Requirements on Congestion Control: Adaptive and Scalable CC wanted!

ISOC Workshop on Reducing Internet Latency

Mirja Kühlewind <mirja.kuehlewind@ikr.uni-stuttgart.de>
September 26, 2013

Universität Stuttgart
Institute of Communication Networks
and Computer Engineering (IKR)
Prof. Dr.-Ing. Andreas Kirstädter

Outline

Problem Statement: Need for Low-Latency Service in the Internet & Network Delay

Potential Networking Solution Approaches

- Configuration of small queues
- Use and enforcement of early congestion feedback
- Service differentiation to support low latency

Claims to Support Low Latency

- Decouple network and transport layer mechanisms!
- Change network first, endsystems will follow!

Requirements on Congestion Control

Adaptivity, Scalability, Convergence, Capacity Sharing

Conclusion: Congestion Control should Adapt to & Scale with all Network Conditions

Problem Statement

Need for Low Latency Support in the Internet

- Internet mainly optimized for high through-put and low loss rates
  - Large enough buffers to absorb small data bursts and provide sufficient space for TCP congestion control to work efficiently (worst case: BDP)
  - Optimized for the transmission of large amounts of data where only the completion time is relevant for the user's Quality of Experience (QoE)

- More and more application with narrow latency requirements/hard deadlines
  - E.g. Real-time audio, interactive cloud services, or financial trading
  - Intelligence in the application layer try to cope with limitation of the current networking performance with respect to latency (e.g. aggressive sending & FEC)
  - Applications cannot reduce network delays introduced by a different entity (e.g. cross traffic filling a large queue)

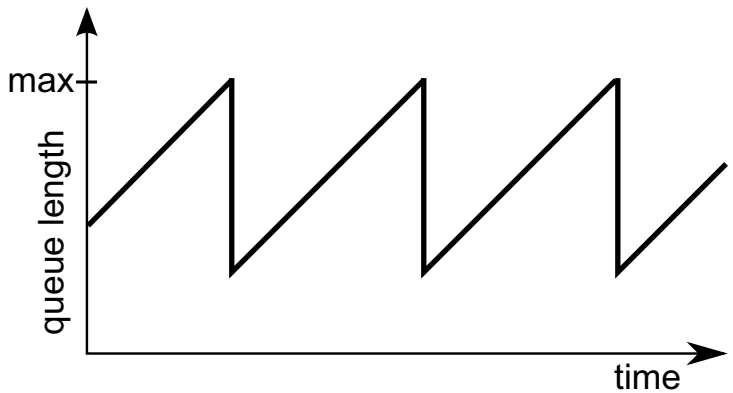
Problem Statement

Network Delay

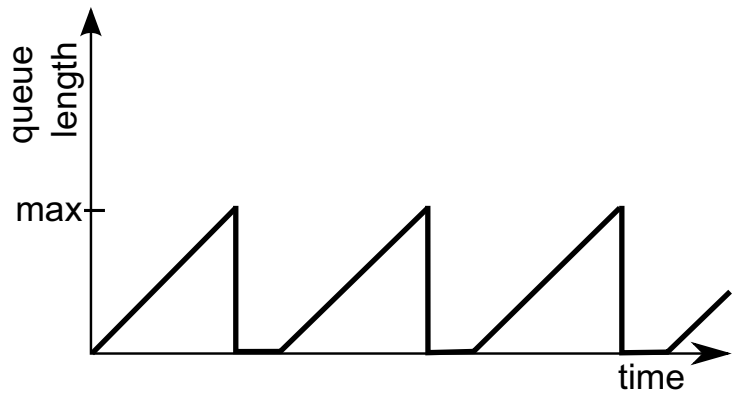
- **Fix component:** propagation delay, processing offsets

  Optimized by e.g. caching/service placement/composition and shortest path routing

- **Dynamic component:** various queuing delays of potentially large network buffers



standing queue adds additional delay



empty queue causes link underutilization

→ **Goal:** Average queuing delay should be minimized!

as some applications can only handle a certain maximum per-packet delay

but still allow for small bursts and be able to keep the link full

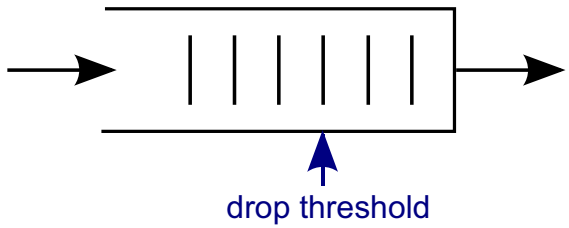
Potential Networking Solution Approaches

1. **Configuration of small queues**

   Utilization might suffer when using today's congestion control

2. **Use and enforcement of early congestion feedback**

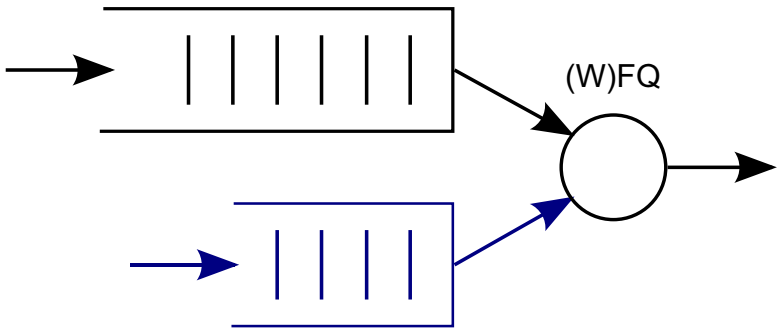
   e.g. by AQM and ECN-based low marking threshold
   to maintain an empty queue and still allow short bursts



3. **Service differentiation (Best-effort and Low-Latency)**

   – Both services provide only benefits for a certain application class (low-latency or low-loss)
   – Implementation: two queues with different queue size/AQM and complex scheduling strategy

     to avoid unfairness / handle unresponsive flows



→ **Neither this complexity belongs in the network layer, nor the network layer should determine these decisions!**

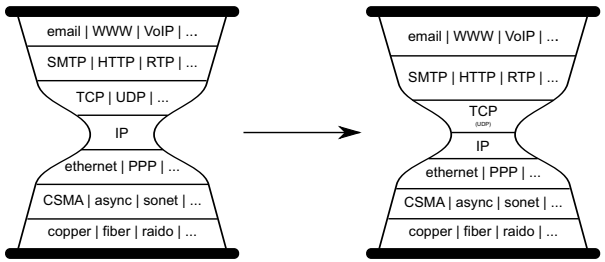
Claims to Support Low Latency

- **Problem:** Breaking the independence of layers

  Networking mechanisms assume a certain behavior of higher layer mechanisms

  mostly loss-based Reno-like congestion control

  → Imposes complexity, impedes enhancements in CC



→ **Goal:** Network & transport layer needs to be decoupled!

  → Network should only decide which QoS parameters should be implemented without making assumption about the higher layers

  → Leave the capacity sharing decision to the transport layer

- **Problem:** TCP congestion control currently needs a certain queue size to fully utilize the link

→ **Goal:** Transport layer needs to cope with any network conditions & always utilize the provided resources most efficiently!

  → Low latency service should be implemented by the network first

  → Transport layer needs to adapt to network

  → Transport layer should provide different services depending on application requirements

Requirements on Congestion Control

Adaptivity

Be able to utilize every link (independent of the buffer size)

Adapt increase/decrease behavior to network conditions

→ (Better) information on queue fill level needed

Scalability

Be able to appropriately adjust to new conditions (even in high speed networks)

Quickly utilize available bandwidth and quickly yield capacity for new flows

→ Frequent network feedback needed to detect changing network conditions early (independently of the available bandwidth)

Convergence

Be able to converge to a stable state (quickly) with or without competing flows

Do not overload the network and avoid unnecessary overshoots

Capacity Sharing

Be able to share the available bandwidth with different congestion control schemes

at least as long as the schemes react to the same congestion feedback signal

→ Provide config interface to change aggressiveness dependent on app needs ("one fits all")

Conclusion

- The network layer should independently implement low latency first

  by e.g. simply implementing small queues, use and enforcement of an early feedback signal, or service differentiation in the network

- Transport layer will follow and implement more advanced congestion control
  - Congestion control should be able to adapt to and scale with all kind of network conditions
  - Transport layer should also be more extensible to better address application requirements