP max burst function limits the maximum burst size per received ACK, hence reducing burstiness for bulk applications.

However, not all applications continuously have data to transmit (e.g. when a server responds to requests for specific data chunks, or when a variable rate video session experiences a scene change). There may therefore be periods in which no TCP data are sent, and hence no ACKs are received—or an application may use an entirely different transport that does not generate an ACK for every few segments. Either of these can result in bursts of packets, and may require explicit pacing at the sender or a traffic shaper within the network.

III. CONCLUSION

After evaluating all conditions with problems, we concluded that if we focus on queue management with MAC buffering and packet scheduling then we could reduce large phase of our queuing delay problem.

REFERENCES

[1] C. Fraleigh, F. Tobagi, and C. Diot, "Provisioning IP backbone networks to support latency sensitive traffic," in Proc. of the IEEE International Conference on Computer Communications (INFOCOM), vol. 1, 2003, pp. 375–385.

[2] M. Maier and M. Reisslein, "Trends in optical switching techniques: a short survey," IEEE Netw., vol. 22, no. 6, pp. 42–47, Nov. 2008.

[3] C. Staff, "Bufferbloat: what's wrong with the internet?" Commun. ACM, vol. 55, no. 2, pp. 40–47, Feb. 2012.

[4] "Specification for 802.3 full duplex operation and physical layer specification for 100 Mb/s operation on two pairs of category 3 or better balanced twisted paircable (100BASE-T2)," IEEE, Std. 802.3X, 1997.

[5] R. Bush and D. Meyer, Some Internet Architectural Guidelines and Philosophy, RFC 3439 (Informational), Internet Engineering Task Force, Dec. 2002.

[6] A. Dhamdhere and C. Dovrolis, "Openissues in router buffer sizing," ACM SIGCOMM Computer Communications Review (CCR), vol. 36, no. 1, pp. 87–92, Jan. 2006.

[7] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, An Architecture for Differentiated Services, RFC 2475 (Informational), Updated by RFC 3260, Internet Engineering Task Force, Dec. 1998.

[8] P. McKenney, "Stochastic fairness queueing," in Proc. of the IEEE International Conference on Computer Communications (INFOCOM), vol. 2, 1990, pp. 733–740.

[9] G. Carofiglio and L. Muscariello, "On the impact of TCP and per-flow scheduling on Internet performance," IEEE/ACM Trans. Netw., vol. 20, no. 2,pp. 620–633, Apr. 2012.

[10] S. Floyd and V. Jacobson, "Link-sharing and resource management models for packet networks," IEEE/ACM Trans. Netw., vol. 3, no. 4, pp. 365–386, Aug. 1995.

[11] J. C. R. Bennett and H. Zhang, "Hierarchical packet fair queueing algorithms," IEEE/ACM Trans. Netw., vol. 5, no. 5, pp. 675–689, Oct. 1997.

[12] F. Guillemin, P. Boyer, A. Dupuis, and L. Romoeuf, "Peak rate enforcement in ATM networks," in Proc. of the IEEE International Conference on Computer Communications (INFOCOM), 1992, pp. 753–758.

[13] B. Briscoe, A. Jacquet, C. Di Cairano-Gilfedder, A. Salvatori, A. Soppera, and M. Koyabe, "Policing congestion response in an internetwork using refeedback," ACM SIGCOMM Computer Communications Review (CCR), vol. 35, no. 4, pp. 277–288, Aug. 2005.

[14] R. Adams, "Active queue management: a survey," IEEE Commun. Surveys Tuts., vol. 15, pp. 1425–1476, 2013.

[15] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," IEEE/ACM Trans. Netw., vol. 1, no. 4, pp. 397–413, Aug. 1993.

[16] R. Pan, P. Natarajan, C. Piglione, M. Prabhu, V. Subramanian, F. Baker, and B. VerSteeg, "PIE: a lightweight control scheme to address the bufferbloat problem," in Proc. of the IEEE International Conference on High Performance Switching and Routing (HPSR), Jul.2013.

[17] K. Nichols and V. Jacobson, "Controlling queue delay," ACM Queue, vol. 10, no. 5, May 2012.