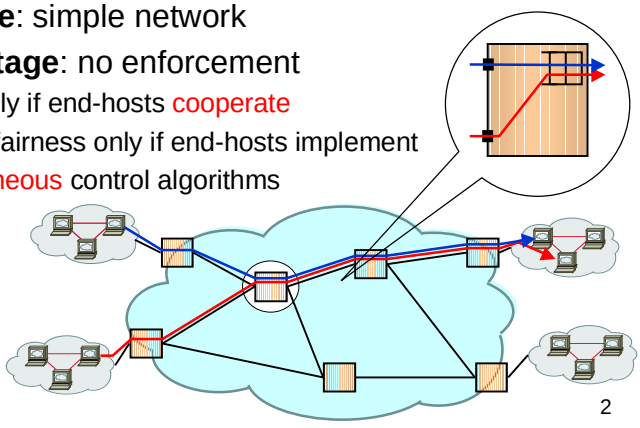
Core-Stateless Fair Queueing: Achieving Approximately Fair Bandwidth Allocations in High Speed Networks

Ion Stoica CMU
Scott Shenker Xerox PARC
Hui Zhang CMU

Congestion Control in Today's Internet

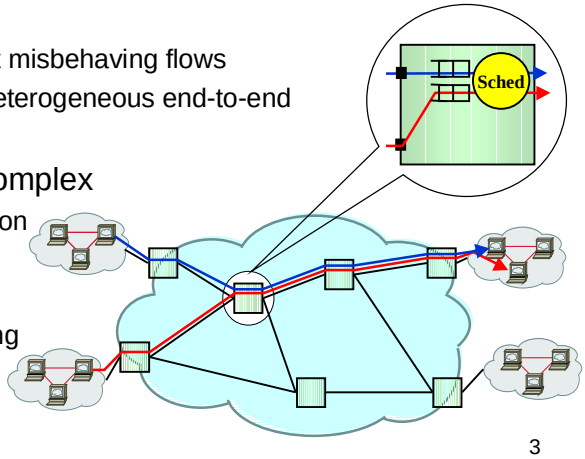
- Rely on end-to-end congestion control (TCP)
  - end-hosts react to congestion indication provided by routers
  - congestion indication: implicit (FIFO, RED) or explicit (ECN)
- **Advantage**: simple network
- **Disadvantage**: no enforcement
  - works only if end-hosts cooperate
  - achieve fairness only if end-hosts implement homogeneous control algorithms



2

Alternate Approach

- Routers enforce protection and fair allocations
  - Flow Random Early Drop (FRED), Fair Queueing (FQ)
- **Advantage**:
  - protection against misbehaving flows
  - co-existence of heterogeneous end-to-end algorithms
- **Disadvantage**: complex
  - packet classification
  - per flow buffer management
  - per flow scheduling (FQ)



3

Challenge

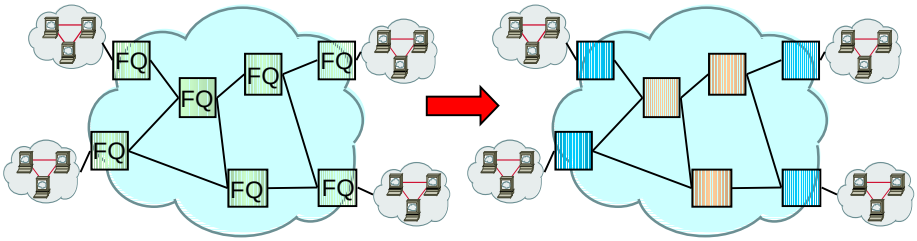
- Router support for congestion control
- Higher speed routers

Achieve protection and fair allocations in high speed networks

4

Our Approach: *Core*-Stateless Architecture

- Approximate a reference network in which every node performs Fair Queueing with a network where
  - edge nodes - do perform per flow management
  - core nodes - do not perform per flow management



Reference network

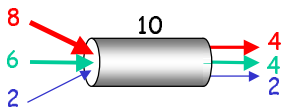
Core-stateless network

edge node core node

5

Fair Queueing

- Work conserving discipline in which each flow is entitled to receive at most the fair rate f associated to the link
  - a flow with arrival rate r receives $\min(r, f)$ bandwidth
  - f computed such that when link congested the aggregate arrival rate equals link's capacity



$f = 4$:
min(6, 4) = 4
min(8, 4) = 4
min(2, 4) = 2

- known algorithms require per flow state
  - e.g.: WFQ, DRR, SCFQ, WF^{2}Q, SFQ

6

Key Insights

1 If each packet of a flow with arrival rate r is forwarded with probability $P = \min\left(1, \frac{f}{r}\right)$

the expected rate of flow's forwarded traffic r' is

$$r' = r \times P = r \times \min\left(1, \frac{f}{r}\right) = \min(r, f)$$

2 No need to maintain per flow state at every node to estimate r, if r is carried by the packet itself

3 To maintain consistency of the estimated rate r, it is enough to updated it with r' as the packet is forwarded

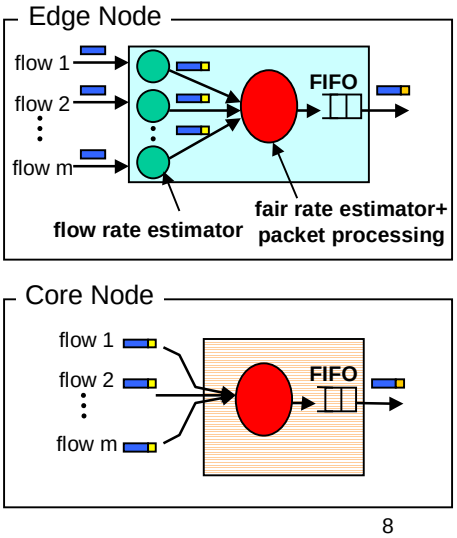
Core-Stateless Fair Queueing Algorithm

- Edge node
  - estimate rate, $\bar{r}$,of each flow and insert it as a label in packet's header
- All nodes
  - estimate fair rate $\bar{f}$ based on link state
  - forward each packet with probability P (where $\bar{r}$ is given by packet's label)

$$P = \min\left(1, \frac{\bar{f}}{\bar{r}}\right)$$

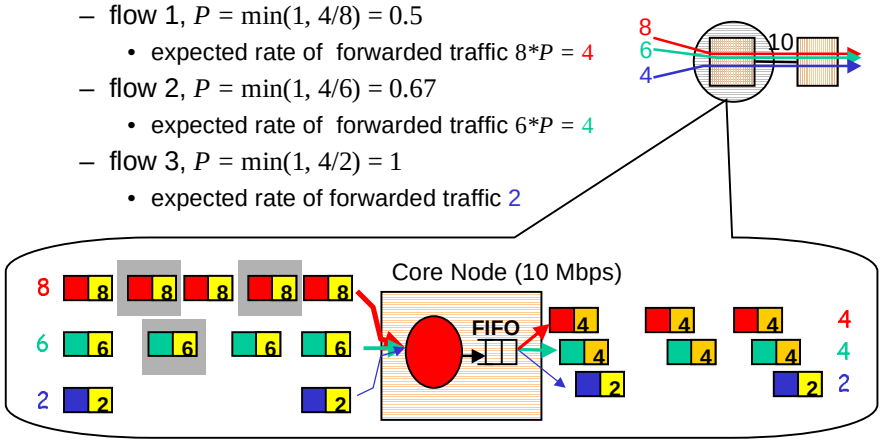
  - update packet label to

$$\min(\bar{r}, \bar{f})$$



Example

- Assume estimated fair rate $\bar{f} = 4$
  - flow 1, P = min(1, 4/8) = 0.5
    - expected rate of forwarded traffic $8*P$ = 4
  - flow 2, P = min(1, 4/6) = 0.67
    - expected rate of forwarded traffic $6*P$ = 4
  - flow 3, P = min(1, 4/2) = 1
    - expected rate of forwarded traffic 2

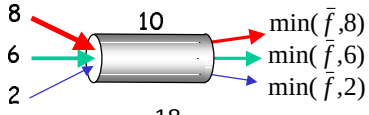


Question: How is the fair rate estimated ?

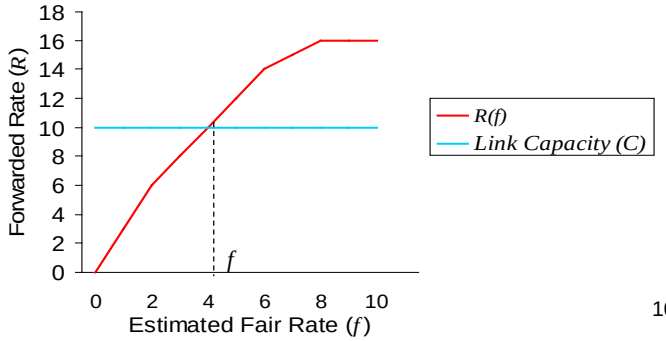
9

Fair Rate Estimation

- Observation - rate of aggregate forwarded traffic (R) is a monotonic and non-decreasing function of the $\bar{f}$ estimated fair rate



$$R(\bar{f}) = \min(\bar{f},8) + \min(\bar{f},6) + \min(\bar{f},2)$$



10

Algorithm Details

- Fair rate estimation - iterative algorithm based on linear interpolation
  - link congested/uncongested - arrival rate is always greater/smaller than link's capacity over a predefined time interval
- Flow rate estimation - exponential averaging

11

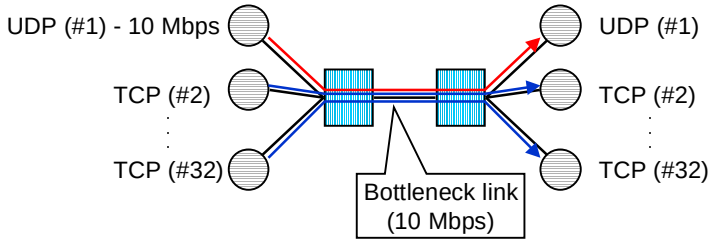
Simulation Results

- Simulations were performed in *ns*-2
- Settings
  - link capacity 10 Mbps, buffer capacity 64 KB
  - link propagation delay 1 ms
- Schemes
  - First-In-First-Out (FIFO)
  - Random Early Detection (RED)
  - Flow Random Early Drop (FRED)
  - Fair Queueing (FQ) implemented by
    - Deficit Round Robin with dropping from the longest queue

12

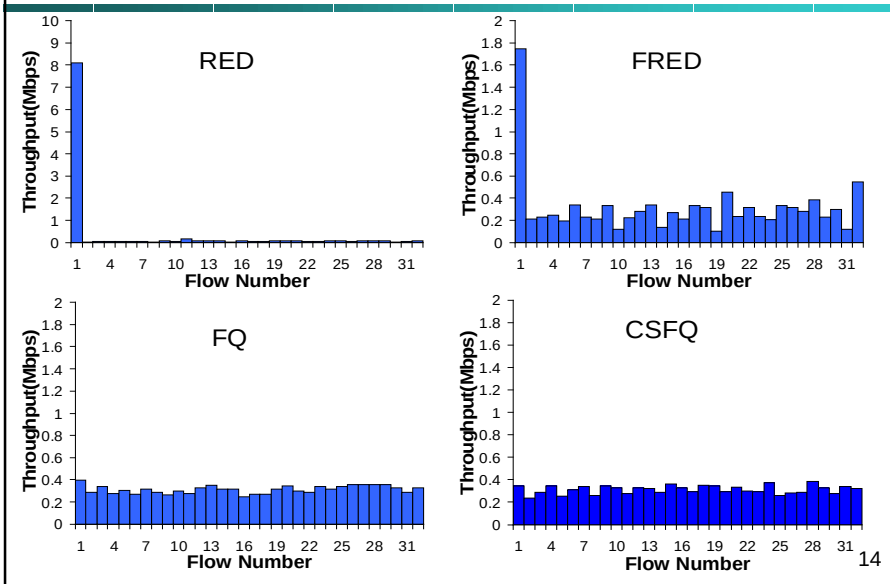
Single Congested Link

- 1 UDP (10 Mbps) and 31 TCPs sharing a 10 Mbps link
- Ideally each flow should receive 10/32 = 0.31 Mbps



13

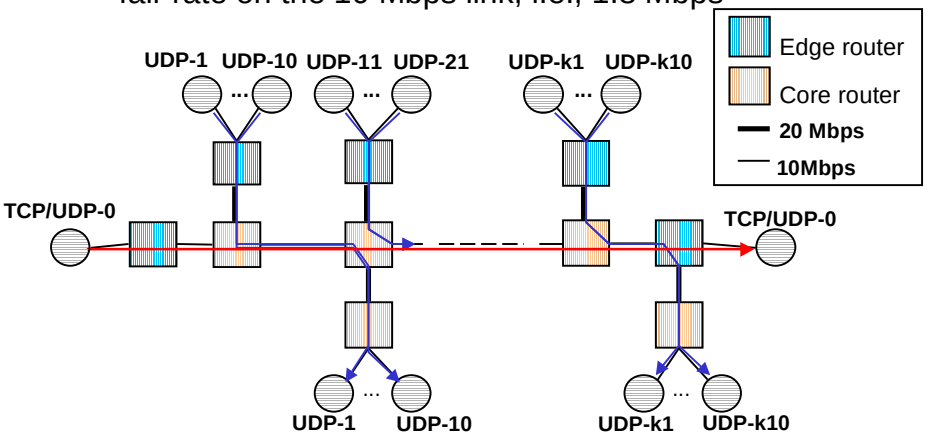
Throughput of TCP and UDP Flows with RED, FRED, FQ, CSFQ



14

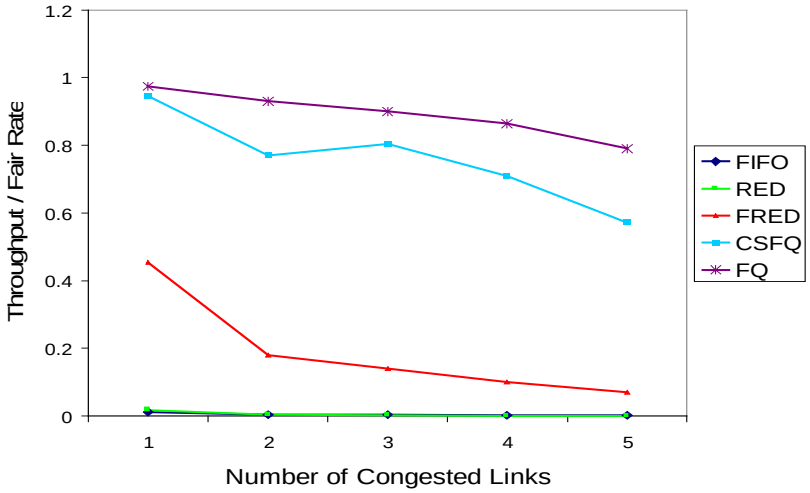
Multiple Congested Links

- Each UDP (excepting UDP-0) sends at twice its fair rate on the 10 Mbps link, i.e., 1.8 Mbps



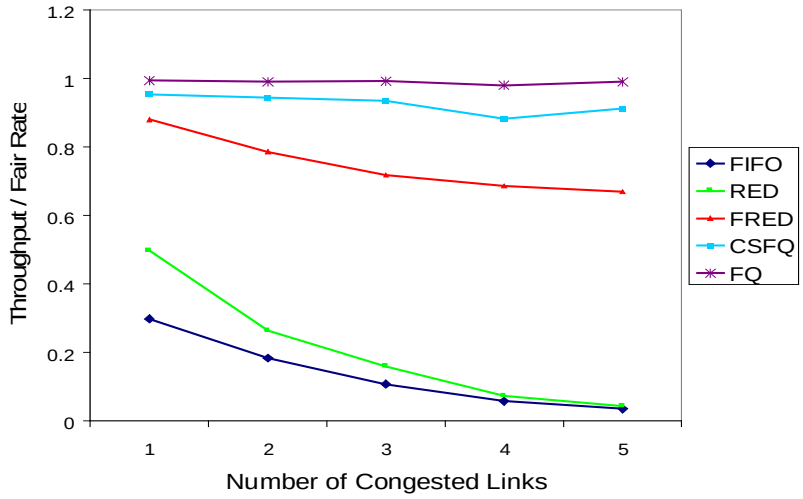
15

Relative Throughput of a TCP over Multiple Congested Links



16

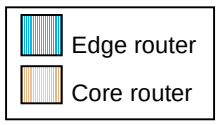
Relative Throughput of a UDP Flow over Multiple Congested Links

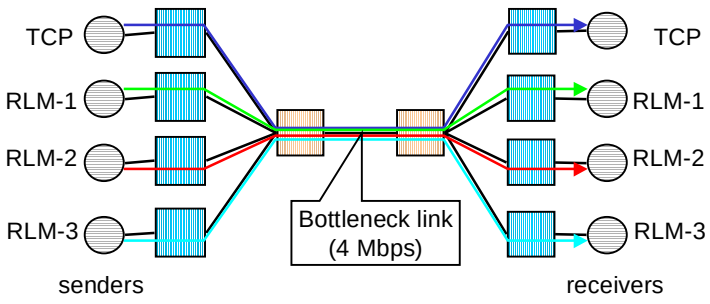


17

Co-existence of Different Adaptation Schemes

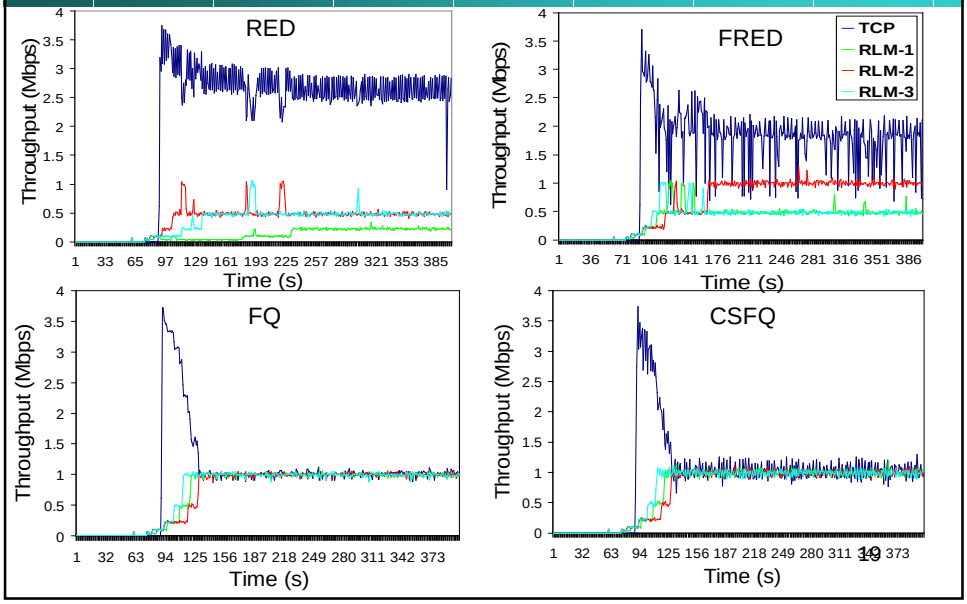
- 4 Mbps single congested link shared by
  - 1 TCP
  - 3 Received-driven Layered Multicast (RLM) flows
  - RLM - 7 levels: 32 Kbps, 64 Kbps, 128 Kbps, ….





18

Throughput of TCP and Three RLM Flows



Conclusions and Future Work

- Architecture and algorithm (CSFQ) for high speed
  - achieve fair allocation close to FQ and comparable or better than FRED under most simulation scenarios
  - do not require core nodes to maintain per-flow state
  - can approximate weighted FQ
- **Open problems**
  - impact of very large latencies
- **Future work**
  - better estimators for flow and fair rates
  - per-flow guarantees without per flow management at core nodes

http://www.cs.cmu.edu/~istoica/csfq

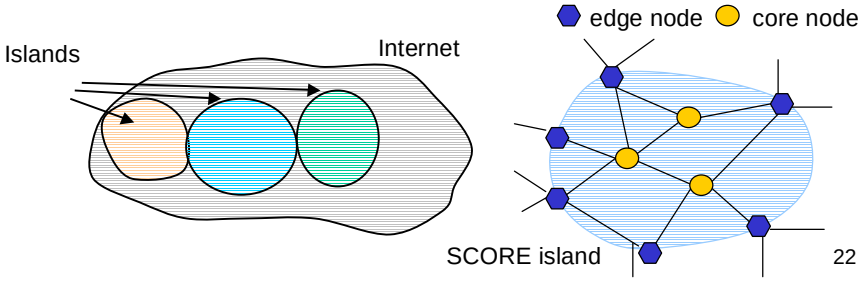
20

Questions ?

21

Core-Stateless Architecture

- Network is partitioned in islands
  - contiguous region of network
  - trusted domain
  - nodes cooperate to achieve a common goal
  - differentiate between
    - edge nodes - do perform per flow management
    - core nodes - do not perform per flow management



22

Related Work

- RED with Identification [Floyd & Fall]
  - use dropping history to identify misbehaving flows
  - punish misbehaving flows
- Advantages: strong incentive for applications to adapt
- Disadvantages:
  - assume homogeneous control algorithms (TCP-friendly)
  - hard to accurately identify misbehaving flows
  - identification process quite complex

23

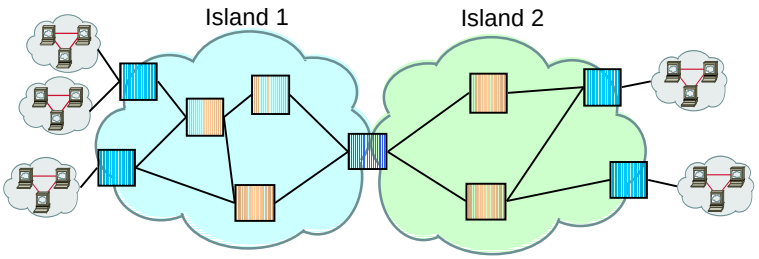
Related Work - CSFQ vs. ATM *ABR*

- ATM *ABR*
  - close loop
  - RM cell - target rate
  - usually, maintain per-VC state (ATM routers maintain anyway per-VC state)
  - designed for networks with few losses
- CSFQ
  - open loop
  - label - current rate
  - no per-flow state at core routers
  - robust in presence of heavy losses

24

Edge Router Types

- Client to island
  - input speed < output speed
  - inputs: can efficiently implement rate estimation
  - outputs: no need to maintain per flow state
- Island to island
  - still need for high speed routers, but
  - fewer than an all FQ design
  - simpler than FQ



25

Bound

- The estimated excess service that a flow sending at a rate no larger than r on a congested link with fair rate f can receive under CSFQ is bounded by

$$f \times K \times \left(1 + \ln \frac{r}{f}\right) + l_{\max} \text{ bits}$$

  - K : averaging constant used for flow's rate estimation
  - l_{max} : maximum length of a flow packet

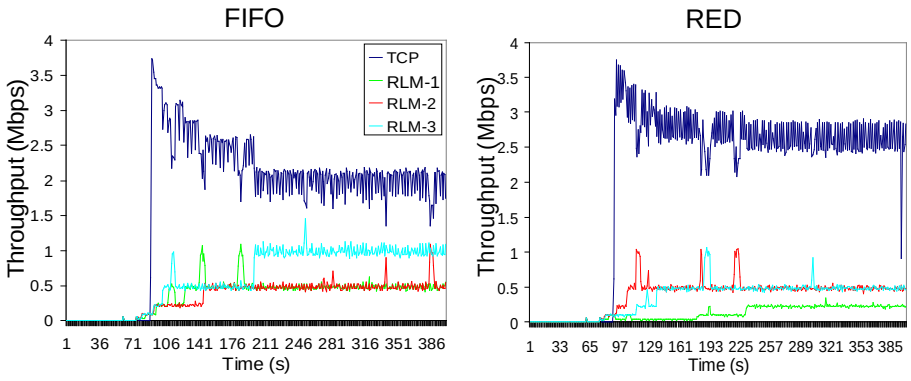
26

Complexity

	FIFO/RED	FRED	FQ	CSFQ
State	O(1)	O(n)	O(n)	O(n) - edge O(1) - core
Time	O(1)	O(1)	O(log n)	O(1)

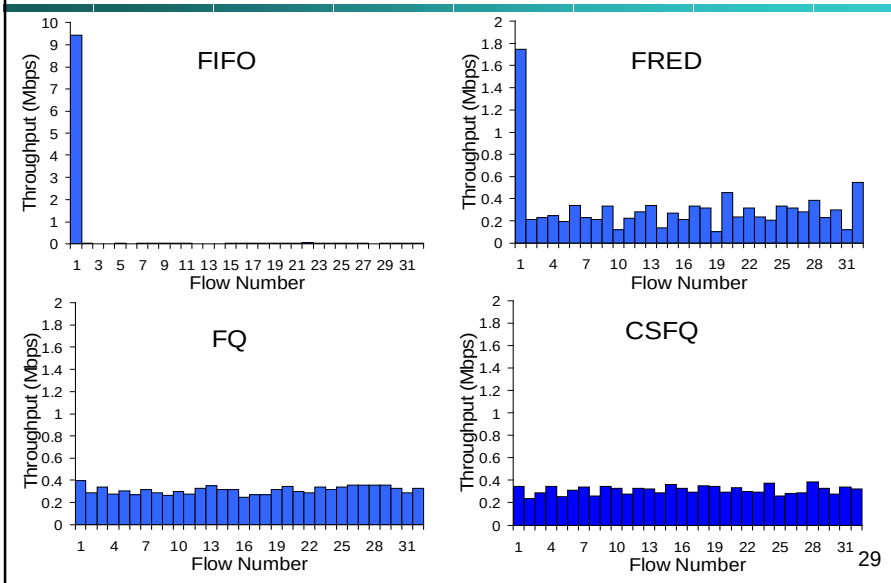
27

Throughput of TCP and Three RLM Flows



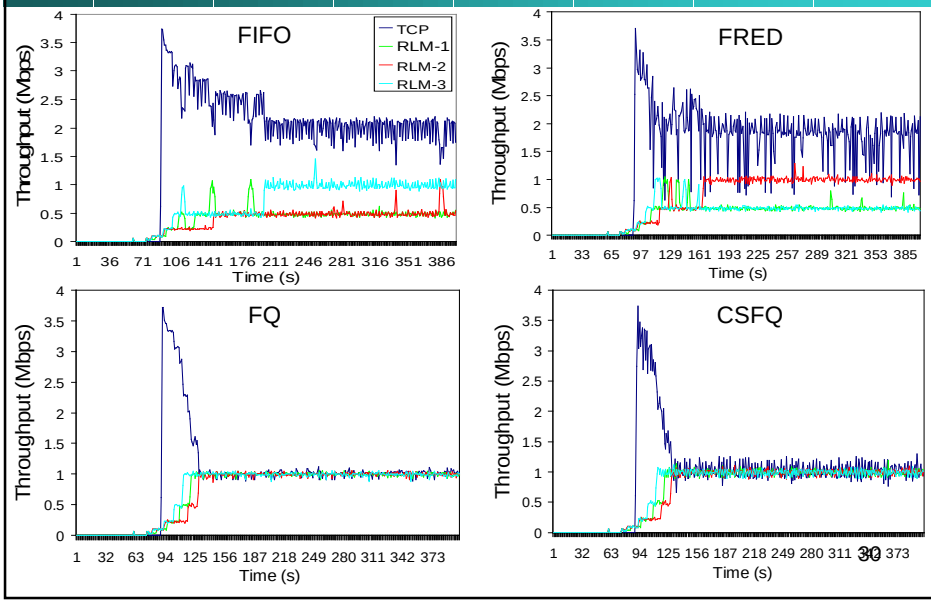
28

Throughput of TCP and UDP Flows with FIFO, FRED, FQ, CSFQ



29

Throughput of TCP and Three RLM Flows



30

Algorithm Details

if link congested

$$\bar{f} = \bar{f} \times C / R$$

if link uncongested

$\bar{f}$ = largest label (rate) seen during the last interval of size K_c

if buffer overflows

decrease $\bar{f}$ by a small fixed percentage (e.g., 1%)

- link congested - arrival rate is always greater than link capacity during an interval of size K_c
- link uncongested - arrival rate is always lower than link capacity during an interval of size K_c , or buffer occupancy is less than a predefined threshold (e.g., half of buffer size)

31