Oscillations and Buffer Overflows in Video Streaming under Non-Negligible Queuing Delay

Presented by Seong-Ryong Kang

Yueping Zhang and Dmitri Loguinov

Department of Computer Science
Texas A&M University
College Station, TX 77843

Outline

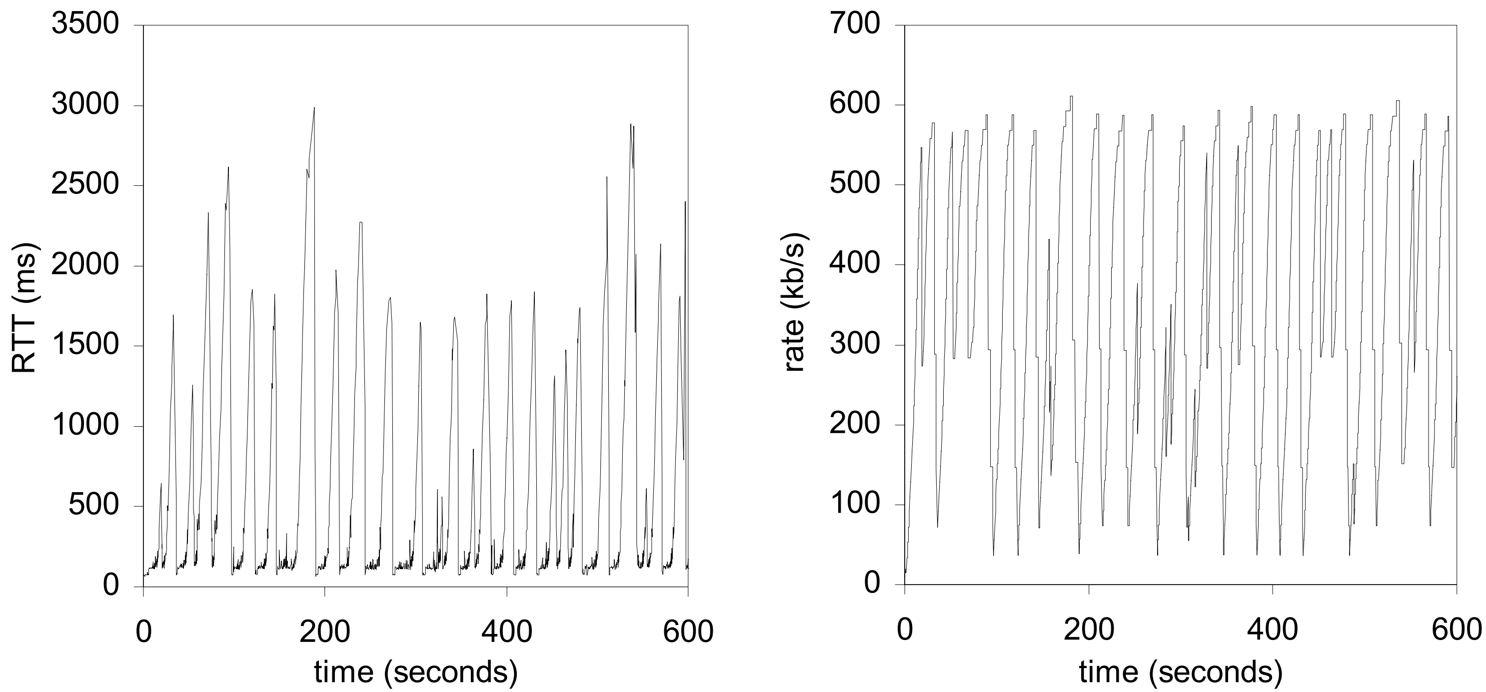
- *Introduction*
- Window-Based AIMD (TCP)
- Rate-Based AIMD
- Scalable TCP
- TFRC
- Conclusion and Future Work

Introduction I

- Feedback delay is widely present in most practical networks
- Internet video streaming applications are challenged by feedback delay in the loop of congestion control
- Queuing delay contributes to the dynamics of Internet congestion controls

Introduction II

- Experiment: Oscillating Behavior of Real-Time Video Streaming Using Rate-Based AIMD
  - Video streaming over 512-kb/s residential DSL. Evolution of the RTT (left) and that of the IP-layer sending rate (right)



Outline

- Introduction
- *Window-Based AIMD (TCP)*
- Rate-Based AIMD
- Scalable TCP
- TFRC
- Conclusion and Future Work

Window-Based AIMD I

We first examine window-based AIMD (TCP)

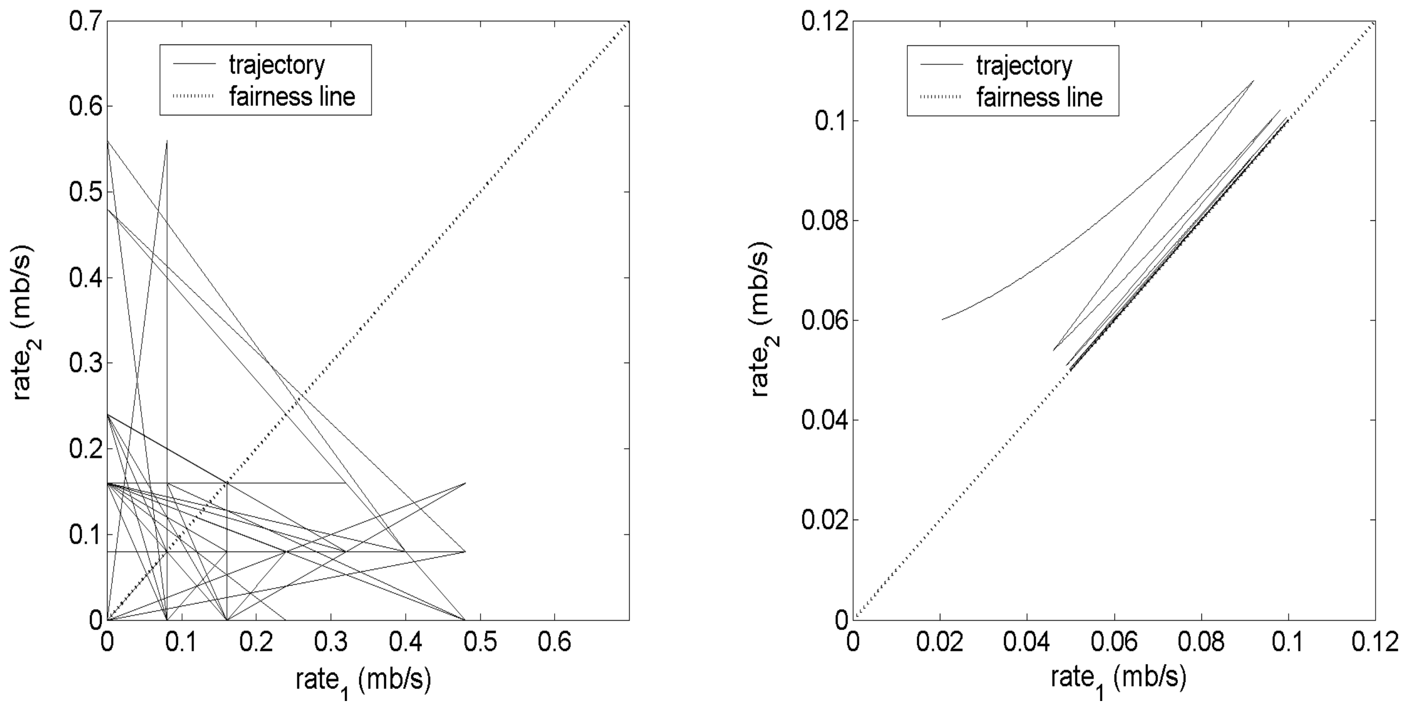
- Control Equations

$$W(t) = \begin{cases} W(t-1) + 1/W(t-1) & \text{per ACK} \\ \beta W(t-1) & \text{per loss} \end{cases}$$

  - For each positive ACK, congestion window $W(t)$ is increased by $1/W(t)$
  - For each packet loss, $W(t)$ is decreased multiplicatively to $\beta W(t)$ ($0<\beta< 1$)
    - In TCP, $\beta = 0.5$

Window-Based AIMD II

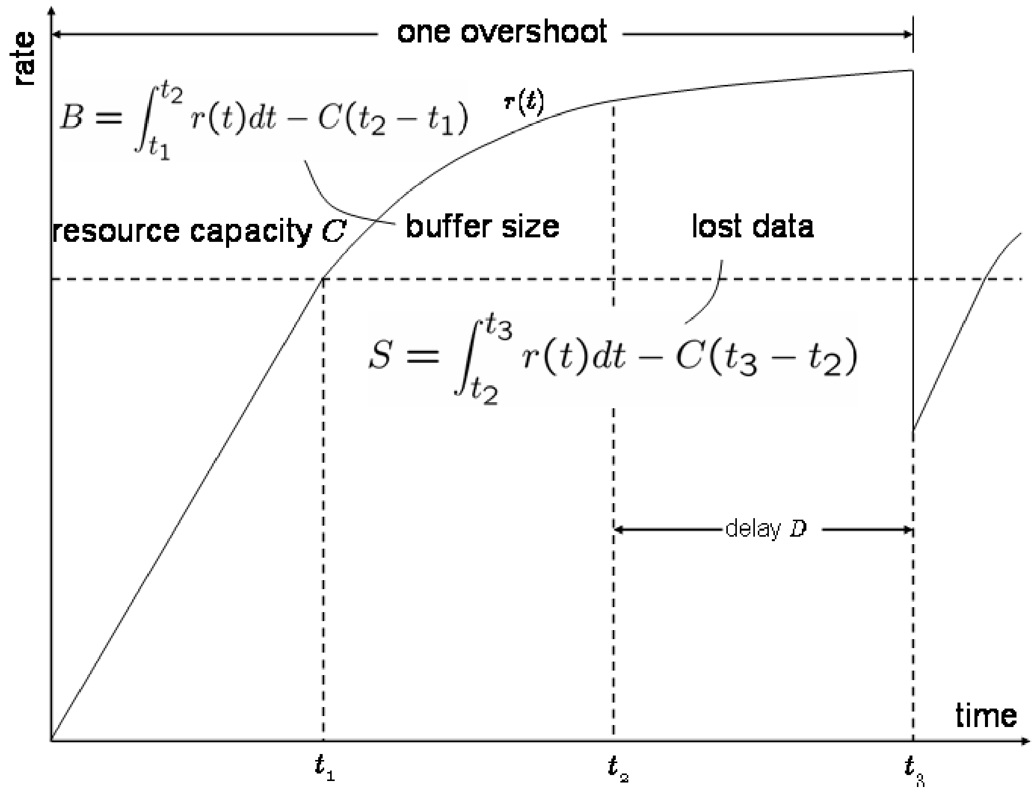
- Delayed-Related Oscillations
  - Trajectories of two TCP flows: 10 ms delay (left) and ideal immediate feedback (right)



Window-Based AIMD III

- Buffer Overflow Problem

To better understand how performance of congestion controls is affected by feedback delay, we examine their buffering behavior under large queuing delay



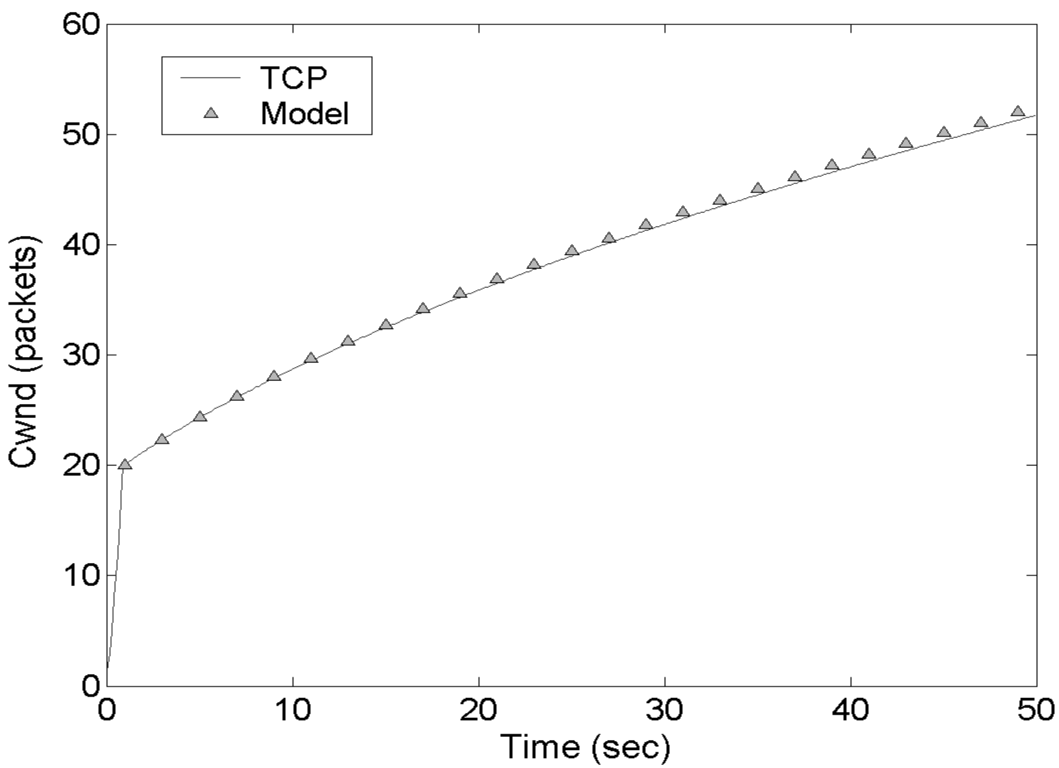
Window-Based AIMD IV

First, we need the following preliminary result

- Window Growth
  - Lemma 1: TCP increases its congestion window $W(t)$ at a rate proportional to the square root of time t after the bottleneck link is saturated
    - $W(t)$ tends to ∞ for sufficiently large delay

Window-Based AIMD V

– Ns-2 simulation: congestion window of a single TCP flow under large queuing delay
  - Packet size $= 1024$ bytes, $C = 2$ mb/s (244 pkt$/$s)

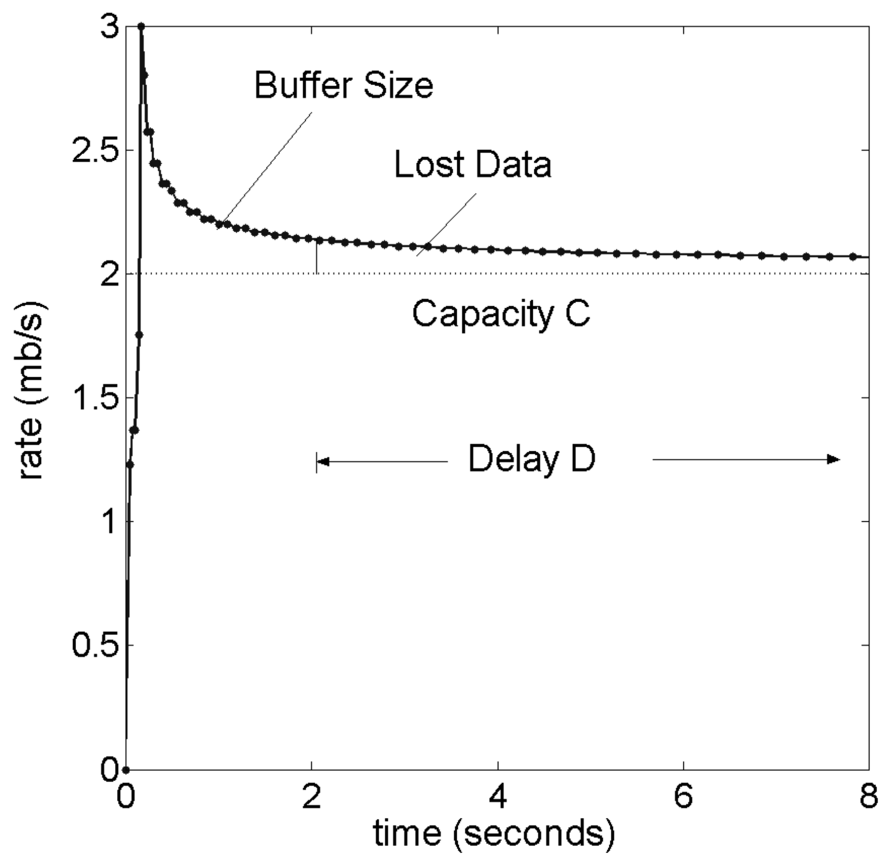


Window-Based AIMD VI

- Buffer overflow
  - Lemma 2: the aggregate amount of lost data $S(t)$ of TCP during each overshoot is proportional to the square root of queuing delay D
    - The fact that TCP keeps increasing its sending rate after the bottleneck link is saturated contributes to its oscillation
    - The larger the queuing delay is, the more lost packets TCP suffers
    - Since TCP sends out $1+1/W(t)$ packets per ACK and $W(t)$ approaches ∞, TCP eventually sends one packet per ACK, i.e., the sending rate converges to C

Window-Based AIMD VII

– Ns2 Simulation: a single TCP flow (slow start is disabled) under large queuing delay



Outline

- Introduction
- Window-Based AIMD (TCP)
- *Rate-Based AIMD*
- Scalable TCP
- TFRC
- Conclusion and Future Work

Rate-Based AIMD I

We next examine rate-based AIMD

- Control equations

$$r(t) = \begin{cases} r(t - RTT) + \alpha & \text{RTT} \\ \beta r(t - RTT) & \text{per loss} \end{cases}$$

  - Rate-based AIMD adjusts its sending rate per RTT, instead of per ACK
  - RTT consists of propagation and queuing delays

$$RTT(t) = d + q(t)/C$$

where instantaneous queue size $q(t)$ is

$$\dot{q}(t) = r(t) - C$$

Rate-Based AIMD II

- Assume constant RTT
  - Unlike window-based protocols, which estimate the RTT based on positive feedbacks, it is difficult for rate-based methods to timely and accurately estimate the RTT
  - A closed-form solution to the exact queuing model coupled with end-flow equations does not exist for both rate-based AIMD and TFRC
  - Thus, we consider the simpler case where the RTT is constant due to its inaccurate estimation by rate-based AIMD and TFRC

Rate-Based AIMD III

- Buffer Overflow
  - Lemma 3: under constant RTT, the aggregate amount of lost data $S(t)$ of rate-based AIMD during each overshoot is proportional to D^2
    - Rate-based AIMD grows the buffer to ∞ under sufficiently large D
    - Under the same queuing delay D, rate-based AIMD suffer more packet loss than window-based AIMD

Outline

- Introduction
- Window-Based AIMD (TCP)
- Rate-Based AIMD
- *Scalable TCP*
- TFRC
- Conclusion and Future Work

Scalable TCP I

We next consider Scalable TCP that is recently proposed as a replacement of the conventional TCP

- Control Equations

$$W(t) = \begin{cases} W(t-1) + 0.01 & \text{per ACK} \\ 0.875 W(t-1) & \text{per loss} \end{cases}$$

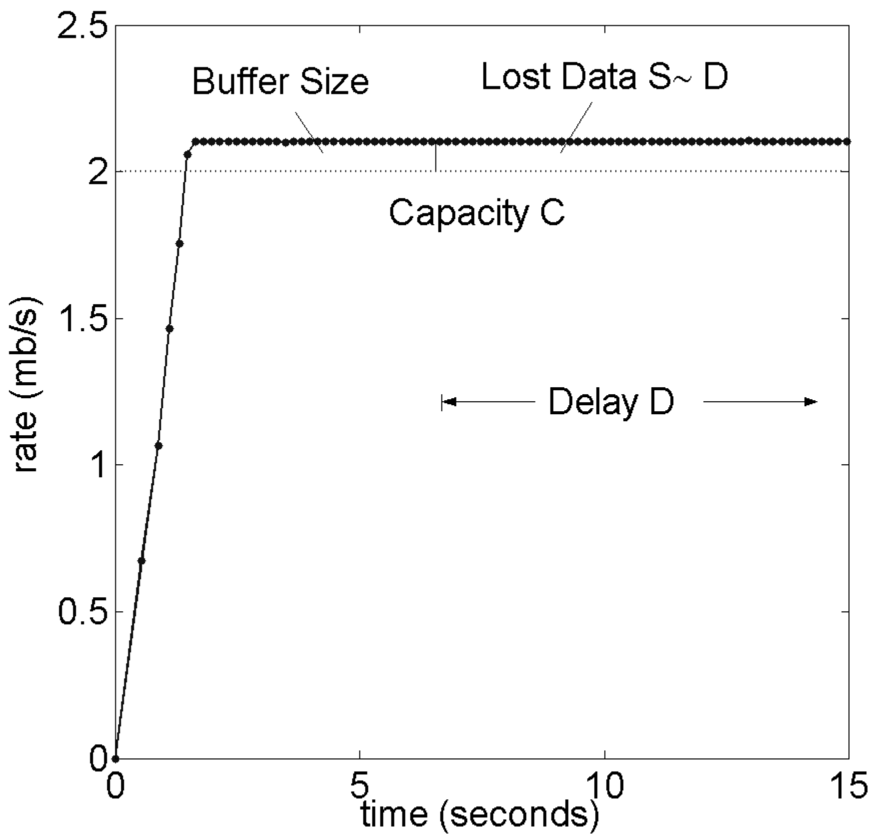
  - Scalable TCP is a window-based MIMD protocol
  - Scalable TCP is suitable for high-bandwidth networks

Scalable TCP II

- Buffer Overflow
  - Lemma 4: the aggregate amount of lost data $S(t)$ of Scalable TCP during each overshoot is proportional to D
    - Under the same queuing delay D, the packet loss of Scalable TCP is between window-based AIMD and rate-based AIMD

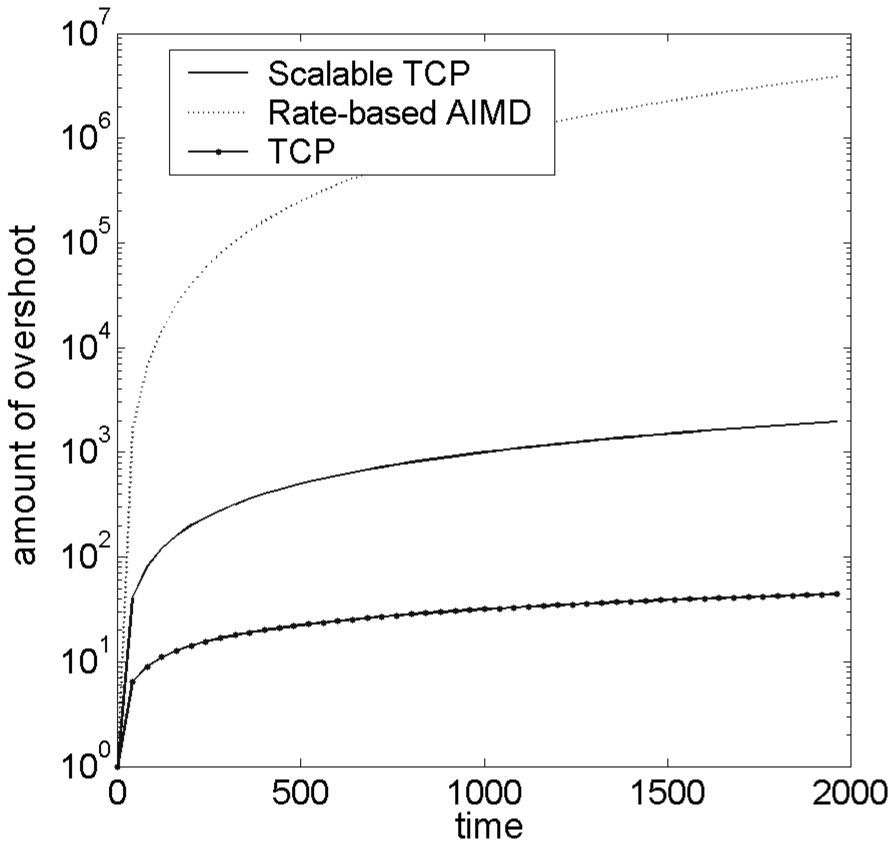
Scalable TCP III

– Ns-2 simulation: rate adjustment of Scalable TCP under large queuing delay



Comparison

– The buffering behavior of TCP, rate-based AIMD, and Scalable TCP under large queuing delay



Outline

- Introduction
- Window-Based AIMD (TCP)
- Rate-Based AIMD
- Scalable TCP
- *TFRC*
- Conclusion and Future Work

TFRC I

Finally examine delayed behavior and stability of TFRC

- Control Equation

$$r(n) = \frac{MTU}{\sqrt{p(n-\Delta_1)}RTT(n-\Delta_2)}$$

  - MTU: maximum transmission unit
  - $p(n)$: long-term average packet loss
  - Metrics Δ_1 and Δ_2 are, respectively, feedback delays of $p(n)$ and $RTT(n)$
  - Same as the discussion of rate-based AIMD, RTT is assumed to be constant

TFRC II

- Buffer Overflow
  - Lemma 5: under constant RTT, the amount of lost data $S(t)$ in TFRC during each overshoot is proportional to D^2
    - The buffering behavior of TFRC under queuing delay is similar to rate-based AIMD

TFRC III

- Stability of AQM-Enabled TFRC
  - Lemma 6: under AQM feedback and constant RTT, TFRC can only be stabilized at the cost of no less than 33% packet loss
    - Under AQM feedback

$$p(t) = \left(\frac{r(t)-C}{r(t)}\right)^{+}$$

letting $\omega = MTU/RTT$, the unique non-negative equilibrium point of sending rate $r(t)$ is

$$r^* = \frac{C+\sqrt{C^2+4\omega^2}}{2}$$

TFRC IV

– Check local stability of r^* by linearization

$$\left.\frac{\partial r(t)}{\partial r}\right|_{r*} = \left.\frac{-\omega C}{2\sqrt{r}(r-C)^{3/2}}\right|_{r*} = \frac{-C}{2(r*-C)}$$

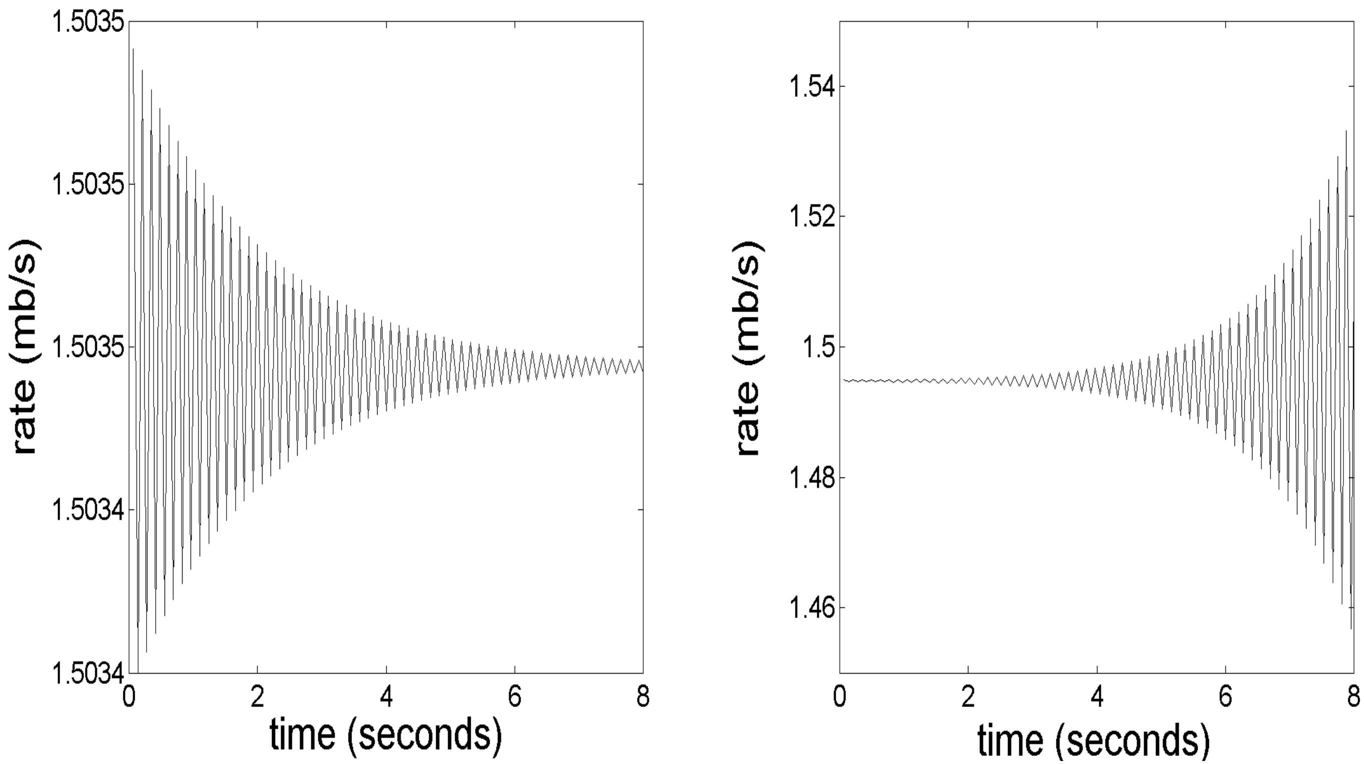
- Locally stable if and only if

$$\left|\frac{-C}{2(r^*-C)}\right| < 1 \Rightarrow r^* > \frac{3}{2}C$$

- AQM-enabled TFRC has to suffer more than 33% packet loss to achieve stability.
- Thus, TFRC does not benefit from AQM feedback

TFRC V

- Simulation: TFRC under AQM feedback when $r^*=1.503C$ (left) and $r^*=1.494C$ (right)
    - $C = 1$ mb/s, $MTU = 1500$ bytes



Outline

- Introduction
- Window-Based AIMD (TCP)
- Rate-Based AIMD
- Scalable TCP
- TFRC
- *Conclusion and Future Work*

Conclusion

- All AIMD and TFRC oscillate and overflow the buffer under large queuing delay
- Window-based protocols offer better (but far from ideal) performance under delay than rate-based protocols
- Our general conclusion: More effort should be put into design of AQM-enabled congestion controls that are provably stable under arbitrary delay

Future Work

- Analyze rate-based AIMD and TFRC under dynamic RTT and an accurate queuing model
- Design AQM-enabled congestion controls that achieve high utilization, freedom from oscillation, small queuing delay, and stability in the equilibrium