# Rate control with packet corruption

Lachlan Andrew, Caltech
with David Hayes

# Outline

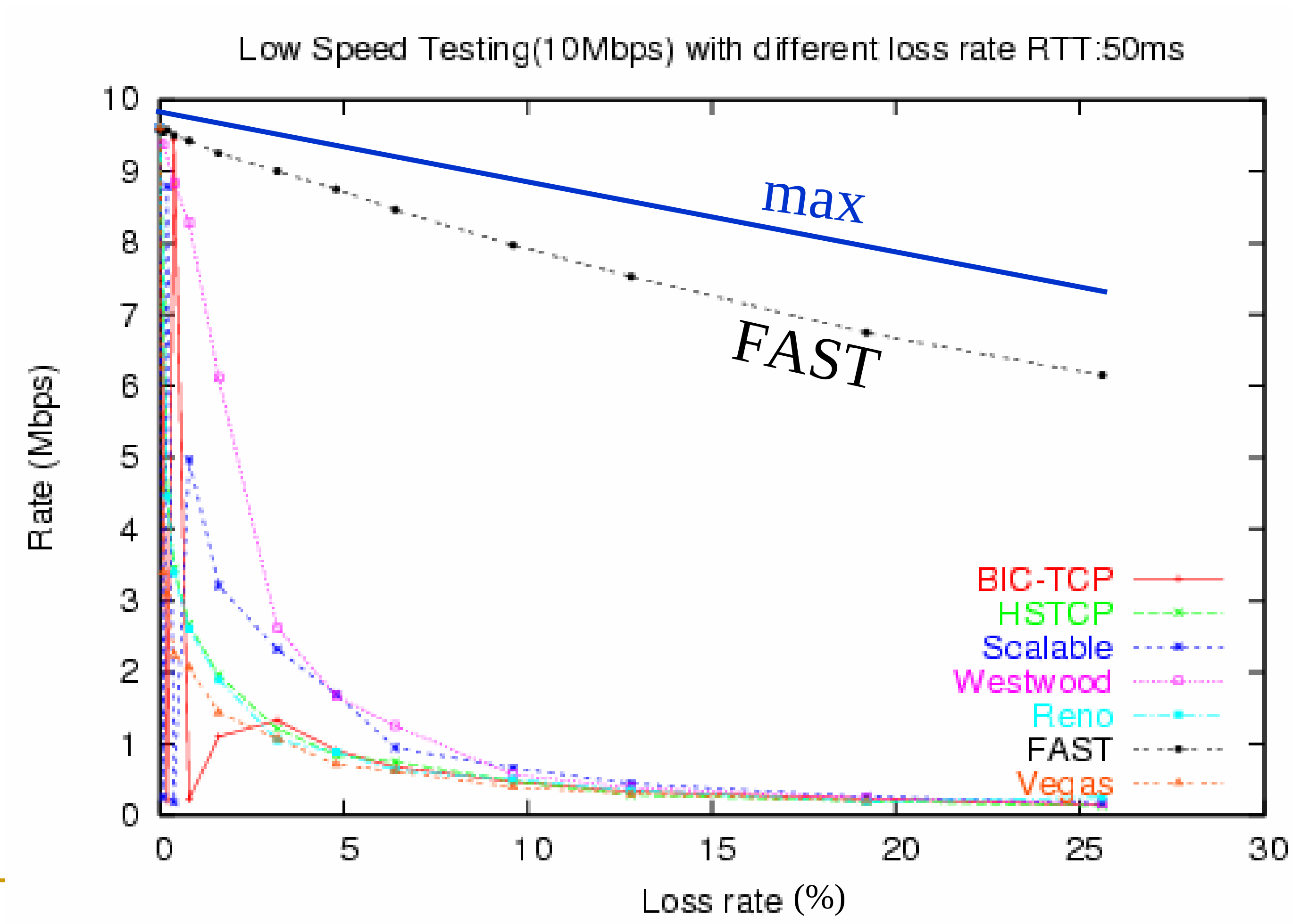
- Problem: What rate is “fair” on lossy links?
- Kelly’s Optimisation framework
- Special case: TCP
- Detecting corruption
- Redundancy

# What rate is fair on lossy links?

- ICCRG mailing list discussion on DCCP
  - How should we respond to corrupt packets?
  - TCP reduces rate; should DCCP?
- Suggestions
  - Ignore loss
  - Slow down anyway
  - Increase redundancy

- What if there is value in corrupted packets?

# Aggressiveness with very high loss




B. Wydrowski
S. Hegde
Caltech, April 2005

# Why slow down?

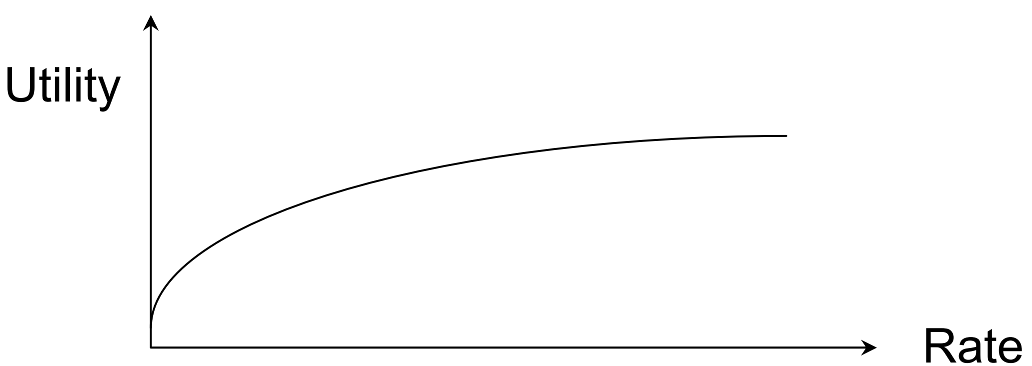
- Lost packets cause congestion before being lost
- If 99% of our packets are lost, we should send very little for network to get high overall throughput

# Increase redundancy

- If application can use corrupt data
  - More corruption $\Rightarrow$ Stronger error correcting code
- Should DCCP's rate refer to
  - Payload, before coding?
  - Raw rate, after coding?
- Corruption could *increase* raw rate
  - Desirable?

# Kelly's utility maximisation

- Best framework for fairness is economics
  - (See Bob Briscoe's talk)
- Standard theory:
  - Users get utility from instantaneous rate
  - Want maximise sum of everyone's utility



# Kelly’s utility maximisation

- Kelly/Low algorithm
  - Links measure their congestion
    - Price *p*
  - Network sums prices of links on a user’s path
    - Loss, ECN, delay, explicit
  - Sources set their rates to maximise their “net benefit” *as if* they were charged *p* per byte
- Distributed
- Fairness governed by choice of utility function

# Special case: Lossy links

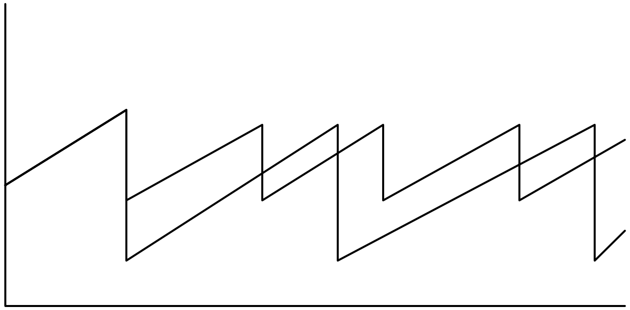
- Assumptions:
  - Corrupt packets still congest all links
    - e.g., WLAN download
  - Sources can detect “corruption loss” vs other loss
  - Utility is $U(x,\varepsilon)=U(x(1-\varepsilon),0)$
    - $x$ = total rate, including corruption
    - $\varepsilon$ = proportion corruption
    - “Benefit comes from the packets we receive correctly”

# Results

- $D(q)$ is the "response function"
  - What rate do I transmit at for congestion level $q$
- Going through the algebra gives

$$D(q,\varepsilon) = \frac{1}{1-\varepsilon} D\left(\frac{q}{1-\varepsilon}\right)$$

- Leading $1/(1-\varepsilon)$:
  - Don't count retransmission as part of the rate
- Inner factor of $1-\varepsilon$:
  - Each congestion loss must count for more



# Special case, including TCP

- Common to use $D(q) = q^{-1/\alpha}$
  - "Alpha fairness"
  - TCP: $\alpha = 2$ Proportional fairness: $\alpha = 1$
  - Max-Min $\alpha \to \infty$ Max-throughput $\alpha \to 0$

- In this case, $D(q,\varepsilon) = (1-\varepsilon)^{(1/\alpha)-1} D(q)$
  - Normal rate control mechanism (e.g., AIMD)
  - Effective window just multiple of what the mechanism calculates

# Special cases

$$\boxed{D(q,\varepsilon) = (1-\varepsilon)^{(1/\alpha)-1} D(q)}$$

- Max throughput: $\alpha \to 0$
  - Most lossy links get vanishing throughput
- Proportional fairness: $\alpha = 1$
  - No change in window – ignore loss!
- Max-Min: $\alpha \to \infty$
  - Retransmit free: window governs *new* packets
- "TCP-friendly": $\alpha = 2$
  - Slight *increase* in transmit rate for higher $\varepsilon$

# Network response

- Setting $D(q;\varepsilon) = (1-\varepsilon)D(q)$ doesn't reduce throughput by ($1$-$\varepsilon$)
- Smaller window $\Rightarrow$less traffic$\Rightarrow$smaller *q*
- Network always reduces price to create bottleneck links

# Detecting corruption

- Ideally: packet header sent with a flag
- Possible alternatives:
  - Successful packet says “I lost a burst of …”
    - How to distinguish different streams?
  - Don’t need to know *which* lost packets corrupted
    - Explicit signalling of *mean* corruption rate
  - Assume all loss is corruption
    - If main congestion signal not loss (delay, ECN)

# Redundancy

- What help can FEC be?
- Capacity of an erasure channel is $1-\varepsilon$
  - Same result as asking for retransmissions
  - Application-level decision
- What if packet not entirely erased?
  - Utility function can include some value for packets marked as corrupt
  - Burst errors mean corrupt packets usually lost

# Conclusion

- Choose flow-level properties
  - Find mechanisms to implement them
- Corruption loss *should* affect rate
- For TCP-like response functions, just scales the window
  - Up, in some cases!
  - If that's not desired, need new fairness measures