The War Between Mice and Elephants

- Liang Guo
- Ibrahim Matta
- Computer Science Department
- Boston University

- Presented by:
- Chris Gianfrancesco and Rick Skowyra

Overview

- Introduction
- Analyzing Short TCP Flow Performance
- Proposed Scheme
- Simulation
- Discussion
- Conclusions

Mice and Elephants

- Elephants: Long TCP connections
  - File downloads
  - Large portion of traffic
  - Small portion of connections (users)
- Mice: Short TCP connections
  - Web browsing
  - Small portion of traffic
  - Large portion of connections
  - Decreased performance when network utilization is high

TCP Issues

- Conservative startup
  - Minimal initial sending window
  - Large ITO before data can be gathered for RTO
- Congestion hurts short flows
  - Any packet loss likely results in timeout
  - Long flows can benefit from fast retransmit
- Because of these factors, long TCP flows handle network congestion better than short flows

Proposal

- TCP protocol and current queueing policies do nothing to help the performance of short flows
- Implement a Diffserv architecture
  - Short flows are given preferential treatment
- Hypothesis: Short flows can be given additional resources to complete faster, with a minimal impact on the performance of long flows

Diffserv and RIO

- Differentiated Services
  - Offer preferential treatment to a certain class of traffic that is more important
  - In this case, use Diffserv to improve performance of short TCP flows, while trying to minimize impact on long flows
- RIO: RED with In and Out
  - Packets have a bit to mark them as “in” or “out”
  - RED algorithm with different parameters for in and out packets

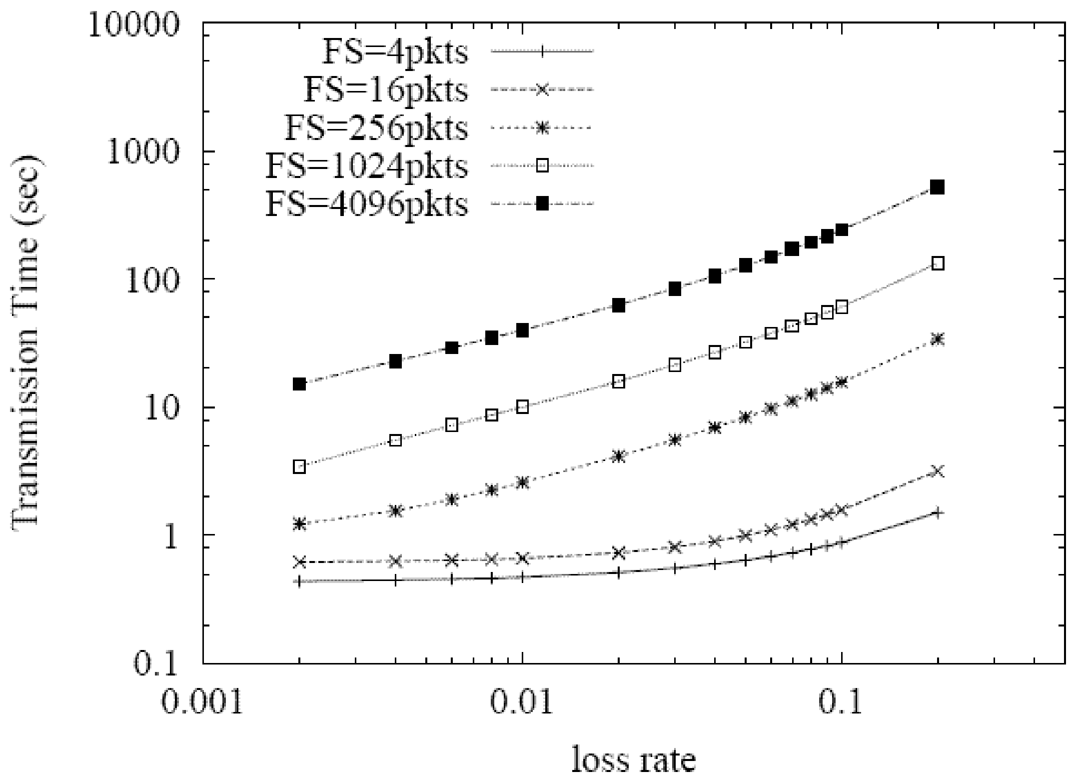
Overview

- Introduction
- Analyzing Short TCP Flow Performance
- Proposed Scheme
- Simulation
- Discussion
- Conclusions

Short TCP Flow Analysis

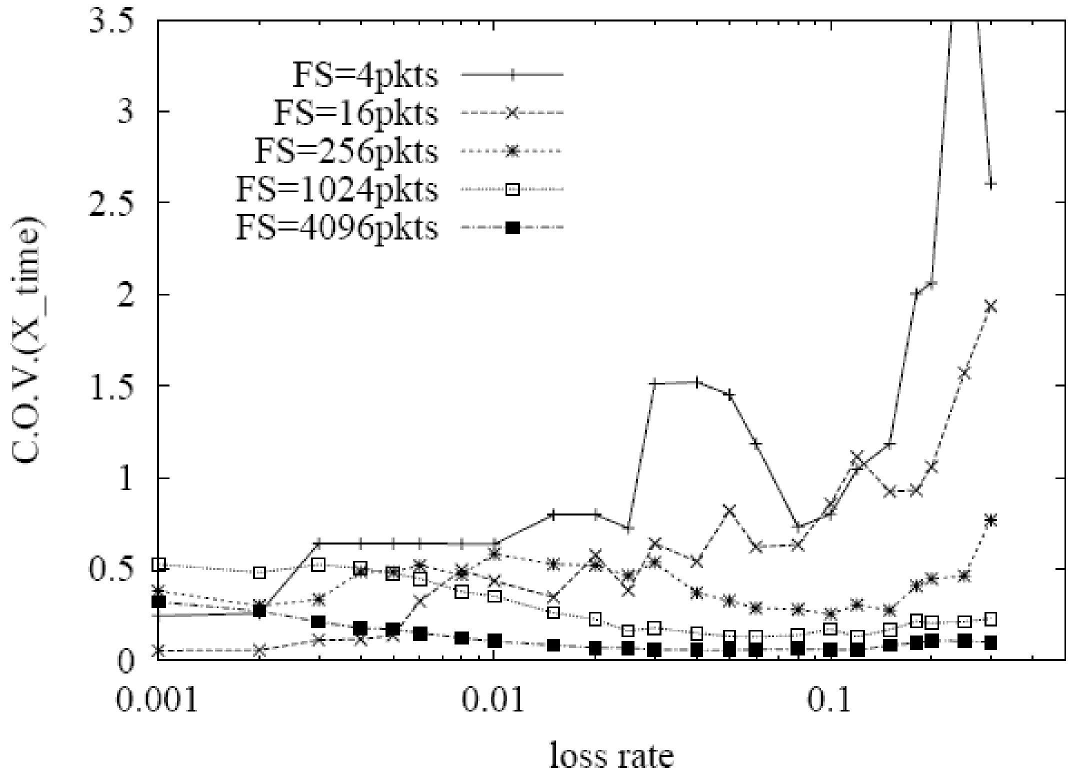
- To determine how best to help short TCP flows, first find what the major factors of poor short flow performance are
- Main contributor is loss rate: as described before, loss of packets in a short flow impacts performance much more than in a long flow

Short TCP Flow Analysis



(a) Average Transmission Time

Short TCP Flow Analysis



(b) Coefficient of Variation

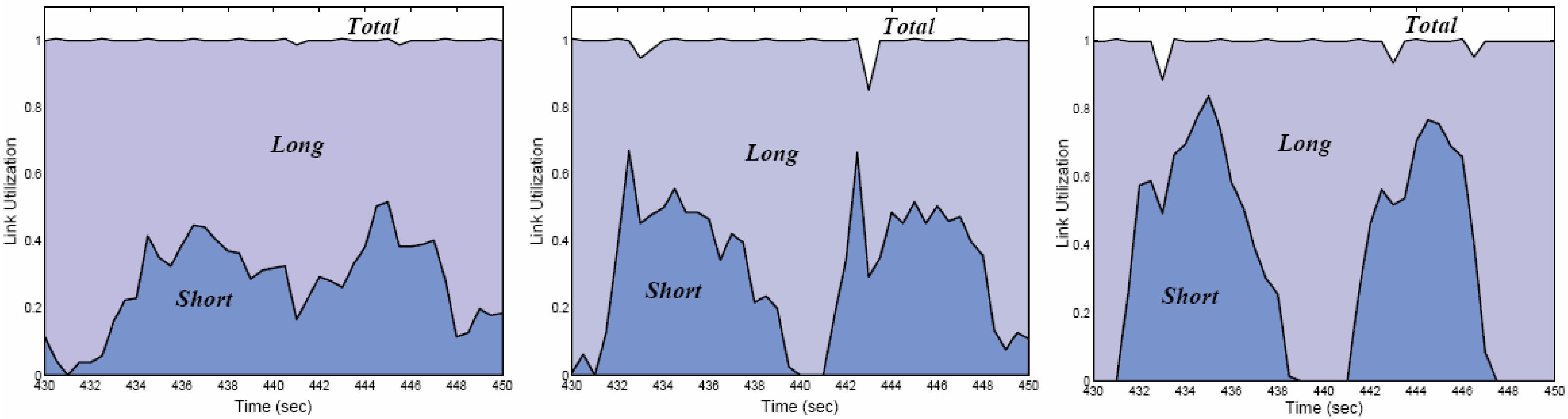
Short TCP Flow Analysis

- Conclusions: As loss rate increases, both transmission time and variability of transmission time for short flows greatly increase
- Packet loss for short TCP flows must be controlled in order to provide more reliable and faster service, with higher fairness relative to long flows

Preferential Treatment: Simulation

- Use ns simulator to test the effect of a queueing strategy with preferential treatment
- Two sets of TCP-Newreno flows competing for a congested 1.25Mbps link:
  - Short (100 packet) flow x 10
  - Long (10000 packet) flow x 10
- Observe network characteristics with Drop Tail, RED, and RIO-PS
  - RIO-PS: RIO with Preferential treatment to Short flows

Simulation Results



Simulation Results

- Conclusions: Preferential treatment can be given without hurting network goodput
- RIO-PS can offer better performance for short TCP flows at a congested link

Link B/W	Flows	DropTail	RED	RIO-PS
1.25Mbps	All	153479	154269	154486
	Short	40973	49897	49945
	Long	112506	104372	104541
1.5Mbps	All	185650	184315	183154
	Short	43854	49990	49990
	Long	141796	134325	133164

TABLE I

NETWORK GOODPUT UNDER DIFFERENT SCHEMES

Overview

- Introduction
- Analyzing Short TCP Flow Performance
- Proposed Scheme
- Simulation
- Discussion
- Conclusions

Proposed Architecture

- Based on Diffserv (Differential Services)
- Routers in a network are classified as 'edge' or 'core'
  - Per-flow operations performed only on edges
  - Per-class operations performed in the core

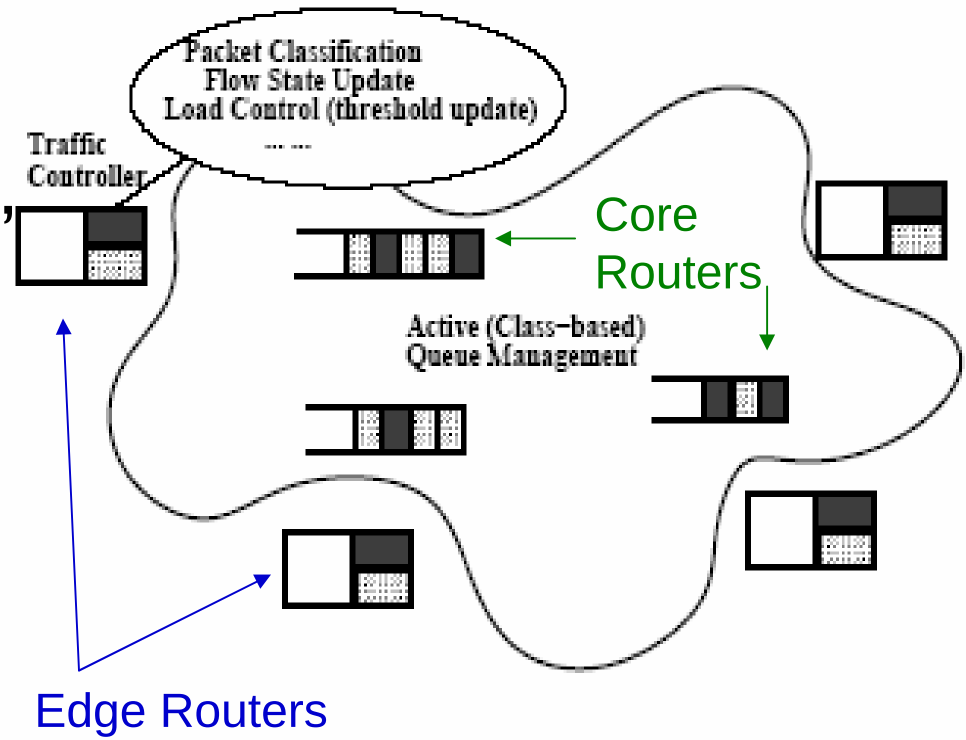


Fig. 3. Proposed Architecture

Edge Routers

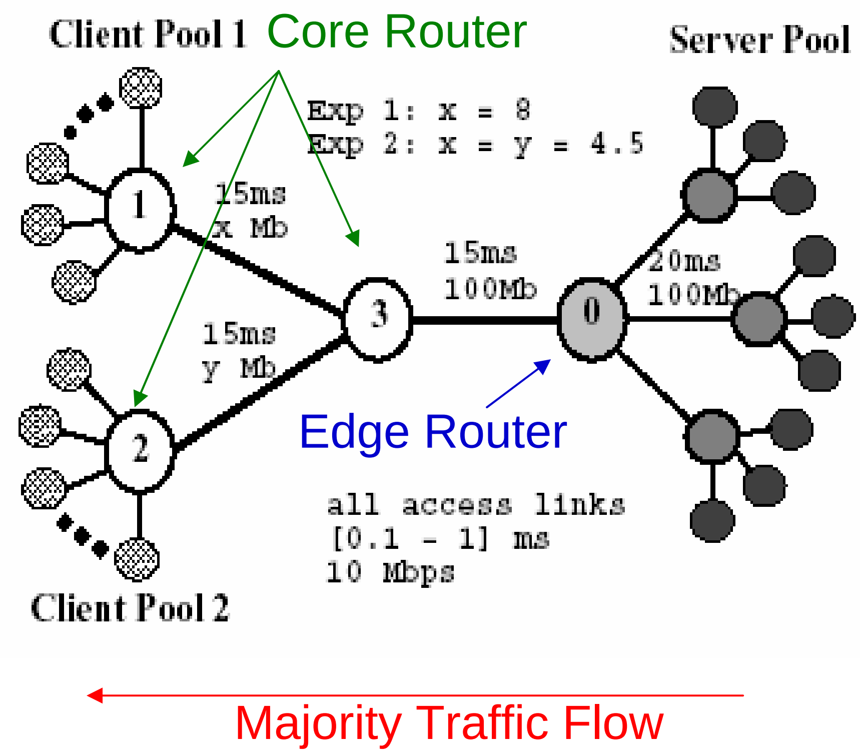
- Determine and label whether a packet belongs to a long or short flow
- Threshold-based approximation
  - L_t – Number of packets beyond which a flow is considered long
    - Dynamic
- Parameters
  - T_u – Time units until a flow is considered terminated
  - SLR – Ratio of active (short) flows to long flows
  - T_c – Intervals between updates of L_t
- All flows initially labeled as short

Core Routers

- Implemented with RIO queues
  - Only one queue per router
    - No packet reordering
  - Preferential treatment given to short flows
    - Drop probability for short-flow packets is not affected by arrival of long-flow packets
    - Drop probability for long-flow packets is affected by arrival of short-flow packets

Simulation

- Web-like TCP flows
- HTTP 1.0
- Clients request a webpage, servers respond
- Load within 90% of bottleneck link capacity



Web Traffic Configuration

Name	inter-page	objs/page	inter-obj	obj size
Exp1 client 1	Exponential mean 9.5	Uniform min 2 max 7	Exponential mean 0.05	Bounded Pareto [4,200000] shape 1.2
Exp2 client 1	Exponential mean 9.5	Uniform min 2 max 7	Exponential mean 0.05	Bounded Pareto [4,500] shape 1.2
client 2	Exponential mean 192	Uniform min 1 max 3	Exponential mean 1.5	Bounded Pareto [400,200000] shape 1.2

- Randomly selected clients surf web pages of different sizes from randomly selected web sites.
  - Web pages may have multiple objects
  - Each object requires a new connection

Simulation Parameters

Description	Value
Packet Size	500 bytes
Maximum Window	128 packets
TCP version	Newreno
TCP timeout Granularity	0.1 seconds
Initial Retransmission Timer	3.0 seconds
B/W delay product (BDP)	$\approx$ 200 pkts (Exp1) $\approx$ 120 pkts (Exp2)
Bottleneck Buffer Size (B)	DropTail: 1.5$\times$ BDP RED/RIO-PS: 2.5$\times$BDP
Q. Parameters	$(min_{th}, max_{th}, P_{max}, w_q)$
RED	(0.15B, 0.5B, 1/10, 1/512)
RIO-PS short	(0.15B, 0.35B, 1/20, 1/512)
RIO-PS long	(0.15B, 0.5B, 1/10, 1/512)
RED & RIO-PS	ecn_ on, wait_ on, gentle_ on
Edge Router	$SLR = 3, T_u = 1\ sec, T_c = 10\ sec$
Foreground Traffic	
(Src, Dest)	(Server Pool, Client Pool)
Long Connection Size	1000 packets
Short Connection Size	10 packets

Overview

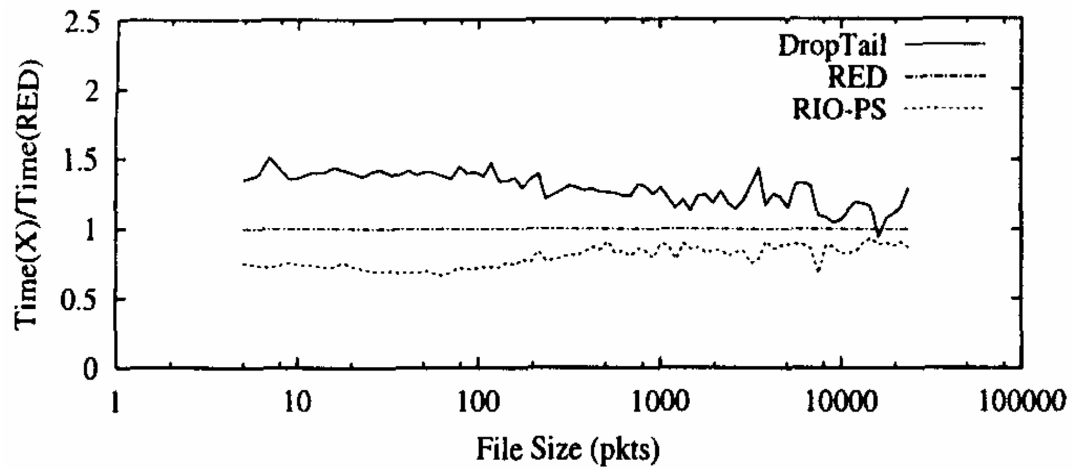
- Introduction
- Analyzing Short TCP Flow Performance
- Proposed Scheme
- Simulation
- Discussion
- Conclusions

Experiment 1: Single Client Set

- Only one client pool used
- Strategies:
  - Drop-Tail
  - RED (ECN)
  - RIO-PS
- 4000sec
  - 2000sec start-up
- Initial Time-Out value controversy

Experiment 1: Single Client Set
Performance versus RED

- ITO = 3
  - May be conservative
- RIO-PS
  - For short/medium flows there is a 25%-30% improvement
- Drop-Tail
  - Usually worse than RED

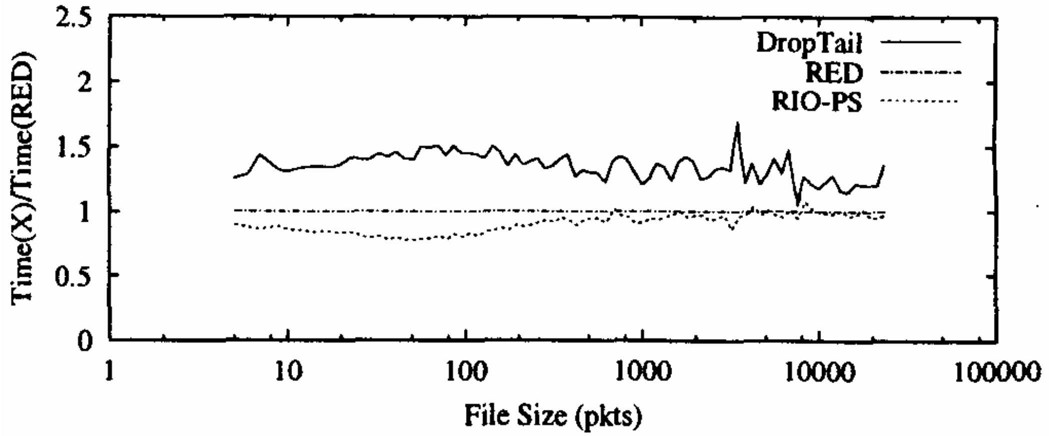


(a) Initial Retransmission Timer 3 seconds

Experiment 1: Single Client Set

Performance versus RED

- ITO = 1
  - May be aggressive
- RIO-PS
  - For short flows there is still a 10%-15% improvement
  - Medium flows still perform well
- Drop-Tail
  - Still worse than RED

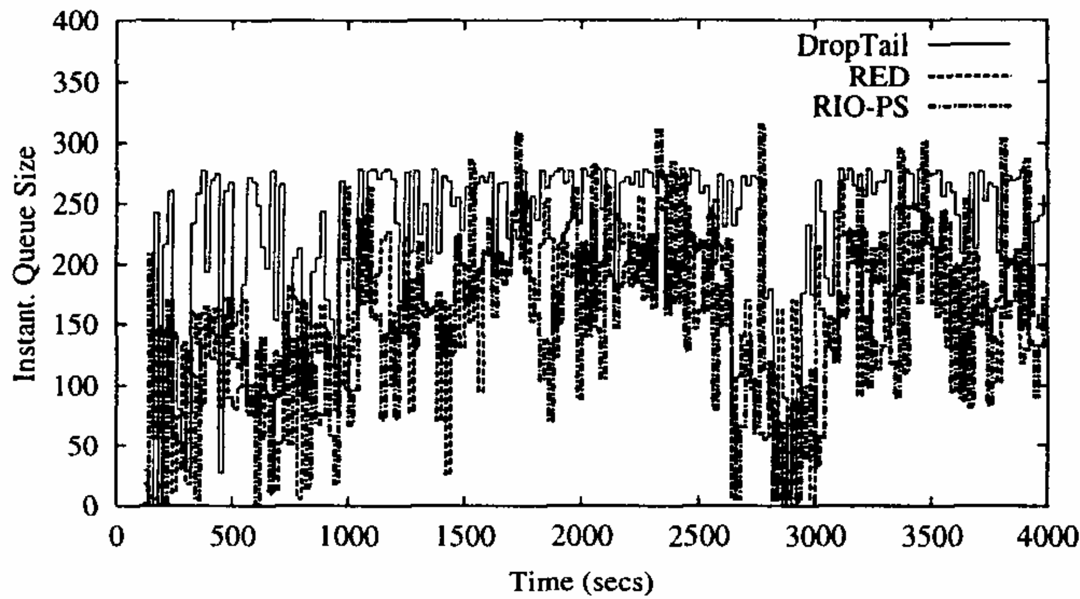


(b) Initial Retransmission Timer 1 second

Experiment 1: Single Client Set

Instantaneous Queue Size

- 20-second 'instants'
- Drop-Tail is the clear loser
- RED and RIO-PS do not display a clear trend
- Overlapping dotted lines are a poor decision

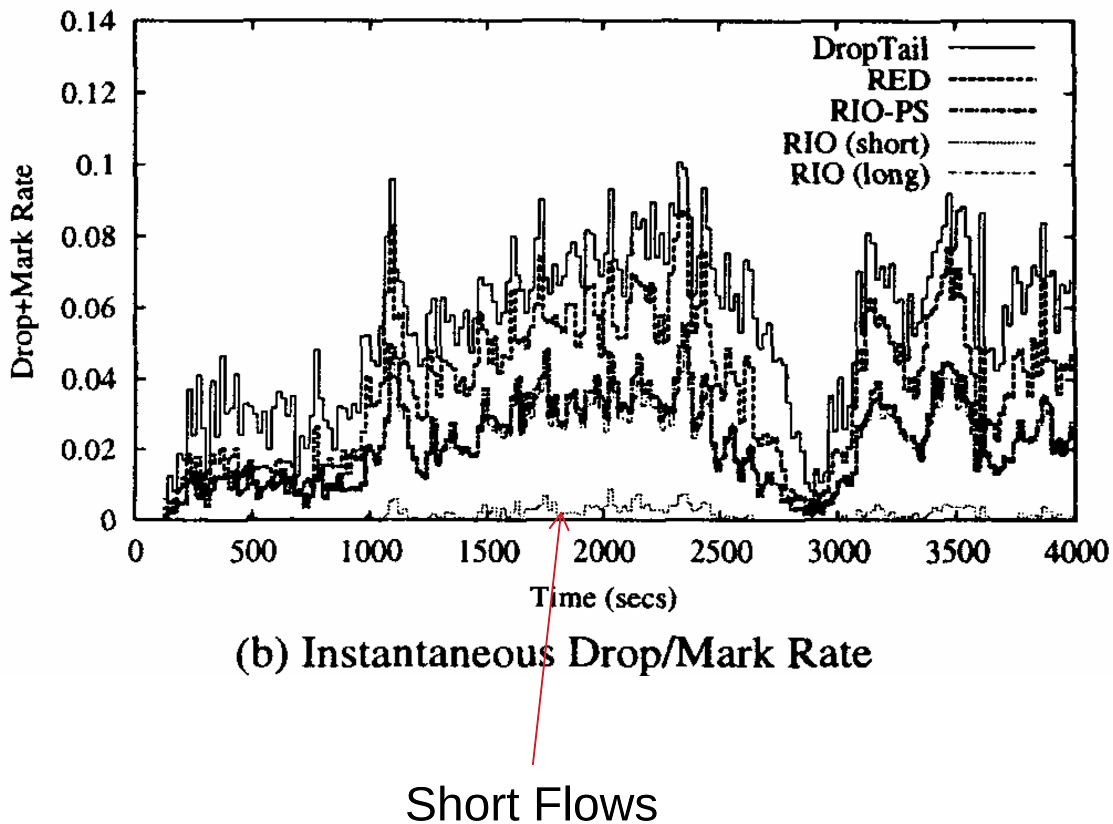


(a) Instantaneous Queue Size

Experiment 1: Single Client Set

Drop/Mark Rate

- 20-second 'instants'
- Drop-Tail drops packets
- RED/RIO-PS mark packets
- Short flows clearly preferred
- In general, RIO-PS outperforms RED



(b) Instantaneous Drop/Mark Rate

Experiment 1: Single Client Set

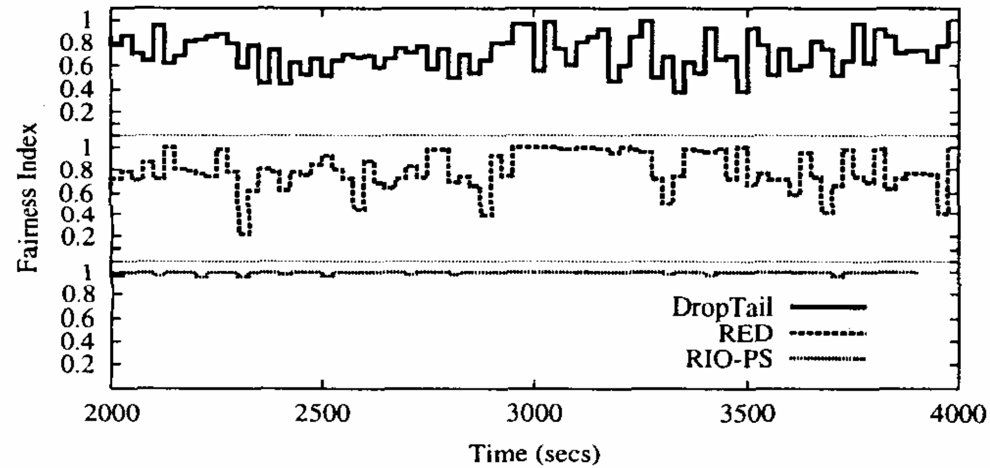
Foreground Traffic

- 10 Short TCP connections injected every 25 seconds
- 10 Long TCP Flows injected every 125 seconds
- Web requests still occurring in background

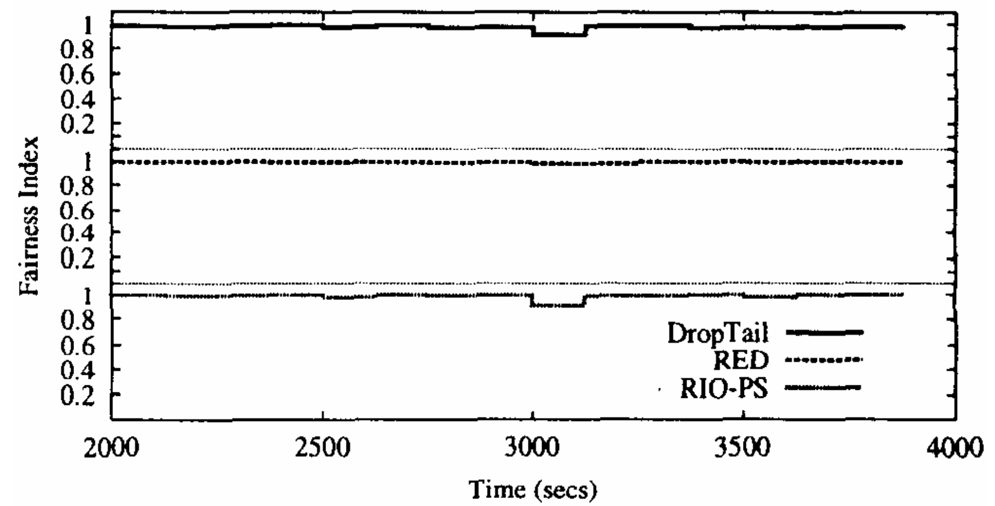
Experiment 1: Single Client Set

Connection Fairness

- Jain's Fairness
- No cross-class comparison
- RIO-PS provides near-perfect fairness between short connections
- No substantial effect on long connections



(a) Fairness Index of Short Connections

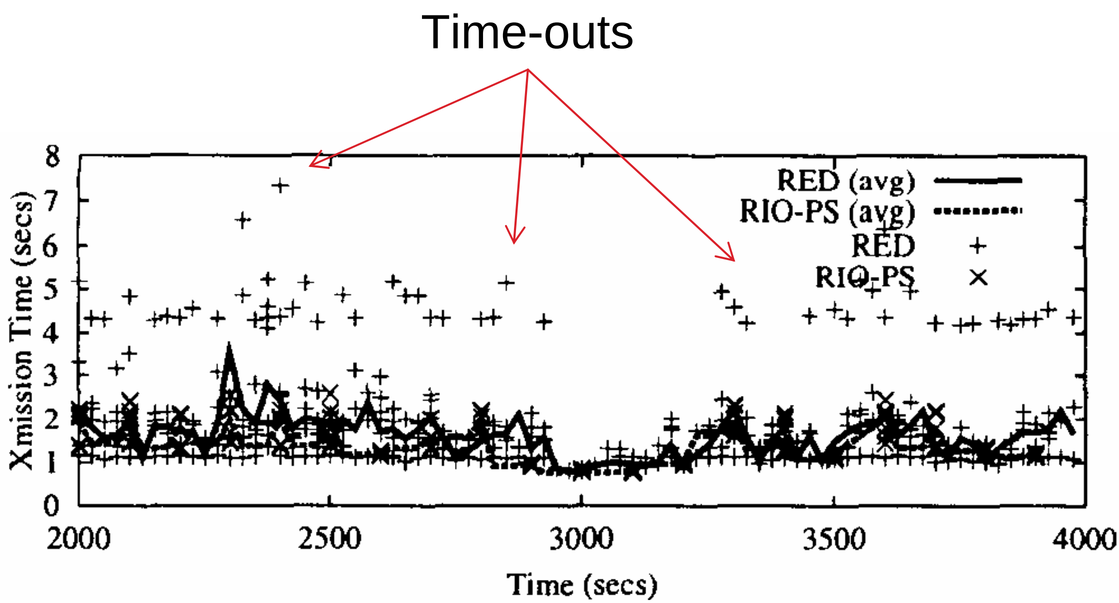


(b) Fairness Index of Long Connections

Experiment 1: Single Client Set

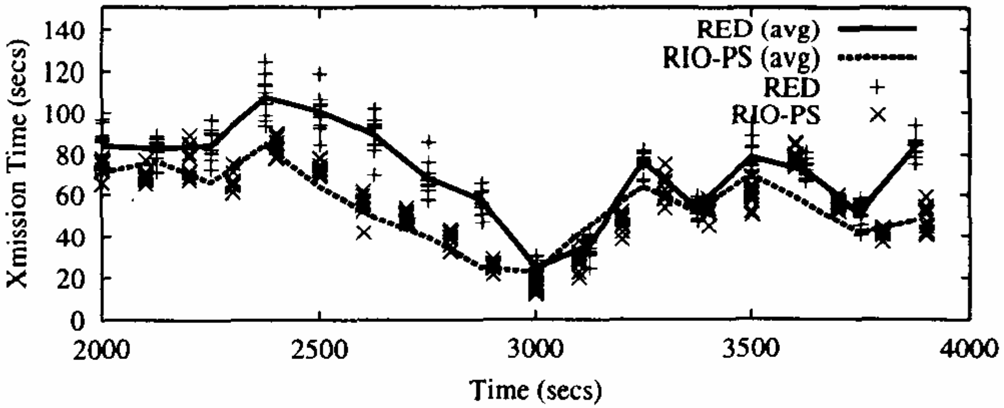
Short Connection Individual Transmission Times

- RIO-PS
  - Experiences dramatically fewer time-outs
  - Better overall transmission times
- RED
  - Vulnerable to SYN/SYN-ACK drops



Experiment 1: Single Client Set
Long Connection Individual Transmission Times

- RIO-PS
  - Noticeably lower transmission times
    - Short flows finishing earlier
- RED
  - More aggressive marking



(b) Transmission Time of Long Connections

Experiment 1: Single Client Set

Summary of Goodput

- Drop-Tail clear loser
  - Dropped packets lower goodput, marked packets do not
- Authors claim RIO-PS increases fairness and does not lower goodput
  - Ambiguous

Scheme	DropTail	RED	RIO-PS
Exp1 (ITO=3sec)	4207841	4264890	4255711
Exp1 (ITO=1sec)	4234309	4254291	4244158

TABLE IV

NETWORK GOODPUT OVER THE LAST 2000 SECONDS

Experiment 2: Unbalanced Requests

- Traffic separated
  - Small files sent on one route, long files in another
- RIO-PS basically reduced to RED, but favoring initial L_t packets of all connections
  - Fewer SYN/SYN-ACK Timeouts

Overview

- Introduction
- Analyzing Short TCP Flow Performance
- Proposed Scheme
- Simulation
- Discussion
- Conclusions

Discussion

- Simulation Comments
  - Dumbbell and Dancehall
  - Does not consider varying propagation delays
  - One-way traffic
- Queue Management Policy
  - No guaranteed Quality of Service
- Deployment Issues
  - More scalable than purely stateful solutions

Discussion (cont)

- Flow Classification
  - Initial packets of all flows protected
- Controller Design
  - More experimentation needed to find optimal parameter settings
- Malicious Users
  - Long flows can be deliberately broken up to emulate short flows
  - Dynamic SLR helps defend against this

Overview

- Introduction
- Analyzing Short TCP Flow Performance
- Proposed Scheme
- Simulation
- Discussion
- Conclusions

Conclusions

- Short flow response time and fairness improved
- Long flows also improved, or at least not harmed
- Overall goodput improved due to less retransmissions
- Flexible and scalable architecture

Our Conclusions

- Experimentation not very thorough
- Only TCP traffic considered
- Did not optimally tune RED parameters
- Fairness charts do not consider overall fairness
- Did not compare RIO-PS performance to other Fair Queueing schemes
- Foreground traffic uses unrealistically low number of flows