Adaptive and Scalable Congestion Control wanted!

Mirja Kühlewind
Institute of Communication Networks and Computer Engineering
University of Stuttgart, Germany
mirja.kuehlewind@ikr.uni-stuttgart.de

ABSTRACT

Today's Internet is mainly optimized for high goodput. This is not sufficient anymore, because more and more applications and thus a growing fraction of the Internet traffic require low latency. One aspect would be minimizing network queuing delay. This task should be performed by the network layer e.g. by configuring small buffers. Such a simple approach could be sufficient, if the transport layer congestion control mechanism would be able to adapt to and scale with any network situation (without service degradation). This is not the case today, as congestion control is regarded to be mostly fixed. If the network is changed first, transport throughput might suffer. But at the same time this will give a strong incentive to enhance congestion control. Shifting the control back to the transport layer, by removing any assumptions on higher layer behavior from the network layer, will enable a less complex network management and higher flexibility for distributed application optimizations.

1. PROBLEM STATEMENT

Today's Internet is mainly optimized for high throughput and low loss rates; thus to utilize network resources most efficiently. This strategy implies large enough network buffers to, on the one hand, absorb small data bursts and, on the other hand, provide sufficient space for TCP congestion control to work efficiently. In the worst case, with low multiplexing and/or high synchronization of flows, today's loss-based congestion control algorithms require a buffer space of one Bandwidth-Delay-Product (BDP). Thus the network is optimized for the transmission of large amounts of data where only the completion time is relevant for the user's Quality of Experience (QoE). But the usage of Internet evolved to be much more than this.

More and more applications are emerging with narrow latency requirements or hard completion deadlines like real-time video conferencing or interactive cloud services. Typically this kind of services implements a lot of intelligence in the application layer to overcome the limitation of the current networking performance with respect to latency. The actual experienced latency adds up different components, where delay induced by the network is one large component which is not under control of the application. Network delay usually has a fix component, which can be optimized by e.g. caching/service placement and shortest path routing, and a dynamic component composed of potentially various queueing delays. Especially for applications which only allow for a certain maximum per-packet delay, queuing delay needs to be minimized. Unfortunately, while the networking layer should be the one responsible for providing a certain Quality of Service (QoS) to the higher layer, it does not take its responsibilities ragarding the need for low latency.

Potential solution approaches.

To address this problem in the network, it has already been proposed to provide e.g. a separate low latency service next to the currently available low loss Best-Effort service [2, 4, 5]. These proposals are based on the assumption that most applications require either low latency or low loss. Thus a sending entity can decide between the trade-off of these to QoS parameters. In a network node two queues are differently configured in terms of maximum queue size and potentially Active Queue Management (AQM). Finally, a scheduling strategy to dequeue the packets needs to be applied. High additional complexity is introduced in the scheduling algorithm to address how to avoid unfairness between the two services and how to handle unresponsive flows. But in fact neither this complexity does belong in the network layer nor the network layer should determine these decisions.

Decouple network and transport.

Unfortunately, many networking mechanisms are based on assumptions about the behavior of higher layer mechanisms. Most often they assume a congestion control scheme similar to today's loss-based mechanisms. This assumption imposes complexity in the network and at the same time impedes any enhancement in congestion control [3]. In fact, breaking the independance of layers is exactly the cause for the large buffer dimensioning and thus part of the bufferbloat problem. The network layer should be decoupled form the transport layer and thus not rely on a specific behavior of the upper layer.

The network QoS parameter should only be set up as desired for the offered service. E.g. if a certain per hop maximum queuing delay is part of such a low latency service, this could be implemented by simply configuring small buffers. An alternative would be to signal congestion early (e.g. by using Explicit Congestion Notification) to keep the buffer empty except when really needed for a short period of time. Such an approach is desirable but only deployable if it can be assured that all flows react to this early congestion signal. Of course today's TCP might suffer in throughput but applications which rely on low latency will gain better QoE.

No matter which kind of service is offered, a network operator should only decide which QoS parameters should be implemented without making assumption about the higher layers and thus leave the capacity sharing decision to the transport layer. Thus this kind of decisions lies in the responsibility of the network layer while the transport layer should be able to cope with any of these situation and to always utilize the provided resources most efficiently. Thus change the network first, the endsystems will follow!

2. REQUIREMENTS ON CONGESTION CONTROL

We now further derive requirements on congestion control of the transport layer to cope with any future networking environments.

Adaptivity Congestion control should adapt its decrease and increase behavior to the network conditions to be able to utilize every link independently of the provided buffer size. Current loss-based congestion control schemes need a certain queue length to be able to fully utilize a link, mainly because of halving the congestion window on congestion events.

Scalability Congestion Control should appropriately adjust to new capacity conditions even in high speed networks. That means congestion control should be able to quickly utilize available bandwidth as well as quickly yield capacity for new flows. To achieve this a congestion control scheme needs to get frequent network feedback independently of the available bandwidth to detect changing network conditions early.

Convergence Congestion control should quickly converge to a stable state with or without competing flows using the same or another congestion control scheme. Moreover, congestion control should not overload the network and avoid unnecessary overshoots. Thus congestion control should make sure that the queue frequently runs empty to give starting flows the chance to grep capacity quickly without causing large overshoots.

Capacity Sharing Congestion control schemes must be able to share the available bandwidth with different congestion control schemes, at least as long as both schemes react to the same congestion feedback signal(s). As loss is the strongest congestion signal today, congestion control should always be able to compete with other flow relying only loss-based feedback. In fact equal per-flow sharing is not desirable [1] as the network does not know the user intent and the application requirements. It is crucial that every flow can at least grab some of the capacity. One approach to control the network share could be realized by providing a higher layer configuration interface to change the congestion control aggressiveness dependent on the actual application requirements. This would also provide a simple way to implement a relative prioritization between flows (of one user).

3. CONCLUSION

A low-latency network service could be provided by e.g. simply implementing small queues, use and enforcement of an early feedback signal, or service differentiation in the network. The network layer should independently implement low latency first. This will provide an incentive for the transport layer to follow and implement more advanced congestion control than used today. The congestion control of the transport layer should be able to adapt to and scale with all kind of network conditions. Moreover, the transport layer would also be more flexible and, thus, could better address application requirements.

4. REFERENCES

[1] B. Briscoe. Flow rate fairness: dismantling a religion. *SIGCOMM Comput. Commun. Rev.*, 37(2):63–74, Mar. 2007.

[2] V. Firoiu and X. Zhang. Best effort differentiated services: Trade-off service differentiation for elastic applications. In *in Proc. IEEE ICT 2001*, 2000.

[3] B. Ford and J. Iyengar. Breaking up the transport logjam. In *HOTNETS*, 2008.

[4] P. Hurley, J.-Y. Le Boudec, P. Thiran, and M. Kara. ABE: providing a low-delay service within best effort. *Network, IEEE*, 15(3):60–69, 2001.

[5] M. Podlesny and S. Gorinsky. RD network services: differentiation through performance incentives. *SIGCOMM Comput. Commun. Rev.*, 38(4):255–266, Aug. 2008.