ISSN:0975-9646

Kiran Babu Kommu et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 2 (6) , 2011, 2479-2481

Active Queue Management for Flow Congestion Control in wireless network

Kiran babu Kommu (M.Tech)
KIET- kakinada

Kakara Ravi Kumar M.Tech
Assistant Professor , KIET Kakinada

Srinivasa Rao S
Dept of physics
S.V.K.P. & Dr.K.S.Raju A &S College,
Penugonda,

B. Ravi Kumar, Dept of computer & information science, HCU,

Abstract

The queue management are flow fairness and queue-length stability However, most prior works dealt with these goals independently. This paper presents two recommendations to the Internet community concerning measures to improve and preserve Internet performance. It presents a strong recommendation for testing, standardization and widespread deployment of active queue management in routers to improve the performance of today's Internet. It also urges a concerted effort of research, measurement, and ultimate deployment of router mechanisms to protect the Internet from flows that are sufficiently responsive to congestion notification.

Introduction

The wireless Wireless communication technology is playing an increasingly important role in data networks. Wireless networks are usually connected to the internet via backbone gateway routers. The packet loss may occur at fusion points that connect the backbone network to the wireless networks. The Internet protocol architecture is based on a connectionless end- to-end packet service using the IP protocol. The advantages of its connectionless design, flexibility and robustness, have been amply demonstrated. However, these advantages are not without cost: careful design is required to provide good service under heavy load. In fact, lack of attention to the dynamics of packet forwarding can result in severe service degradation or "Internet meltdown". Congestion control problem occurs when the demand on the network resources is greater than the available resources and due to increasing mismatch in link speeds caused by intermixing of heterogeneous network technologies. This congestion problem cannot be solved with a large buffer space. Clearly too much traffic will lead to a buffer overflow, high packet loss and large queuing delay. Furthermore, congestion problem cannot be solved by high-speed links or with high-speed processor, because the high-speed link connected via the high-speed switch with the low-speed links will cause congestion at the wireless fusion point of interconnection. Drop Tail has been proposed in [4]. The most operational routers currently use Drop Tail coupled with FIFO (First in first out) scheduling scheme. In Drop Tail, all packets are accepted until the maximum length of the queue is reached and then dropping subsequent incoming packets until space becomes available in the queue. Drop Tail is not appropriate as a feedback control system for high-speed networks because it sustains full queues and this may increase the average queuing delay in the network. More importantly, Drop Tail can cause a lockout due to traffic phase effects and the global synchronization, and thus results in low throughput. The lost packet from a Drop Tail queue will usually be retransmitted by TCP protocol via its retransmission timer. No congestion is detected until the buffer becomes full and the maximum congestion indicator is generated because all arriving packets are dropped. Then each source detects lost packets it will slow down the arrival rate of the sending packets until the queue will be less than the capacity of the link. No congestion indicator will be generated when the queue is not full, each source will increase until overflow happens again In the recent years, Active queue management (AQM) mechanisms have been proposed to provide an efficient queue management by selectively dropping/marking packets when congestion is anticipated so that TCP senders can reduce their transmission rate before an overflow occurs. AQM mechanisms are employed in the Internet by the routers to provide better stability, fairness, and responsiveness to dynamic variations in computer networks. Using queue management. provides a mechanism for protecting individual flows from congestion, introduces its own queue management and scheduling algorithms [Shenker96, Wroclawski96]. Similarly, the discussion of queue management and congestion control requirements for differential services is a separate issue. However, we do not expect the deployment of integrated services and differential services to significantly diminish the importance of the best-effort traffic issues discussed in this paper.

Active Queue Management

The queue managing router lengths is to set a maximum length (in terms of packets) for each queue, accept packets for the queue until the maximum length is reached, then reject (drop) subsequent incoming packets until the queue decreases because a packet from the queue has been transmitted. This technique is known as "tail drop", since the packet that arrived most recently (i.e., the one on the tail of the queue) is dropped when the queue is full.

Random Early Detection Algorithm

The RED, is an active queue management algorithm for routers that will provide the Internet performanc advantages cited in the previous section [RED93]. In contrast to traditional queue management algorithms, which drop packets only when the buffer is full, the RED algorithm drops arriving packets probabilistically. The probability of drop increases as the estimated average queue size grows. Note that RED responds to a time-averaged queue length, not an instantaneous one. Thus, if the queue has been mostly empty in the "recent past", RED won't tend to drop packets (unless the queue overflows, of course!). On the other hand, if the queue has recently been relatively full, indicating persistent congestion, newly arriving packets are more likely to be dropped.

The RED algorithm itself consists of two main parts: estimation of the average queue size and the decision of whether or not to drop an incoming packet.

Average Queue Size

RED estimates the average queue size, either in the forwarding path using a simple exponentially weighted moving average (such as presented in Appendix A of [Jacobson88]), or in the background (i.e., not in the forwarding path) using a similar mechanism.

Note: The queue size can be measured either in units of packets or of bytes. This issue is discussed briefly in [RED93] in the "Future Work" section. when the average queue size is computed in the forwarding path, there is a special case when a packet arrives and the queue is empty. In this case, the computation of the average queue size must take into account how much time has passed since the queue went empty.

Aggressive Managing Flows

The keys to the success of the Internet has been the congestion avoidance mechanisms of TCP. Because TCP "backs off" during congestion, a large number of TCP connections can share a single, congested link in such a way that bandwidth is shared reasonably equitably among similarly situated flows. The equitable sharing of bandwidth among flows depends on the fact that all flows are running basically the same congestion avoidance algorithms, conformant with the current TCP specification [HostReq89]. We introduce the term "TCP-compatible" for a flow that behaves under congestion like a flow produced by a conformant TCP. A TCP- compatible flow is responsive to congestion notification, and in steady-state it uses no more bandwidth than a conformant TCP running under comparable conditions (drop rate, RTT, MTU, etc.) It is convenient to divide flows into three classes: (1) TCP- compatible flows, (2) unresponsive flows, i.e., flows that do not slow down when congestion occurs, and (3) flows that are responsive but are not TCP-compatible. The last two classes contain more aggressive flows that pose significant threats to Internet performance, as we will now discuss.

Non-Responsive Flows

There is a growing set of UDP-based applications whose congestion avoidance algorithms are inadequate or nonexistent (i.e, the flow does not throttle back upon receipt of congestion notification). Such UDP applications include streaming applications like packet voice and video, and also multicast bulk data transport [SRM96]. If no action is taken, such unresponsive flows could lead to a new congestion collapse. In general, all UDP-based streaming applications should incorporate effective congestion avoidance mechanisms. For example, recent research has shown the possibility of incorporating congestion avoidance mechanisms such as Receiver- driven Layered Multicast (RLM) within UDP-based streaming applications such as packet video [McCanne96; Bolot94]. Further research and development on ways to accomplish congestion avoidance for streaming applications will be very important. However, it will also be important for the network to be able to protect itself against unresponsive flows, and mechanisms to accomplish this must be developed and deployed. Deployment of such mechanisms would provide incentive for every streaming application to become responsive by incorporating its own congestion control.

Conclusion

This paper presents the implementation and deployment of RED will also enable the introduction of other new functionality into the Internet. One example of an enabled functionality would be the addition of explicit congestion notification [Ramakrishnan97] to the Internet architecture, as a mechanism for congestion notification in addition to packet drops. A second example of new functionality would be implementation of queues with packets of different drop priorities; packets would be transmitted in the order in which they arrived, but during times of congestion packets of the lower drop priority would be preferentially dropped.

References

[1] L. Kalampoukas, A. Varma, and K. K. Ramakrishnan, “Explicit window adaption: A method to enahance TCP performance,”
IEEE/ACM Trans. Networking, vol. 10, pp. 338–350, June 2002.

[2] T. Lakshman and U. Madhow, "The performance of TCP/IP for networks with high bandwidth-delay products and random
loss," *IEEE/ACM Trans. Networking*, vol. 5, pp. 336–350, June 1997.

[3] T. V. Lakshman, U. Madhow, and B. Suter, "TCP/IP performance with random loss and bidirectional congestion," *IEEE/ACM Trans. Networking*, vol. 8, pp. 541–555, Oct. 2000.

[4] H. Balakrishnan, V. N. Padmanabhan, S. Seshan, and R. H. Katz, "A comparison of mechanisms for improving TCP performance over wireless links," *IEEE/ACM Trans. Networking*, pp. 756–769, 1997.

[5] H. M. Chaskar, T. Lakshman, and U. Madhow, "TCP over wireless with link level error control: Analysis and design
methodology," *IEEE/ACM Trans. Networking*, vol. 7, pp. 605–615, Oct. 1999.

[6] D. A. Eckhardt and P. Steenkiste, "Improving wireless LAN performance via adaptive local error control," in *Proc. IEEE Int.
Conf. Netw. Protocls (ICNP)*, pp. 327–338, Mar. 2003.

[7] R. Ludwig, A. Konrad, A. Joseph, and R. Katz, "Optimizing

[8] Prashanth A., Ashish S., Elizabeth B., Kevin A. and Konstantina P. " Congestion-Aware Rate Adaptation in Wireless Networks: A Measurement-Driven Approach", *IEEE SECON*, San Francisco, CA, June 2008.

[9] Jian Pu; Hamdi, M., "Enhancements on Router-Assisted Congestion Control for Wireless Networks," *Wireless Communications, IEEE Transactions on* , vol.7, no.6, pp.2253-2260, June 2008.

[10] Acharya, P.A.K.; Sharma, A.; Belding, E.M.; Almeroth, K.C.; Papagiannaki, K., "Congestion-Aware Rate Adaptation in Wireless Networks: A Measurement-Driven Approach," *Sensor, Mesh and Ad Hoc Communications and Networks, 2008. SECON '08. 5th Annual IEEE Communications Society Conference on* , vol., no., pp.1-9, 16-20 June 2008.

[11] Jacobson V., (1988), "Congestion avoidance and control", *ACM SIGCOMM Computer Communication
Review*, Vol. 18(4), p.314-329.