Promoting the Use of End-to-End Congestion Control in the Internet

Kevin Fall

(joint work with Sally Floyd)

Network Research Group,

Lawrence Berkeley National Laboratory

Berkeley, CA

http://www-nrg.ee.lbl.gov/{kfall,floyd}

Scope and Outline

- *Scope:* **best-effort traffic**
- **Issues to be covered:**
  - Why the need for congestion control?
  - Handling not-well-behaved flows
  - Some simulation results
  - Issues and future work

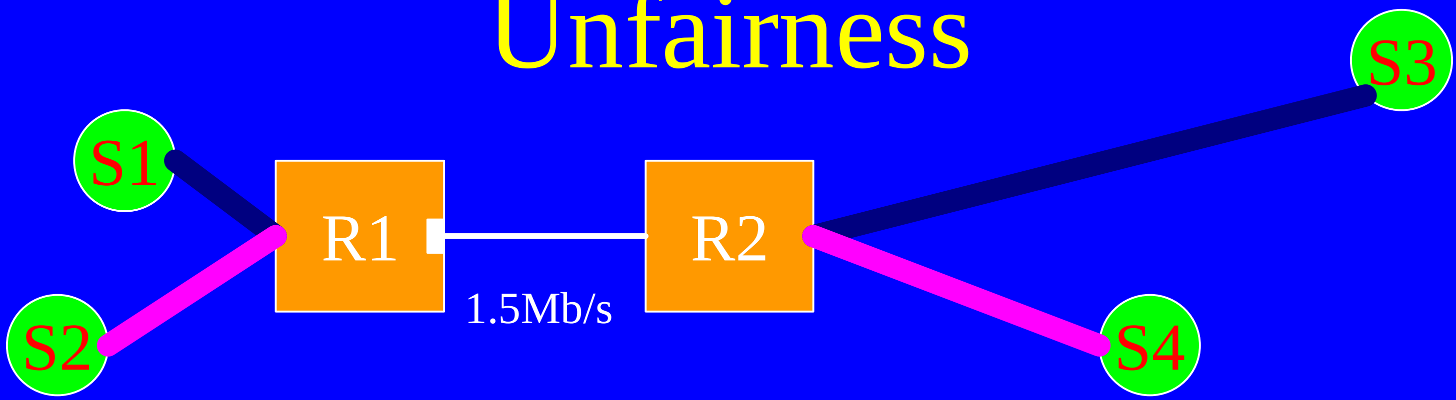
Internet Robustness

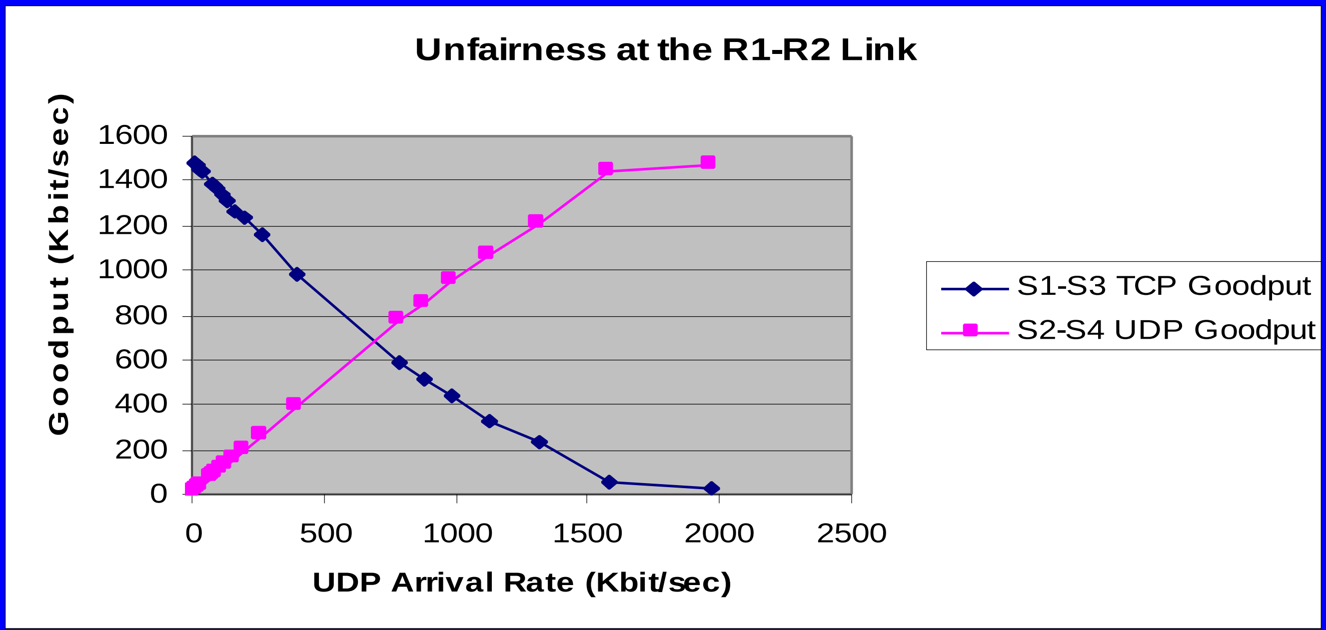
- Reliance on good behavior of endpoints
- Properly-implemented TCP a key component
- Most traffic today is TCP
- Historically, a closely-knit user/developer/research community

Threats to Internet Robustness

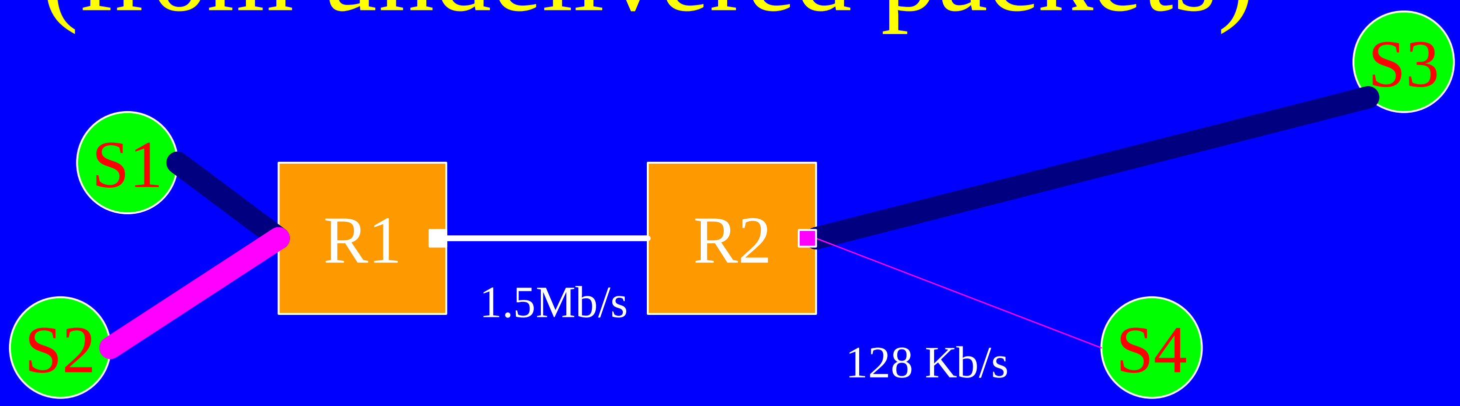
- Malicious or buggy TCPs
- New Applications lacking congestion control (e.g. UDP-based multimedia)
- Unbalanced incentive structure (which does not directly penalize bandwidth hogs)
- <u>May result in Unfairness and Congestion Collapse</u>

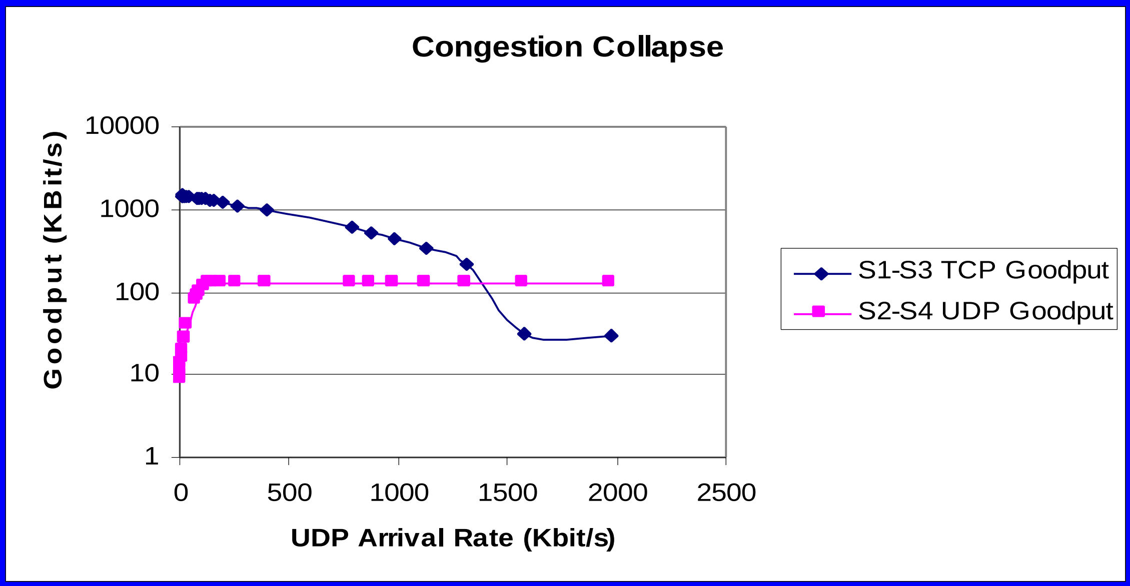
Unfairness


S1
S2
R1
R2
S3
S4
1.5Mb/s


Unfairness at the R1-R2 Link
Goodput (Kbit/sec)
1600
1400
1200
1000
800
600
400
200
0
0
500
1000
1500
2000
2500
UDP Arrival Rate (Kbit/sec)
S1-S3 TCP Goodput
S2-S4 UDP Goodput

Congestion Collapse (from undelivered packets)

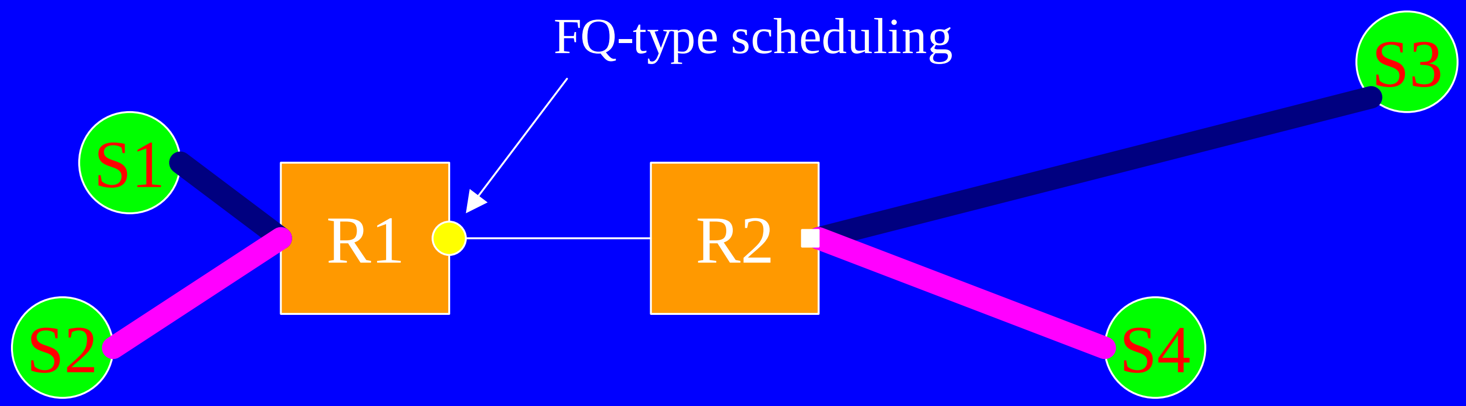


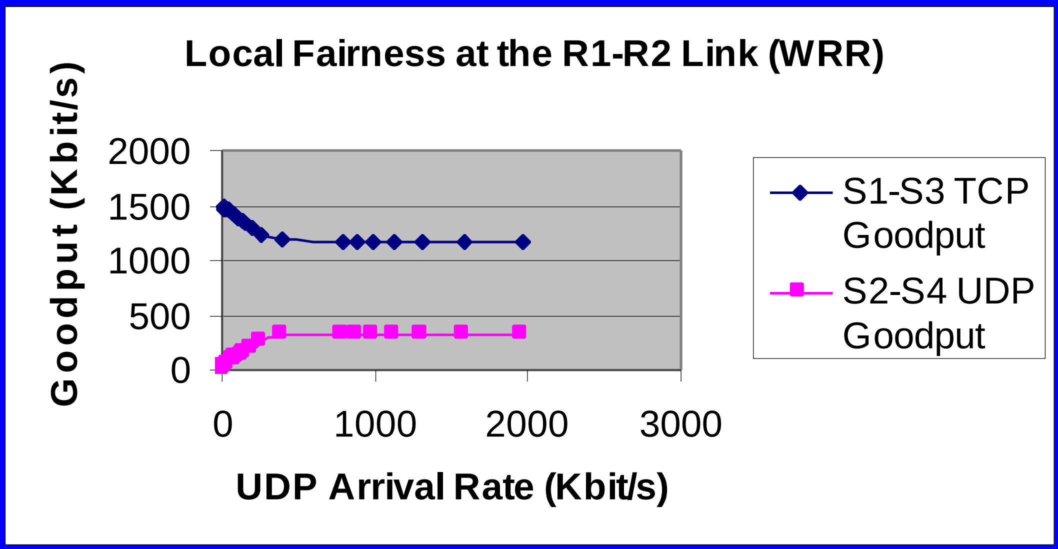


Possible Countermeasures

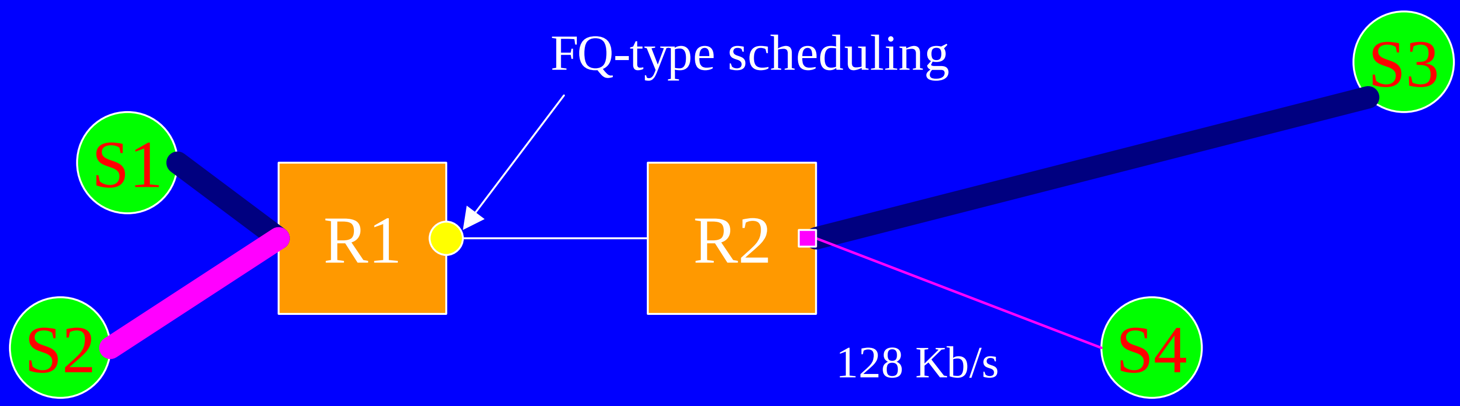
- per-flow scheduling mechanisms
  - (e.g. FQ, RR, and variants)
- volume-based pricing
- congestion mechanisms in routers
- *these are not necessarily mutually exclusive*

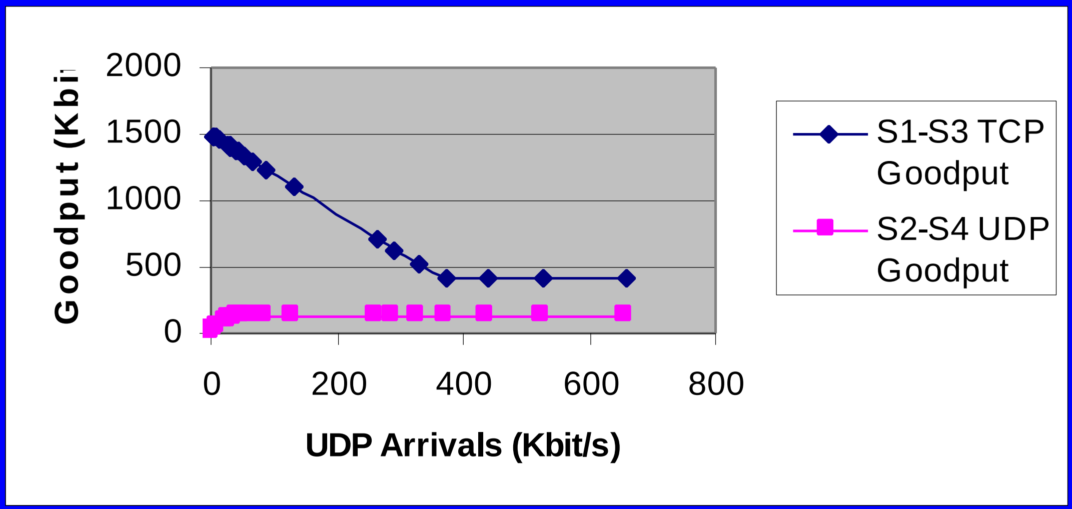
Fairness with FQ/RR Scheduling





Congestion Collapse with FQ/RR





Incentives of Approaches

- per-flow scheduling
  - loss-tolerant *fire-hose* applications not discouraged
  - uniform treatment of all flows
- pricing
  - may discourage congestion, but no assurance
- router mechanisms
  - by penalizing non-reactive flows, encourages congestion control/adaptation

Router Mechanisms

- Look for high-bandwidth and non-adaptive flows during times of congestion
- Reduce service of ill-behaved flows
- Increase service of penalized flows if they become well-behaved
- *Upshot:* encourages use of congestion-control in protocols and applications at endpoints to avoid degraded service

Detecting “Bad” Flows

- The “TCP-friendly” test:
  - does the flow bandwidth exceed the rate of an aggressive TCP in comparable circumstances?
- The “responsive” test:
  - does the flow reduce its arrival rate in response to an increase in the packet drop rate?
- The “disproportionate-bandwidth” test:
  - does the flow use *significantly more* than its “fair share” of the link bandwidth when there is likely to be suppressed demand?

The "TCP-Friendly" Test

- A flow is not "TCP-friendly" if its rate exceeds a multiple of:

$$1.5\sqrt{2/3}\left(\frac{B}{R\sqrt{p}}\right)$$

B – packet size
R – path round-trip time
p – packet drop probability

- Requires:
  - upper bound for the packet size (e.g. link MTU)
  - lower bound for the connection RTT (e.g. useful if link prop delay is a significant portion of RTT)
  - overall count of packet drop rate (simple in router)

The "Unresponsive" Test

- TCP throughput equation suggests a relationship between packet drop rate and flow arrival rate:

$$T \sim \frac{1}{\sqrt{p}}$$

- Example: increase of drop rate by 4x should result in arrival decrease of 2x
- Requires estimates of flow arrivals and drops over long time scales; difficulties with variable demand

The "Disproportionate" Test

- a flow is using disproportionate bandwidth if, for n flows present, it uses much more than $(1/n)$ share of the link **and** there is likely to be more demand
- so, look for flows that use much more bandwidth than others:

$$\boxed{\frac{\log(3n)}{n}}$$

Stays above $(1/n)$ for increasing n

Perspective on Tests

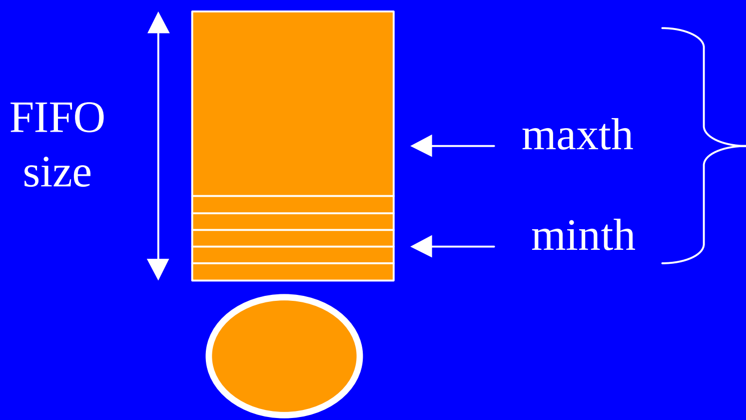
- Prototypical representatives for tests
  - future work needed
  - implemented in simulation
- Coarse grained
  - does not attempt to impose local “fairness”
  - attempts to regulate egregious bandwidth hogs
- *Issue:* How to measure arrival/drop statistics?

Measuring Arrival/Drop Statistics

- need per-flow estimates of arrivals/drops?
- per-flow counters are expensive
- Idea:
  - using RED queue, count per-flow packet drops
  - drops will be proportional to flow's arrival rate
  - only care about candidate "bad" flows in times of congestion

RED (Random Early Detection)

- Active buffer management technique
- Manages underlying (FIFO) queue:



$\bar{q}$ < minth (queue arriving packet)

minth < $\bar{q}$ < maxth (drop with proportional probability)

$\bar{q}$ > maxth (drop packet)

- A flow's portion of the dropped packets is roughly equal to its portion of the aggregate arrivals

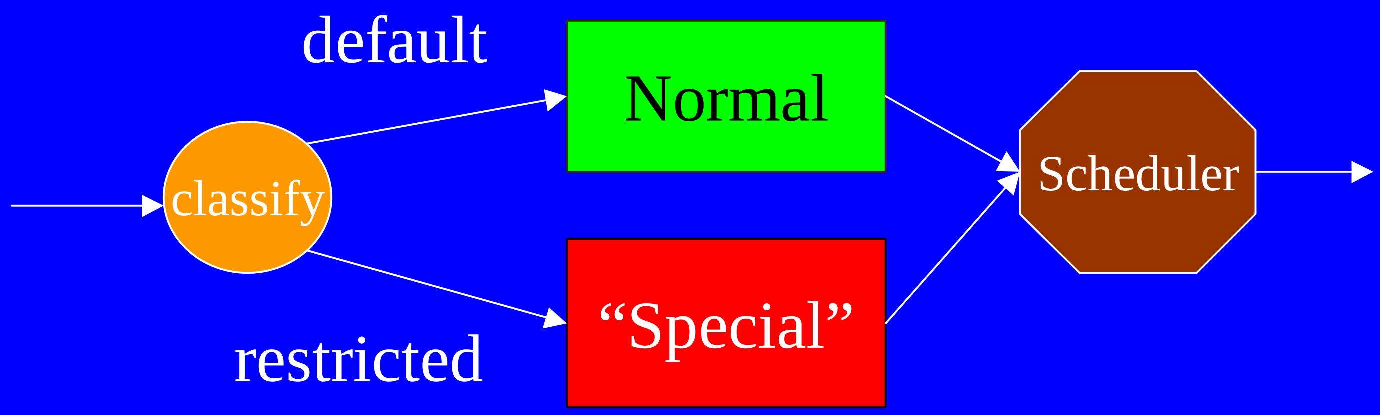
Packet Drops with RED

- RED can drop packets in two ways
  - when the average queue size exceeds **minthresh** and is less than **maxthresh** (an “unforced” drop)
  - when the underlying FIFO overflows or maxthresh is exceeded (a “forced” drop)
  - unforced drops should dominate for traffic mixes using end-to-end congestion control

Drop Metrics

- Packet Drop Metric
  - ratio of flow's dropped packets to total dropped packets
  - good estimate of flow's arrival rate for *unforced* drops
- Byte Drop Metric
  - ratio of flow's dropped bytes to total dropped bytes
  - good estimate of flow's arrival rate for *forced* drops
- Combined Drop Metric
  - weighted average of packet and drop metrics
  - good overall arrival estimate over some interval

Router Mechanisms



- Apply tests using bw estimate from RED drops
- Regulate by adjusting classifier and scheduler
- Re-adjust based on future dynamics

Flow Regulation

- **need some way of restricting bandwidth of a flow**
- *Packet scheduling*:
  - simple priority
  - WFQ, WRR
  - CBQ
- *Priority drop*:
  - variants of FRED (fair RED) [SIGCOMM97]

Requirements Summary

- flow classifier
  - determines whether a flow is to be restricted
- RED queues
  - gives drops in proportion to arrival rate
- flow-based drop analyzer
  - accounts for drops based on flow
- analysis/policy machinery
  - determines when and how to adjust scheduler and flow classifier
- priority-capable scheduler or drop mechanism

Simulations using NSv2 (simulator base for VINT project)

- USC/ISI: Deborah Estrin, Mark Handley, John Heideman, Ahmed Helmy, Polly Huang, Satish Kumar, Kannan Varadhan, Daniel Zappala
- LBNL: Kevin Fall, Sally Floyd
- UCBerkeley: Elan Amir, Steven McCanne
- Xerox PARC: Lee Breslau, Scott Shenker
- **VINT is currently funded by DARPA through mid-1999**

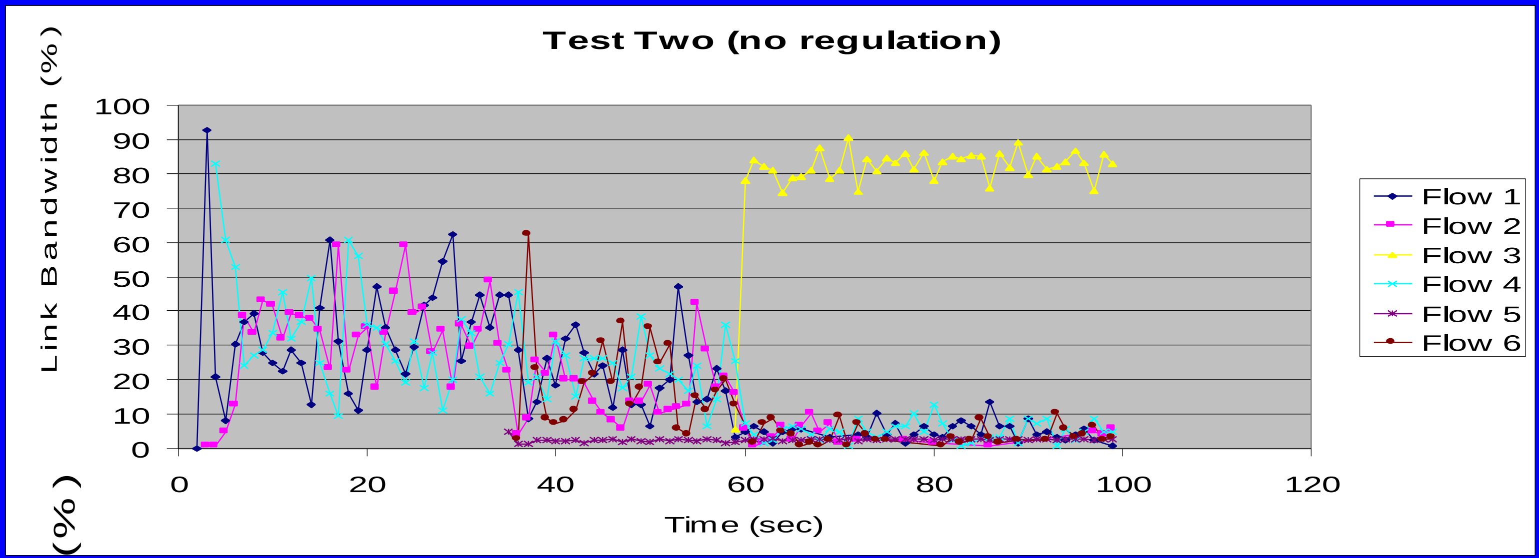
The VINT Simulation Environment

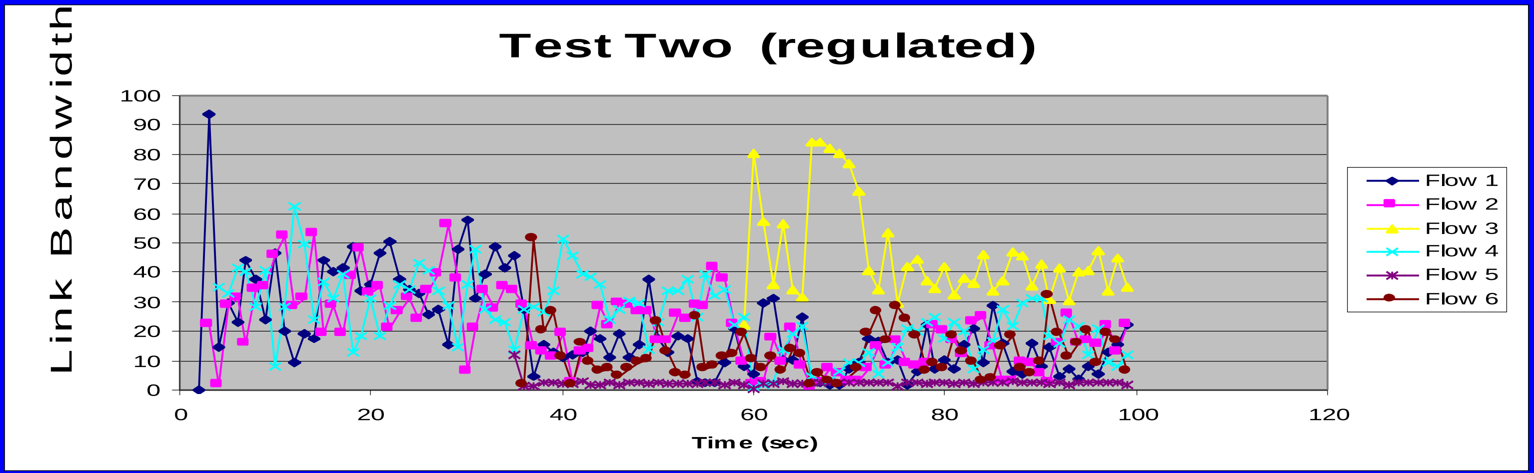
- Components: **ns2** and **nam**
- **NS2** (network simulator, version 2):
  - Discrete-event C++ simulation engine
    - scheduling, timers, packets
  - Split Otcl/C++ object "library"
    - protocol agents, links, nodes, classifiers, routing, error generators, traces, queuing, math support (random variables, integrals, etc)
- **Nam** (network animator)
  - Tcl/Tk application for animating simulator traces
- available on UNIX and Windows 95/NT

NS Supported Components

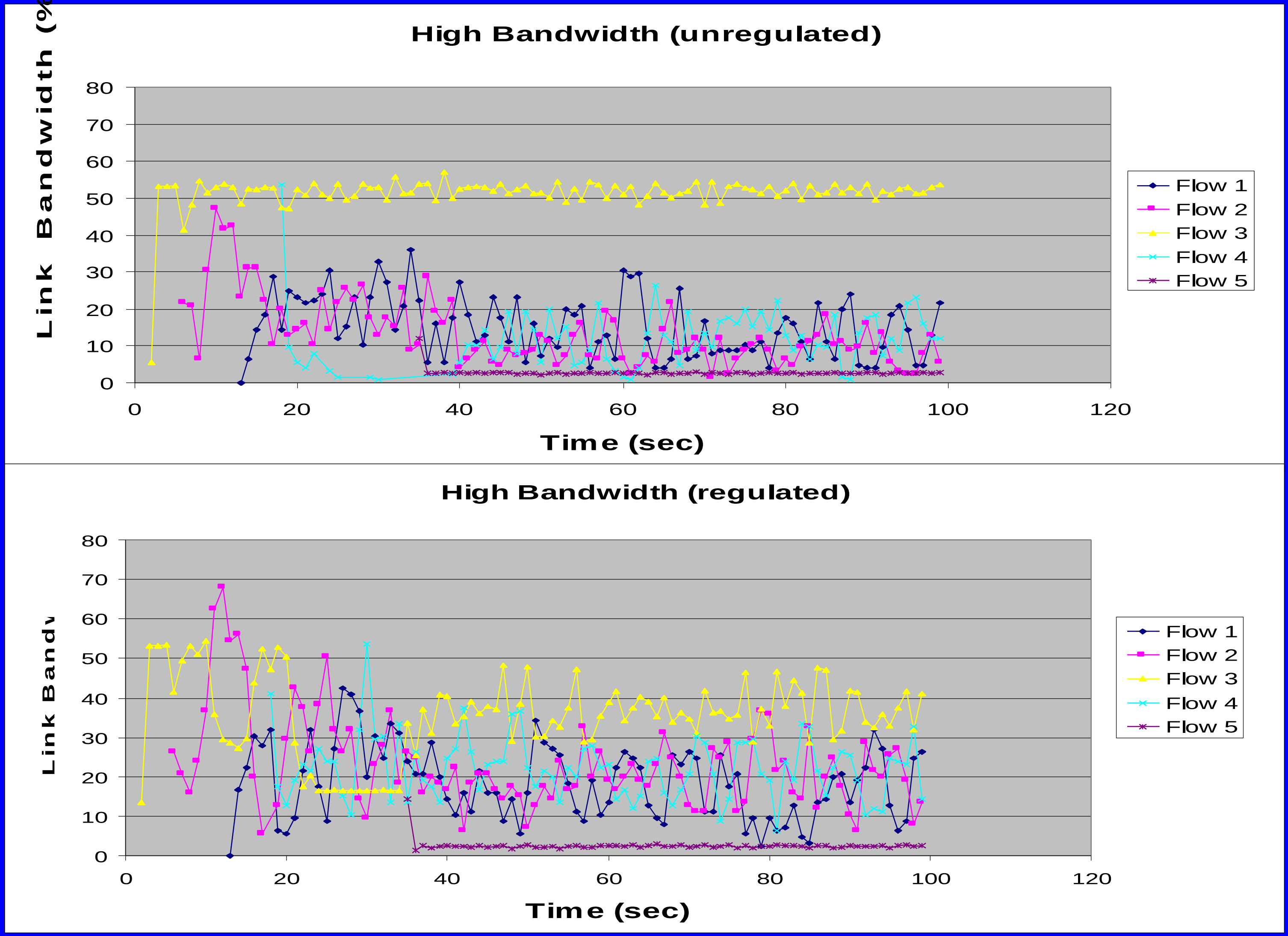
- Protocols:
  - TCP (2modes + variants),UDP, IP, RTP/RTCP, SRM, 802.3 MAC, 802.11 MAC
- Routing
  - static unicast, dynamic unicast (distance-vector), multicast
- Queuing and packet scheduling
  - FIFO/drop-tail, RED, CBQ, WRR, DRR, SFQ
- Topology: nodes, links Failures: link errors/failures
- Emulation: interface to a live network

Example: TCP-Friendly Regulation





Example: High Bandwidth Regulation



Conclusions

- Growing concern over non-congestion-controlled Internet traffic
- Several possible approaches, but want incentive structure that rewards good behavior
- Router mechanisms are a step toward this goal, but much to be done (e.g. exact nature of tests, choice of scheduler/drop management, domain of applicability)

Issues and Future Work

- Issues with implementation
  - configured RTT lower bound in TCP-friendly test limits usefulness
  - TCP model is loose
  - detecting unresponsive flows is tricky in variable-demand environments
  - choice of scheduling/drop mechanism
- Issues with policy
  - which flows to punish, and by how much?

Additional Information

- Router Mechanisms Page
  - http://www-nrg.ee.lbl.gov/floyd/end2end-paper.html
- Vint and NS Pages
  - http://www-mash.cs.berkeley.edu/ns
  - http://netweb.usc.edu/vint
  - http://www.ito.darpa.mil/Summaries97/E243_0.html
- majordomo@mash.cs.berkeley.edu
  - "subscribe ns-users"