Supporting Low Latency near the Network Edge and with Challenging Link Technologies

Markku Kojo*, Ilpo Järvinen*, Hannes Tschofenig†, and Aaron Yi Ding*

* University of Helsinki † Nokia Siemens Networks

In this paper we focus on discussing the challenges of providing low-latency IP service when bursty (Web style) traffic is present on bottleneck access links and, in particular, when the Internet access is with challenging link technologies such as cellular and many other wireless WAN access networks.

Web traffic patterns differ significantly from patterns that occur with a long-running bulk data transfer as Web traffic is bursty consisting of several parallel TCP connctions transferring small amount of data [Bro], [Sou08], [Ram10]. Active Queue Management (AQM) should ensure that such bursty Web style traffic too plays nicely with real-time interactive and other delay-sensitive traffic. As the effect of burstiness with such traffic increases as the inverse of flow aggregation, various edge or close to edge routers such as home routers and cellular access are affected more heavily than largely aggregated core routes. Moreover, as the last-hop or first-hop link on the end-to-end path often is the bottleneck, the end hosts and edge routers play the key role in providing low-latency IP service for typical end users.

In [JCD+13] we show that there are enough buffers with 3G/HSPA networks to create harm for delay sensitive traffic also when only bursty Web-style traffic is introduced. While with a bulk data TCP transfer, the delay grows to seconds in our tests due to excessive buffering, also with TCP flows transferring short Web-objects, the delay becomes large due to parallel TCP connections and the burst of segments sent during the transfer of TCP initial window. In [JDNK12] we show problems with AQM based on Random Early Detection (RED) [FJ93], indicating that RED is still suboptimal even if specifically tuned for such traffic. The main problem with RED is its slowness. Without our optimizations, the exponentially weighted moving average (EWMA) used in calculating average queue lenght in RED fails to detect TCP slow start [APB09] in time and only responds after the moment where tail-drop becomes necessary as the physical buffer space runs out. We believe that when the network has a low load at the time a slow start begins, auto-tuning in general is often subject to similar slowness during the rapid slow start. The Web-traffic optimized Harsh RED we proposed responds rapidly to slow start. However, the next round trips after the initial response often cause unnecessary drops which we could not solve with RED because of how EWMA behaves. We believe that Codel [NJ12], [NJ13] and PIE [PNP+13], [PNP+12] are likely to suffer from similar problems even though being better than RED in general. We also believe that it is too optimistic to try to develop an AQM mechanism that does not require any parametrisation or tuning for specific link environments. The short history with the new AQM proposals (CoDel and PIE) already seems to indicate that specific link technologies may require special attention also with these mechanisms [Whi13].

Even an ideal AQM mechanism together with flow separation and appropriate scheduling at the IP layer is not enough to deliver low-latency IP service with many link technologies. The hardest problem to AQM is uncoordinated buffer usage at layers 2 and 3, which is a cross-layer issue. With many link technologies there is a significant amount of buffer space at layer 2 that eats packets injected by the IP layer. This means that no AQM at the IP layer can work nicely because the packets stand in the link buffers while the IP queues are empty. Solving this is not possible unless the layer 2 and 3 buffers are coordinated appropriately. This is hard as typical link-layer interfaces hide (almost) all details from the IP layer and there is a large number of different link technologies that are typically developed very independently from IP development - and standardization is also quite unconnected for these two layers. Thus, the hard part is not the mechanisms, but how to coordinate currently unconnected development of the two layers. In addition, cellular access as well as many other wireless technologies involve handovers where all traffic for a mobile host is moved from one access router/base station to another as the mobile host moves. This generates an additional challange as new flows may suddenly appear to or disappear from a bottleneck queue.

There are two common reasons for excessive link buffering. First, implementing an efficient link-layer Automatic Repeat reQuest (ARQ) mechanism requires storing the sent but not yet acknowledged data frames in a relatively large send buffer for possible retransmission. Many link-layer ARQ implementations accept a full send buffer of packets from the upper layer to be queued at the link head prior to transmission over the link. However, buffering unsent packets at the link layer is often superfluous. A link sender does not need to buffer a full window of unsent data, but it is quite enough to accept only one or a few unsent IP packets to be buffered. This requires proper flow control between the IP layer and link layer, allowing majority of packets to be kept in the IP queues subject to proper IP-level queue management. Second, some Medium Access Control (MAC) layers, for example those employing a Bandwidth-on-Demand (BoD) capacity allocation mechanism, require a certain amount of data to be stored in the MAC buffer for efficient link capacity allocation. This, however, can also be solved with proper layer 2 and 3 coordination, allowing effective IP level AQM. The link interface may inform the IP layer of the data still waiting for transmission in the link buffers such that this can be taken into account by the IP layer AQM. Alternatively, the IP layer may explicitly inform the MAC layer about the pending data (packets) at the IP layer. This would allow the MAC layer to perform the link capacity allocation efficiently as it knows the amount of the pending data even though not all data is yet in the MAC buffer. In addition, there are also other reasons in particular with many wireless technologies that require some amount of data to be buffered at the link layer for efficient transmission over the link. Hence, a well designed cross-layer flow control mechanism together with proper information sharing between the layers is the key to a working IP level AQM and low latency.

Finally, one significant source for additional delay and delay variation with many challenging (wireless) link technologies is the link ARQ. While the link ARQ is an absolute necessity with many wireless link technologies, additional delay introduced by the link ARQ is often highly undesirable or unacceptable for delay-sensitive IP flows such as interactive audio and video, for example. Therefore, such link layers strive for treating different IP flows with different classes of service and turn off the link ARQ mechanism for flows not benefiting from it. This requires implementing several logical link channels over a single physical link. When ARQ is employed separately only on those logical channels that carry traffic benefiting from ARQ, it allows avoiding head-of-line blocking as the frames for other channels do not need to wait for the retransmission of frames. Again, solving the low-latency problem requires attention from the link layer as flow differentiation at the IP layer alone is not enough. In addition, the flow differentiation and scheduling at these two layers calls for coordination. Therefore, Stochastic Fair Queueing (SFQ) [McK90] that has been suggested as a part of the IP layer solution for low latency is not necessarily suitable as is, because the queues at the IP layer cannot be arranged and scheduled indepedently in all cases but must be mapped to proper link layer queues with coordinated scheduling at the two layers.

As a part of our paper [ANJK08] we show one way of coordinating layers 2 and 3 by providing inter-layer flow control that allows keeping packets at IP queues and by providing proper mapping between IP and link layer queues with QoS support. We also minimize the additional delay due to the retransmissions by reducing the number of retransmission attempts down to one attempt only. The number of retransmission attempts and additional delay can be kept in minimum by protecting the retransmissions with Forward-Error Correction (FEC) in a novel way, resulting in a hybrid-ARQ solution. Moreover, as the link errors on a wireless medium tend to occur in bursts, often being long enough to badly corrupt the entire frame or even several consecutive frames, the FEC-encoded redundancy should not be added separately to each frame as usual. Instead, we organize the retransmitted frames as FEC blocks in such a way that even entirely corrupted frames can be recovered. Link channels for delay-sensitive traffic are not employing ARQ, but may use FEC to protect the frames from bit corruption.

References

[ANJK08] D. Astuti, A. Nyrhinen, I. Järvinen, and M. Kojo. SLACP: a novel link-layer protocol for Wireless WANs. In *Proceedings of The Seventh International Conference on Networking (ICN 2008)*, pages 121–130, April 2008.

[APB09] M. Allman, V. Paxson, and E. Blanton. TCP congestion control. RFC 5681, September 2009.

[Bro] Browserscope. http://www.browserscope.org/?category=network&v=1.

[FJ93] S. Floyd and V. Jacobson. Random Early Detection Gateways for Congestion Avoidance. 1(4):397–413, August 1993.

[JCD+13] I. Järvinen, B. Chemmagate, Y. Ding, L. Daniel, M. Isomäki, J. Korhonen, and M. Kojo. Effect of Competing TCP Traffic on Interactive Real-Time Communication. In *Proceedings of the 14th Passive and Active Measurement conference*, March 2013.

[JDNK12] I. Järvinen, Y. Ding, A. Nyrhinen, and M. Kojo. Harsh RED: Improving RED for Limited Aggregate Traffic. In *Proceedings of the 26th IEEE International Conference on Advanced Information Networking and Applications (AINA)*, March 2012.

[McK90] P. McKenney. Stochastic Fairness Queueing. In *Proceedings of IEEE INFOCOM '90. Ninth Annual Joint Conference of the IEEE Computer and Communication Societies. 'The Multiple Facets of Integration'*, volume 2, pages 733–740, June 1990.

[NJ12] K. Nichols and V. Jacobson. Controlling Queue Delay. *ACM Queue*, 10(5), May 2012.

[NJ13] K. Nichols and V. Jacobson. Controlled Delay Active Queue Management. Internet Draft, February 2013. Work in progress.

[PNP+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, December 2012. Work in progress.

[PNP+13] 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. In *Proceedings of the 2013 IEEE Conference on High Performance Switching and Routing*, July 2013. To appear.

[Ram10] S. Ramachandran. Web metrics: Size and number of resources, May 2010. http://code.google.com/speed/articles/web-metrics.html.

[Sou08] S. Souders. Roundup on Parallel Connections, March 2008. http://www.stevesouders.com/blog/2008/03/20/roundup-on-parallel-connections/.

[Whi13] G. White. Active Queue Management Algorithms for DOCSIS 3.0. WHITE PAPER, CABLELABS, April 2013. http://www.cablelabs.com/downloads/pubs/Active_Queue_Management_Algorithms_DOCSIS_3_0.pdf.