Template filename: vgdemo.frm Release 3.0

Bellcore
Bell Communications Research

# Policing Issues, and Are Peak Rates Necsssary?

Bruce Davie
July IETF, 1995

Copyright © 1995, Bellcore
All Rights Reserved

# *Policing Issues*
## *Overview*

- Definitions
- The need for policing:
  - at the edges
  - at source merge points
  - at bifurcation points
- The danger of rigid policing

.



Bellcore
Bell Communications Research

# Policing
## Definitions

- Treatment of packets that do not conform to the Tspec of the current reservation
- With token buckets, non-conformance may be due either to excess ‘average’ rate or excess ‘burstiness’
- Possible treatments:
  – Drop, delay, mark, send as best-effort

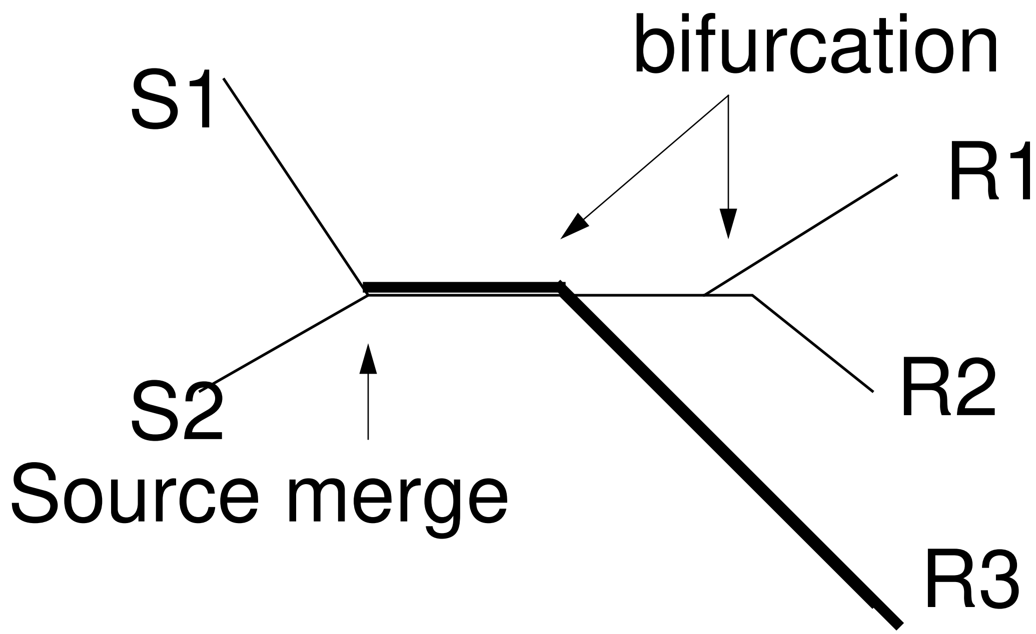
# *Why police?*

- Routers have made admission control decisions based on Tspec; admitting non-conforming traffic degrades performance of other users in class (maybe even in other classes)



Bellcore
Bell Communications Research

# *Where to Police*




.

# *Problems with non-edge policing*

- Policing is based on token bucket, which means we are policing on burstiness as well as rate
- No matter how well behaved I am, my traffic may become more bursty as it progresses through the network
- Thus a policing policy of dropping may penalize me for network behavior I don’t control - I may do worse than best effort



Bellcore
Bell Communications Research

# *Ways to police fairly*

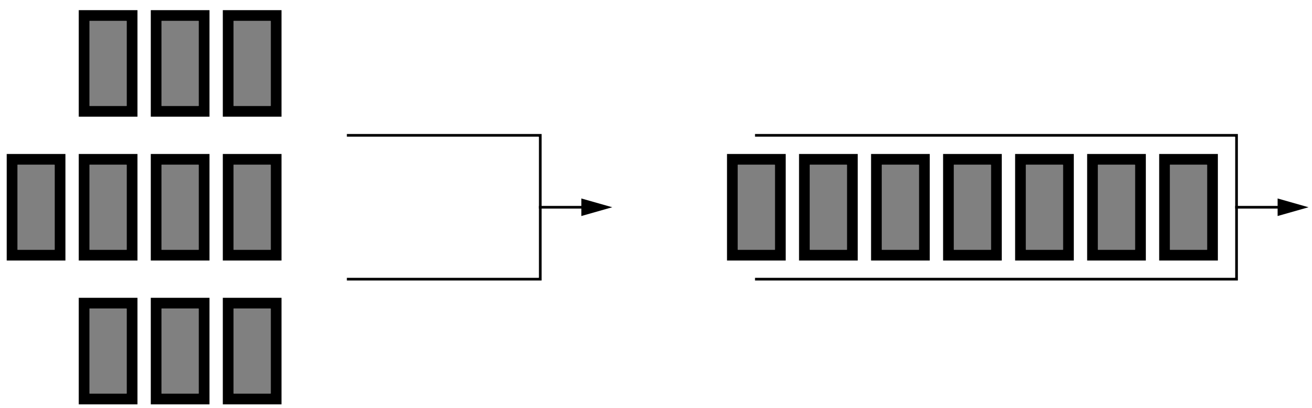
- Police on rate only
  - How do you define rate? Measure over some interval longer than B/r, but how much longer?
- Mark but don’t drop non-conforming packets
  - drop only when link is congested
  - send as best effort
- Reshape to restore conformance (adds delay)

# *Are Peak Rates Necessary?*

- Token buckets describe an ‘average’ rate (b/s) and a burst size (bytes)
- The most hostile thing a source can do is save tokens and then burst a full bucket load
- Without a peak rate spec, a full bucket could be dumped at the input line rate
- In fact, either the application or a bottleneck link may prevent full rate bursting


Bellcore
Bell Communications Research

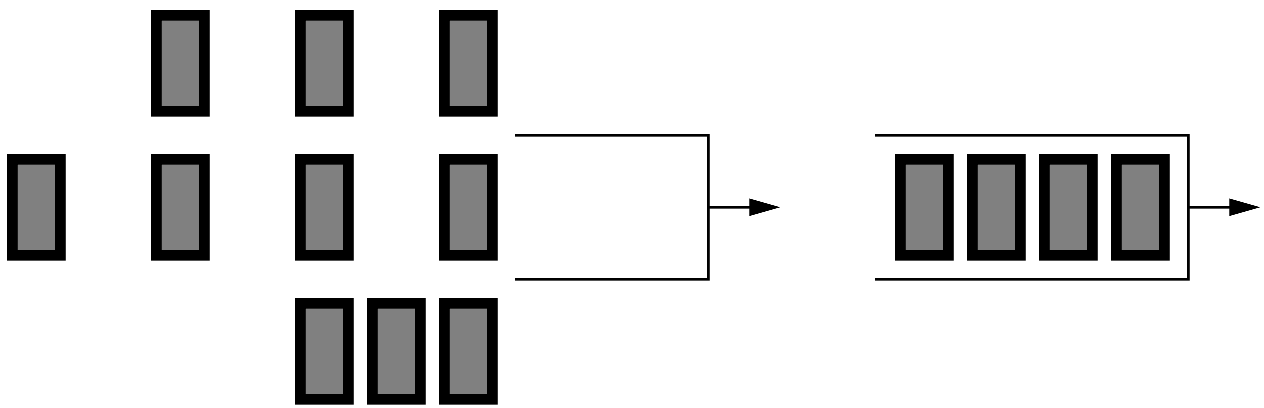
# An example





Bellcore
Bell Communications Research

# Example with peak rates





Bellcore
Bell Communications Research

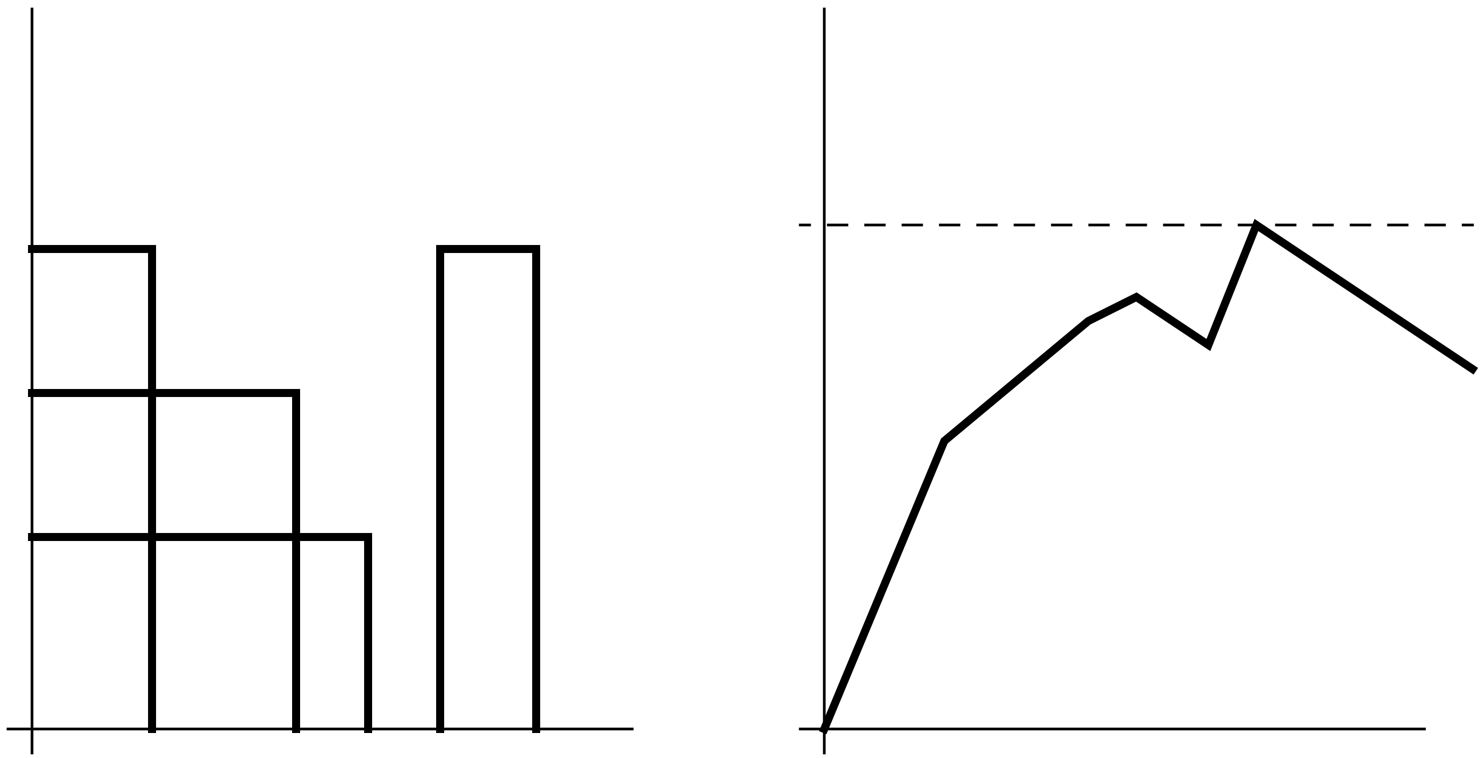
# *How to use peak rates*

- We can define peak rate by the minimum interval in which a bucketload can be injected
- I can estimate worst case queue buildup by assuming that everyone bursts at their peak rate for one bucket load, within the constraints of my incoming interfaces



Bellcore
Bell Communications Research

# *Queue length calculation*



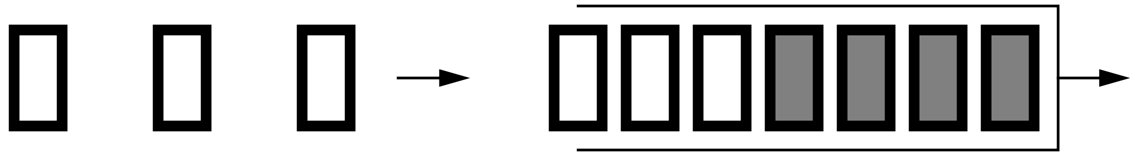


Bellcore
Bell Communications Research

# *Difficulties with peak rates*

- They change
  - get smaller at bottlenecks
  - can get larger at queuing points
- Bottleneck links can shrink the advertised peak
- In the worst case, it’s hard to ensure that peak won’t grow to link speed

# *Increasing peak rate*



.



Bellcore
Bell Communications Research

# Peak rates
## Summary

- While shrinkage of peak rate at bottlenecks is guaranteed, increase of peak is a statistical event
  - In many situations, peak is unlikely to reach link speed
- Peak rate specs may be useful to admission control, so why not provide them?