

Using Differentiated Services to Improve Protocol and Application Performance

Venkata N. Padmanabhan

Microsoft Research

<http://www.research.microsoft.com/~padmanab>

June 1999

Outline

- ★ Differentiated services (*diffserv*) overview
- ★ Opportunity for protocols and applications
 - ★ Example #1: TCP fast start
 - ★ Example #2: Asymmetric networks
- ★ Issues and concerns
- ★ Related Work
- ★ Summary and Future Work

Differentiated Services Overview

- ★ Lightweight mechanisms for service differentiation in the Internet
- ★ Diffserv bits invoke per-hop behavior
 - ★ examples: priority dropping, priority scheduling
- ★ Avoids expensive state lookup
- ★ Signaling on “administrative” time scale
- ★ Focus on coarse-grained differentiation
 - ★ example: individual ISP customer

Opportunity for Protocols and Applications

- ★ Packet-level differentiation could enable end host to exercise control at granularity finer than RTT
- ★ End host can exploit context-specific protocol/application knowledge
- ★ Potential for significant performance win over pure end-to-end schemes

Transport Protocols

- ★ Improving connection startup performance
 - ★ use cached congestion information safely
 - TCP fast start
 - ★ decrease frequency of timeouts
 - high drop priority to SYN and other initial packets
- ★ Alternate congestion response
 - ★ initial congestion signal \Rightarrow start marking packets
 - ★ sustained congestion \Rightarrow shrink window
- ★ Bidirectional traffic in asymmetric networks

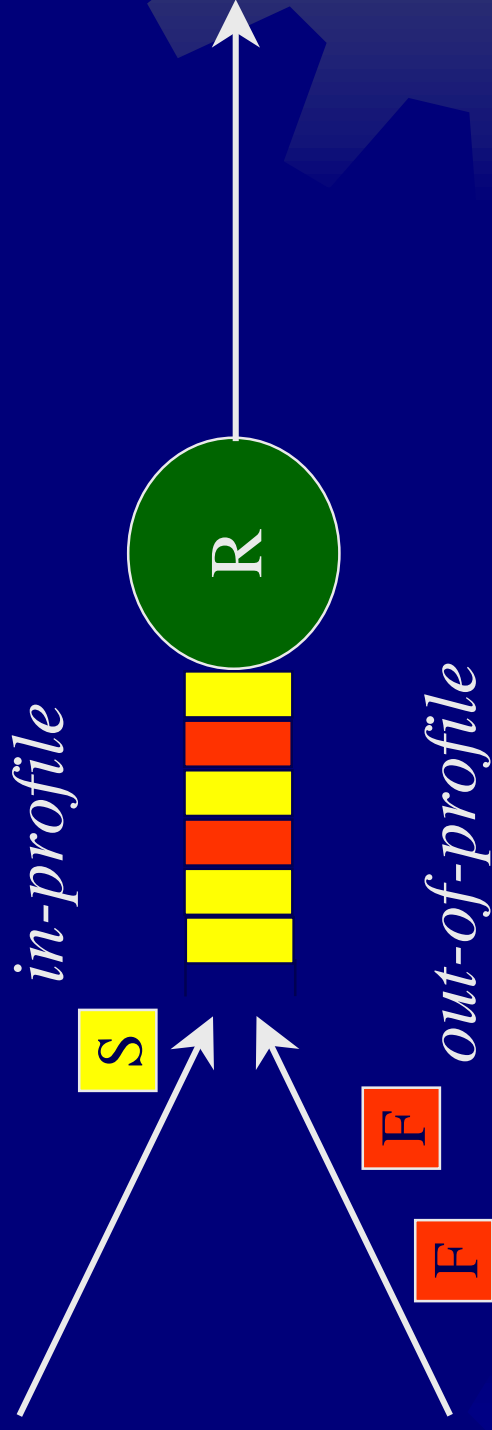
Applications

- ★ Adapting dynamically to user actions in a Web browsing session
 - ★ Multiple concurrent data streams
 - ★ TCP session [Pad98] addresses e2e problem
 - ★ Priority scheduling can address queuing problem
- ★ Web prefetching
 - ★ lower drop/scheduling priority for packets of prefetch connections

Example #1: TCP Fast Start

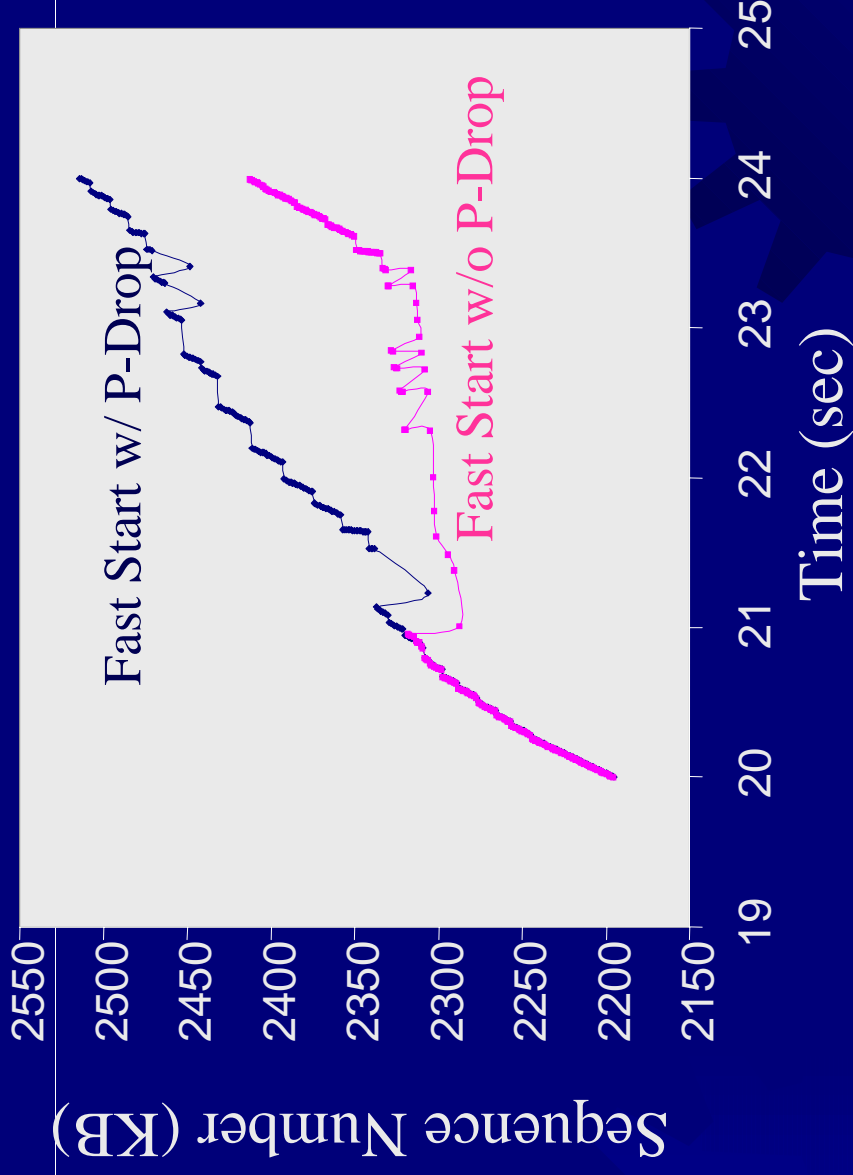
- ★ Short connections have little opportunity to probe network for bandwidth
- ★ Reusing cached congestion info *may* help
- ★ **Problem:** cached info could be stale
 - ★ for instance, due to flash crowds
- ★ **Solution:** use priority dropping to shield “normal” packets
- ★ Enables expedited recovery upon pkt loss

TCP Fast Start Dynamics



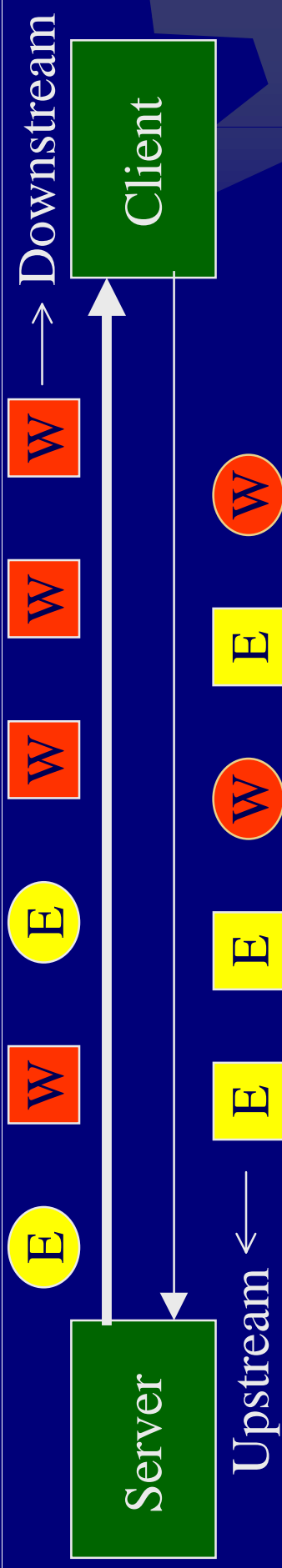
- ★ Upon congestion, fast start packets are dropped preferentially

Impact on Competing Traffic



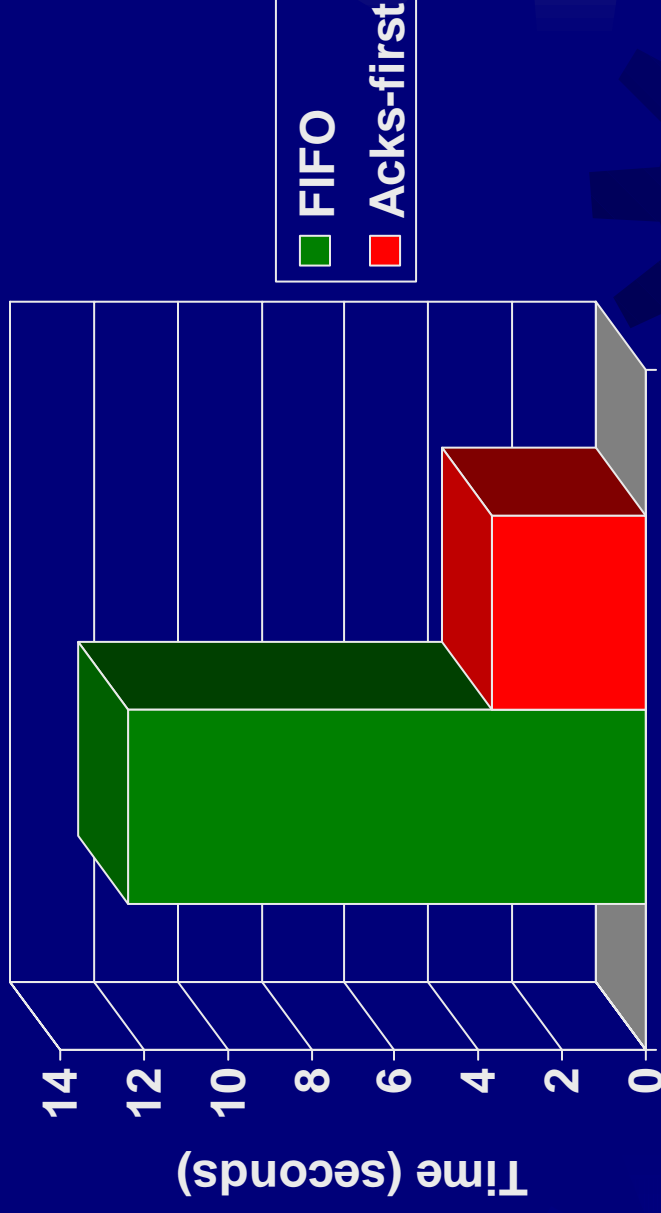
- ★ Priority dropping reduces adverse impact of stale congestion information

Example #2: Asymmetric Networks



- ★ **Problem:** data pkts of upstream connection block acks of downstream connection
- ★ **Solution:** acks-first scheduling upstream
 - ★ use in conjunction with ack congestion control

Acks-first Scheduling: Performance



- ★ 10 Mbps downstream, 28.8 Kbps upstream
- ★ 40 KB Web page download

Issues and Concerns

- ★ Ideally we would like 3 levels of service
 - ★ normal, better than normal, worse than normal
- ★ Lack of end-to-end support for diffserv
 - ★ no worse than best-effort in some cases
 - ★ need support at least at bottleneck in other cases
- ★ Disincentive for good behavior
 - ★ not fundamentally different from best-effort
 - ★ makes policing more challenging

Related work

- ★ Blitz protocol [Kim95]
 - ★ non-feedback-based congestion control for high-speed networks
- ★ RED extensions (probabilistic differentiation)
 - ★ Enhanced RED [FKS+97], RED In/Out [CW97]
- ★ Priority encoded transmission [ABE+94]
 - ★ MPEG I-frames assigned higher priority than P and B frames

Summary and Future Work

- ★ Significant win from combination of diffserv and protocol/application-specific knowledge
- ★ Two specific examples:
 - ★ TCP fast start to improve startup performance
 - ★ Acks-first scheduling in asymmetric networks
- ★ Future work
 - ★ evaluate other possible uses of diffserv
 - ★ study impact of partial diffserv support and/or packet re-marking