Implementing stable *TCP variants*

IPAM Workshop on Large Scale Communications Networks
April 2002

Tom Kelly

ctk21@cam.ac.uk

Laboratory for Communication Engineering

University of Cambridge

Overview

- Problem, tools and goals: a systems perspective
- Real world heterogeneity and constraints
- Some common implementation dilemmas
- An implementation of a primal scalable TCP
- Performance: why, what and how to measure it?

The problems

- The Internet's *best effort* packet delivery unsuitable for some applications
  - Drop rates, inter-packet jitter, and long round trip times
- TCP AMID dynamics not helpful for some applications
  - Rapidly fluctuating quality for some apps
- The Internet's resource allocation decisions are opaque and fixed
  - RTT unfairness fixed, multiple connections, etc.
  - Why is resource allocation *policy* a fixed part of the architecture?

The goals

- Is a low-loss and low-delay IP network possible with a new congestion control framework?
- Must be as decentralized, simple, flexible, and robust as the current Internet
- If so, what does it look like?
  - Range of dynamic rate options? (smoothed through noisy)
  - Robust to a wide range of traffic patterns? (web, inelastic traffic, etc.)
  - Scaling to future high speed networks?
  - How is resource allocation policy expressed?
  - Does resource allocation policy need enforcement?
  - What happens when something goes wrong? (bugs, attacks, etc.)

The tools

- Control theory
  - flow stability with heterogeneous lags and links
- Optimization methods
  - resource allocation framework
- Stochastic processes
  - guides allocation policy, queue and packet transmission processes
- Computer network engineering
  - explicit congestion signaling and implementation methods

Heterogeneity makes things hard!

- The Internet is heterogeneous in many dimensions
  - Round trip delays in low-loss and low-delay networks could be 1ms-1000ms
  - Link speeds already scale from 56kbps through 10Gps
  - Multiplexing levels from a couple of connections through millions
- Might need to scale over *more* orders of magnitude than before?
- Designing a decentralized flow control which meets the goals in real scenarios is *hard*

A scalable control theorem

Let $cwnd_r$ be the window on route r:

- no mark: $cwnd_r \mapsto cwnd_r + a_r cwnd_r^n$
- mark: $cwnd_r \mapsto cwnd_r - b_r cwnd_r^m$

Theorem (Vinnicombe): This network is locally stable if

Source condition: $$a_r \left(\hat{x}_r T_r\right)^n < \frac{1}{\gamma} \qquad \forall r \in R$$

Link condition: $$\frac{\hat{y}_j p_j'(\hat{y}_j)}{p_j(\hat{y}_j)} \leq \gamma \qquad \forall j \in J$$

- Choosing $n = 0, m = 1$ is *scalable* and appears to converge globally

Windows and RTT

- An example:
  - RTT 5ms, bandwidth 100Mbps, 100 connections, and packets of size 1000 bytes (e.g. UCL $\leftrightarrow$ Cambridge, today)
  - Each connection needs an average window of 0.5 packets!
- Trends will make RTTs smaller: better content distribution, faster computers, faster links
- Can’t clock transmission off acks without some delay
- Credibility of fluid model in such scenarios

A rate pacing hybrid

- Rate based but keep conservation of packets principle
- Maintain $cwnd$ as a real number
- Let $\bar{T}_r$ be an estimator of T_r
- Use a paced rate of $\frac{cwnd}{\bar{T}_r}$
- Use $\lceil cwnd \rceil$ as limit on packets in flight
- Careful $\bar{T}_r$ might wander with queuing delay
- A constantly re-sampled minimum for $\bar{T}_r$ seemed better than some

Resource allocation

Equilibrium point at:

$$x_r = \frac{s.a_r}{T_r b_r} \frac{1 - P_r}{P_r}$$

where s is packet size and P_r is marking rate

- Remove RTT bias by setting $\tilde{b}_r = T_r.b_r$
- a_r has an upper bound due to stability
- s trades overhead against information feedback
- Share determined by weight $\tilde{b}_r$

Weighing problems

- Small RTTs give $b_r > 1$ due to RTT bias removal
- In practice rate variance too high for $b_r > 0.25$
- Cap $b_r \leq 0.25$ and scale a_r so that share is the same
- Allowing weights is good for policy expression...
- ...but how do incentive apply in a decentralized system
  - ECN noncing, congestion cost accounting, etc.

Convergence time and response

- Keep low-loss and low-delay when flow arrival and departure dynamics included
- Averaging to remove noise conflicts with sensitivity
- More work needed with real arrival processes
- Slow start is not great here; high marking rates
- My approach was to inflate the increase a_r at startup

Other comments

- State and computational complexity seems fine
  - May need fixed point arithmetic in reality
- Delayed acks unhelpful at low rates but might work at high rates

A link algo

Let a packet arrive to find a virtual queue of size b.
The packet is marked with probability:

$$1 - e^{-\frac{\phi b}{s}}$$

where s is the median data packet size

- In practice $\phi \leq \frac{1}{4}$ and $b \in [0, 20s]$
- This is needed to maintain fast queue dynamics

A Gaussian traffic model

- Suppose work arrives over a time period τ is Gaussian, with mean $y\tau$ and variance $y\tau\sigma^2$
- By stability theorem the system is stable if:

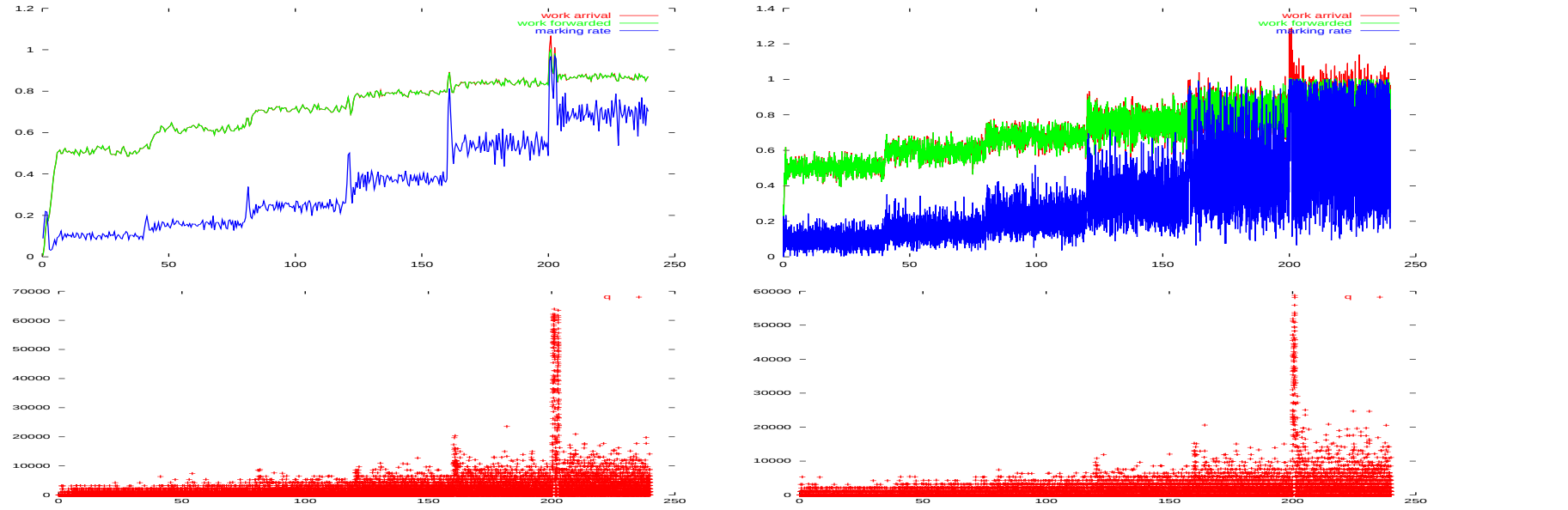
$$\frac{yp'(y)}{p(y)} = \frac{1}{1 - \frac{y}{\theta C}\left(1 - \frac{\phi\sigma^2}{2}\right)}$$

$$< \begin{cases} \frac{2}{\phi\sigma^2} & \text{if } 1 - \frac{\phi\sigma^2}{2} \geq 0 \\ 1 & \text{if } 1 - \frac{\phi\sigma^2}{2} < 0 \end{cases}$$

Traffic sensitivity

- Bootstrapping off underlying packet stochastics
- Need to ensure marking scheme is stable under a wide input set
- Actually want the sending scheme to be slightly bursty!
  - In simulation a small exponential jitter was added to the pacing timer

Uncensored results



Steady state 30Mbps, 16-512srcs, RTT Left: 256ms, Right: 16ms

Top: Util & Marking av 2RTT. Bottom: Queue 10ms samp

- Noise everywhere but it copes
- Very tight queue; hard bit is flow arrival & departure

Performance metrics

- Have used a service bound of 99th percentile under 10 packet queuing delay and no more than 0.1% loss
- Other metrics?
  - max-min queuing delay spread per 50ms, impulse convergence times, sharing metrics
- Model validation: currently an eyeball affair
  - coefficient of variance of flow rates over different timescales or Fourier transform of queue better?

Challenges

- Small windows are a real problem in theory and practice
- Striking a balance between sensitivity and noise reduction
- Rapidly fluctuating loads as seen in real networks
- Primal problems: utilization and low-delay with $> 90\%$ marking hard
  - Slow timescale adaption of AQM for utilization coming soon!
- Dual problems: sharing properties and low rates
  - Slow timescale adaption of sources for sharing on its way!