Congestion control algorithm for lower latency and lower loss media transport – initial draft

draft-ohanlon-rmcat-dflow-00

Piers O'Hanlon
Ken Carlberg

Background

- Most existing congestion control is loss-based
  - Results in full queues => **high delay and loss**
    - E.g. TCP{new/Reno, Cubic}, TFRC, SCTP, TFWC
- Some delay-based schemes – But most don't aim to *minimise* delay
    - E.g. CARD, Tri-S, Vegas, CTCP (partial), CxTCP, LEDBAT
- A few do now (but mostly unpublished)
    - Ghanbari fuzzy logic, Google RRTCC

DFlow: Objectives

- **Lower Delay**: Needed in today's bufferbloated Net
  - Should stay below 150ms [ITU.G114] (not exceed 400ms)
- **Lower Loss**: Loss is bad for media (limited retransmit)
  - Low delay usually implies low loss as queues not full
- **Smoothness**: Codec output generally smooth
  - Within the constraints of the media, codec, and network path.
- **Fairness**: Should aim to be reasonably fair
  - Initially we aim for self fairness and we aim to tackle TCP fairness in later rev.

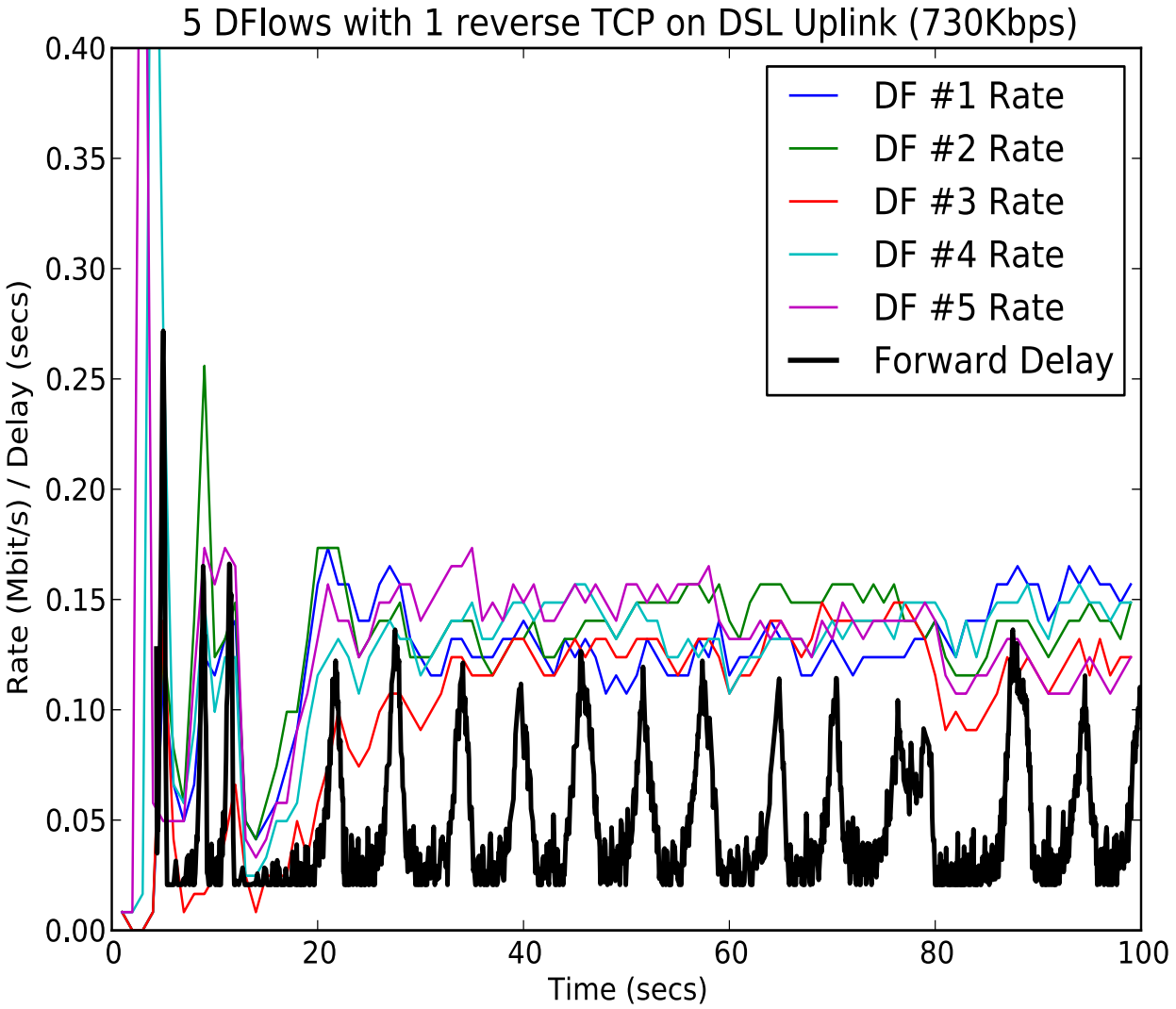
Objectives (Planned)

- [**Burst Management**]: Mechanisms to handle the bursty nature of media
  - E.g. Allow bursts when conditions permit
  - Providing for smoother quality
- [**Loss-based mode**]: Mechanisms to allow for 'fair' thruput against loss-based CC flows
  - Without additional network support (e.g. Codel) delay (and loss) would be largely beyond control.

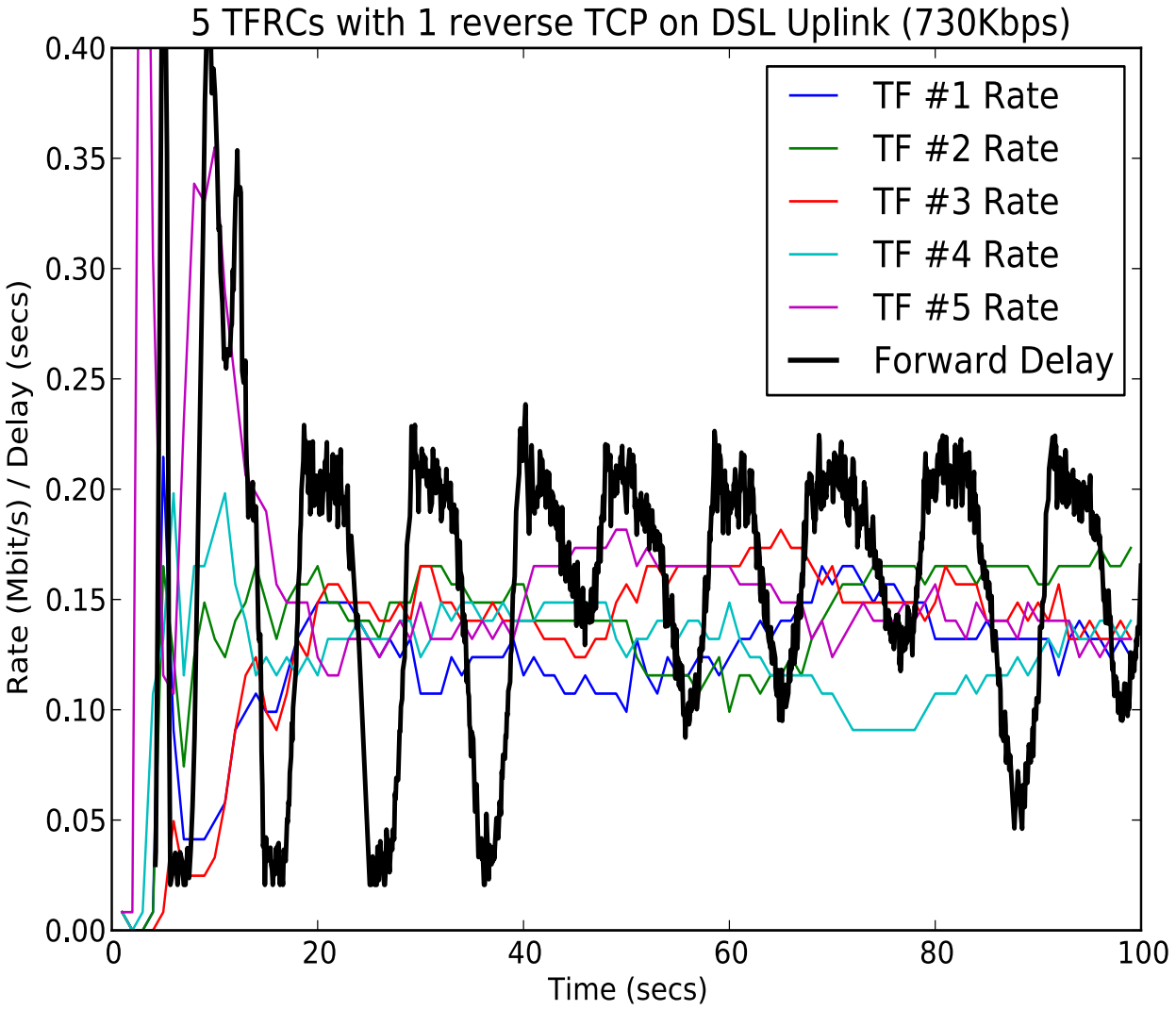
Design Outline

- Loosely based on TFRC design
  - Disabled delay-based oscillation smoothing
    - Due to interference with delay-based behaviours
- TCP equation to derive an operating rate
- Utilises ‘delay losses’
  - Based on relative delay and its derivative
- Building ‘congestion event history’
  - Based on TFRC ‘loss event history’ mechanism

Simulation: DFlow



Simulation: TFRC



Discussion

- This is preliminary work and we're seeking feedback
- Refinement of planned objectives
- More simulations and testing