LAWIN: a Latency AWare InterNet architecture: an alternative reducing a latency

Katsushi Kobayashi
University of Tokyo
ikob@acm.org

ABSTRACT

This position paper presents our proposal of LAWIN (Latency AWare Internet) architecture instead of existing reducing Internet latency approaches. [1]

1. INTRODUCTION

A large end-to-end latency reaches up to 10 seconds has been raised as one of the most important issue, after the bufferbloat discussion[6]. The buffer of access equipment becomes larger, and it mismatched with its link capacity. The large buffer is occupied by long-lived data flows. As the result, the user faces the worst quality of experiments on interactive application as online-game.

Couple types of approaches are proposed to reduce the latency caused by the bufferbloat. The first approach is to reduce the buffer size of access circuit equipment, such as the latest DOCSIS recommendation[1]. The second is AQM, such as Codel and PIE, that keeps the queueing latency within target value [11, 12] The both approaches cannot support different latency requirements, although they can be deployed with existing packet format. In addition, All AQM proposals expected that end systems responds in better by dropping and marking. All network engineers already learned such approach has never worked, because the cheating application pretends as good manner. Thus, the network should not tussle over how much to allocate or confiscate the resource with applications. Instead, network should provide a space that applications can deal each other, either implicit or explicit ways.

In this paper we propose Latency AWare InterNet (LAWIN) that can support different latency requirements. LAWIN provides the framework that applications can trade off the latency and the bandwidth. In the rest part of this paper, we will describe LAWIN architecture, implementation, and its summary.

[1]We are afraid that LAWIN proposal is beyond the scope of the ISOC workshop, because of little far from the industry. However, we believe our idea will contribute the workshop as a input from academic community. We also apologize this paper exceeds three pages, one more than the limit, due to reference section.

2. LAWIN ARCHITECTURE

Before describing the LAWIN architecture, we first review large latency studies on the datacenter network. Emerging parallel applications in DC, such as map-reduce, has been imposed tighter deadline usually less than 1 msec. The long-lived traffic of other application also shares the same DC networks, which occurs queue build up[2]. The result of longer filled queue degrades the parallel application unable to meet the deadline. Although the latency requirements are quite different, this is the similar as buffer bloat we mentioned the above. EDF scheduling with explicit per-packet deadline demonstrated to improve job completion time[3]. Note that we refer EDF scheduling with per-packet deadline as "latency aware" in this paper.

It is a big challenge to apply the "latency awareness" to the Internet, because the single operation policy cannot be imposed to all Internet operators and users. If highest priority is given to the closest deadline packet, greedy application specifies an earlier deadline to own packets. Therefore, we add reneging onto "latency aware" DC network using simple EDF schedular[9]. On reneging version EDF scheduling, customers leave the service when their deadline are elapsed. Intuitionally, this reneging will prevent the race of earlier deadline among the applications, because the earlier deadline will have the higher risk not to get the service. That is why, we propose EDF with reneging and per-packet deadline for LAWIN scheduler.

In order to realize LAWIN, couple of fundamental issues have to be considered, a large propagation delay and multi-hop property on the Internet. The propagation delay on the Internet usually reaches up to 300 msec. in case of intercontinental communication. A latency sensitive application decide own deadline taking into account this propagation delay varying with geographical location of the ends. Note that whether LAWIN discriminates between queuing and propagation delay, or not is still under design. However, from the viewpoint from quality of experience, whole end-to-end latency is the most important. The multi-hop property would complicates the prediction of the packet behavior on LAWIN, in case of a packet traverse more than one heavy load hops. For instance, a packet stays on the first hop router until close to its deadline, the drop probability of the packet will increase on the second router. We believe this case is not a critical. Because a router should referred the per packet deadline not as the departure time, but as the priority of forwarding and/or dropping. In the above case, the end host has to specify longer deadline, if it wants to decrease the drop probability. Otherwise, the end host simply re-transmits the packet, because it loses the competition with other traffic always happen even on FIFO queue. Another issue caused by the multi-hop property is which deadline notation, absolute or relative time, we should choose. We cannot address which notation is better, since we should explores more. So, we just enumerate the pros and cons on this paper. The absolute time is better in case of every node does not compatible to LAWIN, such as under deployment. It requires all LAWIN nodes, including end systems, synchronize in enough accuracy, even though time and clock synchronization mechanisms are common as NTP and PTP. The relative time does not require global time synchronization. But the router has to decreases sojourn time of packet, such as original definition of IP TTL field.

The bandwidth of TCP friendly flow should follow the inverse square root of packet loss rate and the inverse of round trip time (RTT)[10]. In case of large propagation delay, the bandwidth is determined by packet loss rate only. Therefore, LAWIN provides the framework that applications can trade off the latency and the bandwidth.

3. EARLIEST DEADLINE FIRST QUEUEING WITH RENEGING

We will make brief introduction of EDF queue behavior under heavy traffic condition. From a theoretical study of generic EDF queueing system, the customers in earlier deadline area elapsed own deadline in higher probability[4]. We also conducted a preliminary simulations on EDF system with reneging. The results support our intuition that the earlier deadline packet will be in higher drop rate. We will show the simulation detail in the workshop venue.

Even though we have not implemented LAWIN on any system, we believe it is a feasible. Because LAWIN can be realized by a small modification of the packet shaping system that we implemented seven years ago[8]. Our implementation scaled up to GbE speed with keeping an enough timing accuracy. This packet pacing mechanism comprises two components of end system, that is, per-packet timer and calendar queue schedular. The per-packet timer that is specified by application or transport protocol stack notifies its departure time to the network interface driver as cross-layer manner. The calendar queue scheduler decreases the cost to keep track of the packet departure sequence and time. In case of LAWIN, the per packet timer is accommodated into the IP header, such as an IP option, able to be referred from out side of the end system,. And the calendar queue scheduler on the router will contribute the EDF scheduler to sort packets in deadline order.

Asymmetric Best Effort (ABE) and EDS impose a higher drop risk to small delay class flows[7, 5]. These proposals classify the packet of flow using a code point of headers as Diffserv. On the contrary to LAWIN, these proposals cannot satisfy the end-to-end deadline requirement from applications due to the lack of deadline notification mechanism. Furthermore, the limited number of service class in ABE and EDS cannot provide enough flexibility.

4. CONCLUSION

We propose LAWIN architecture that provides mechanism trading off between end-to-end latency and loss probably. We believe LAWIN has an advantage compared with existing AQM, even if the header modification is required. AQM possibly drops a packet regardless of good application or not, it seems counter-intuitionally of many network engineers[13]. LAWIN drops packet according its deadline that specified by the end system itself.

5. REFERENCES

[1] Cable Modem Buffer Control. Technical report.

[2] M. Alizadeh, A. Greenberg, D. A. Maltz, J. Padhye, P. Patel, B. Prabhakar, S. Sengupta, M. Sridharan, C. Faster, and D. Maltz. Dctcp: Efficient packet transport for the commoditized data center. In *Proc. Sigcomm*, 2010.

[3] M. Alizadeh, S. Yang, S. Katti, N. McKeown, B. Prabhakar, and S. Shenker. Deconstructing datacenter packet transport. In *Proceedings of the 11th ACM Workshop on Hot Topics in Networks*, pages 133–138. ACM, 2012.

[4] B. Doytchinov, J. Lehoczky, and S. Shreve. Real-time queues in heavy traffic with earliest-deadline-first queue discipline. *Annals of Applied Probability*, pages 332–378, 2001.

[5] B. Gaidioz and P. Primet. Eds: a new scalable service differentiation architecture for internet. In *Computers and Communications, 2002. Proceedings. ISCC 2002. Seventh International Symposium on*, pages 777–782. IEEE, 2002.

[6] J. Gettys and K. Nichols. Bufferbloat: Dark buffers in the internet. *Queue*, 9(11):40, 2011.

[7] P. Hurley and J.-Y. Le Boudec. A proposal for an asymmetric best-effort service. In *Quality of Service, 1999. IWQoS'99. 1999 Seventh*

International Workshop on, pages 132–134. IEEE, 1999.

[8] K. Kobayashi. Transmission timer approach for rate based pacing tcp with hardware support. *PFLDnet 2006*, 2006.

[9] Ł. Kruk, J. Lehoczky, K. Ramanan, and S. Shreve. Heavy traffic analysis for edf queues with reneging. *The Annals of Applied Probability*, 21(2):484–545, 2011.

[10] M. Mathis, J. Semke, J. Mahdavi, and T. Ott. The macroscopic behavior of the tcp congestion avoidance algorithm. *ACM SIGCOMM Computer Communication Review*, 27(3):67–82, 1997.

[11] K. Nichols and V. Jacobson. Controlling queue delay. *Communications of the ACM*, 55(7):42–50, 2012.

[12] R. Pan, P. Natarajan, C. Piglione, M. Prabhu, V. Subramanian, F. Baker, and B. V. Steeg. Pie: A lightweight control scheme to address the bufferbloat problem. Internet-Draft draft-ipng-gseaddr-00.txt.

[13] Wikipedia. Active queue management, 2013. [Online; accessed 22-June-2013].