Core Stateless Fair Queueing

Stoica, Shanker and Zhang - SIGCOMM 98

- Rigorous fair Queueing requires **per flow state**: too costly in high speed core routers
- Yet, some form of **FQ essential** for efficient, fair congestion control in the backbone network
- **Proposed solution**:

(a) **per flow** accounting and **rate labeling** at **edge routers**

(b) **packet state**: packets carry **rate labels** (eg, in TOS field)

(c) **stateless FQ** at core routers: no per flow state kept; packet drop probability computed directly from pkt label

Key Elements of CSFQ

- Edge router estimates **current rate r(i)** of each flow and stamps it in IP header (eg, TOS field)
- Flow **rate value adjusted** as pkt travels through various bottlenecks in the backbone
- Core router **estimates max/min fair share** on its links based on aggregate traffic measurements
- Core router **probabilistically drops** packets in a flow which exceeds fair share

Fair Share Computation at Router

- Assume N flows arrive at core router
- Each flow rate **r(i)** is stamped in header
- Max-Min fair operation:

  (a) all **bottlenecked** flows get "**fair share**" rate "**a** " (the excess rate packets are dropped)

  (b) **non-bottlenecked** flows are granted their **full rate**

Thus, at full trunk utilization:

Sum (over i = 1..N) of min{ r(i,t), a(t) } = **C**

where **C** = trunk capacity

Fair Share Computation (cont)

If all r(i) are known at the router, fair share **a** can be easily computed:

(a) try an arbitrary fair share threshold **a(0)**

(b) from "fair share" formula compute the resulting link throughput R

(c) compute new value **a(1)** = C/R

(d) go back to (b) and iterate until **a(n)** converges to fixed point

Probabilistic Dropping at Router

- If aggregate arrival rate A < C, no pkt is dropped
- If A > C (ie, congested link):

  (a) bottlenecked flow (ie, r(i,t) > a(t)): drop the fraction of "bits" above the fair share, ie (r(i,t) -a(t))/r(i,t)

  (b) non-bottlenecked flow: no dropping

  Equivalently:

  packet drop probability = max (0,1- a(t)/r(i,t))

- adjust rate label value: r(i,t) <= min (r(i, t), a(t))

Implementation details (cont)

(a) **flow arrival rate** at edge router computed with exp avg

(b) **fair share computation at core router**:

measure aggregate arrival rate A(t) using exp averaging

If router is congested (ie, A(t) >C), then:

measure (exp avg) the fraction F of bits currently accepted

ie, F(t) = current acceptance rate

Assume F is a linear function of **a** (in reality concave function). Then:

New fair share value: **a(new) = a(old) C/F(t)**

More details..

- Occasionally, router buffer overflows:
- then, decrease a(t) by 1%
- Never increase a(t) by more than 25%
- Link is considered uncongested if occupancy < 50% of buffer capacity
- Weighted CSFQ option:

  if w(i) is the weight of flow i, then:

  r(i) <= r(i)/w(i)

Simulation Experiments

- **FIFO**
- **RED** (FIFO + Random Early Detection)
- **FRED** (Flow Random Early Drop, SIGCOMM 97): extension of RED to improve fairness; it keeps state of flows which have one or more pkts in queue; it preferentially drops pkts from flows with large queues
- **DRR** (Deficit Round Robin): **per flow queueing**; drops packets from **largest** queue

Single Congested Link Experiment

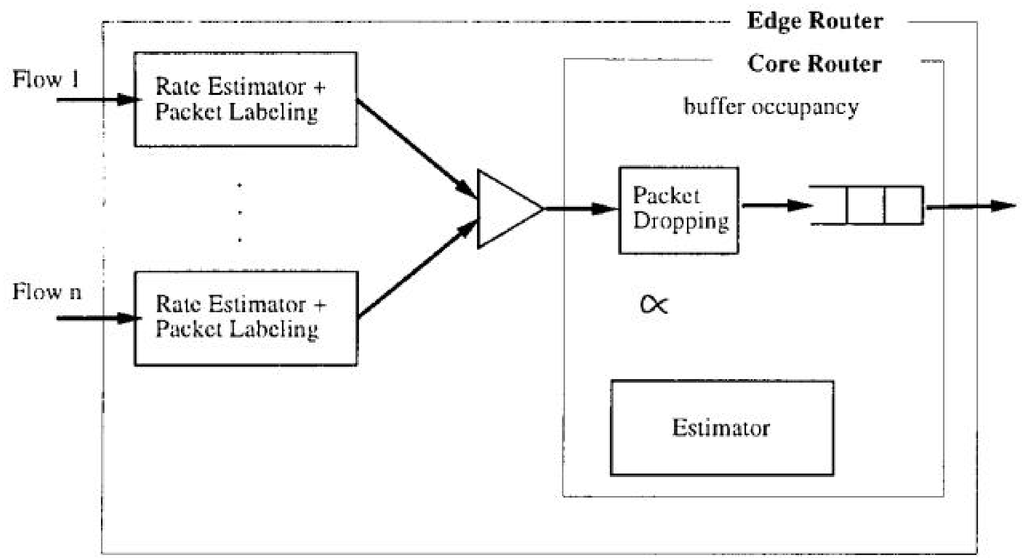
10 Mbps congested link shared by N flows

(a) 32 UDP flows with linearly increasing rates

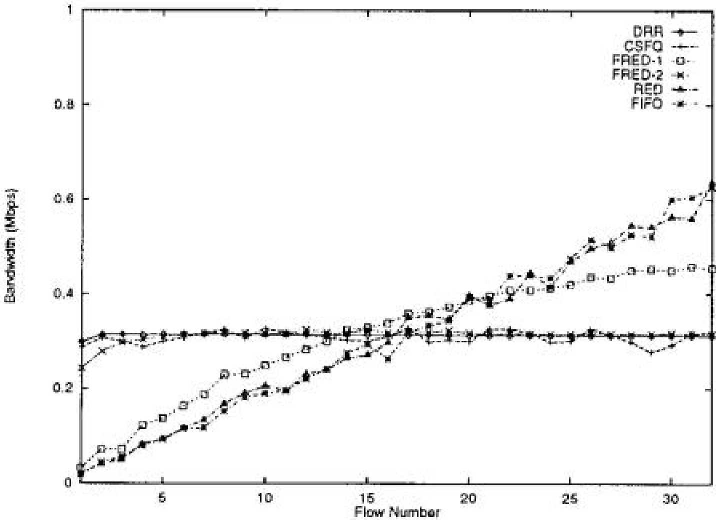
(b) single “ill behaved” UDP flow; 31 TCP flows

(c) single TCP flow; 31 “ill behaved” UDP flows

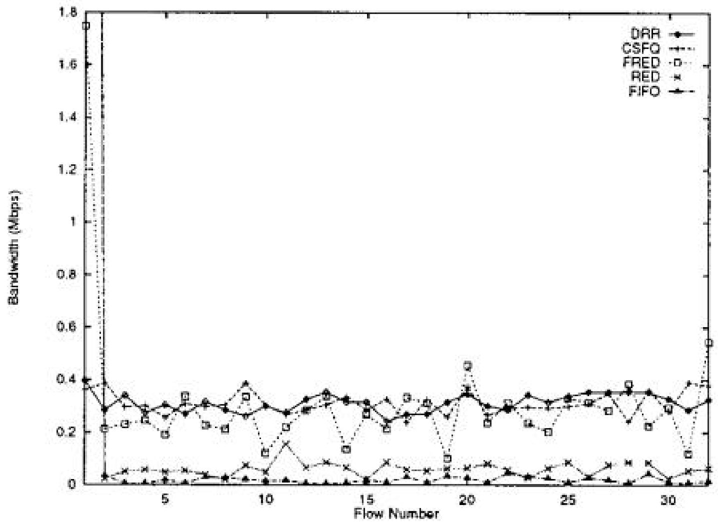
Edge and Core Routers



(a) linear rate UDPs; (b) single UDP + 31 TCPs

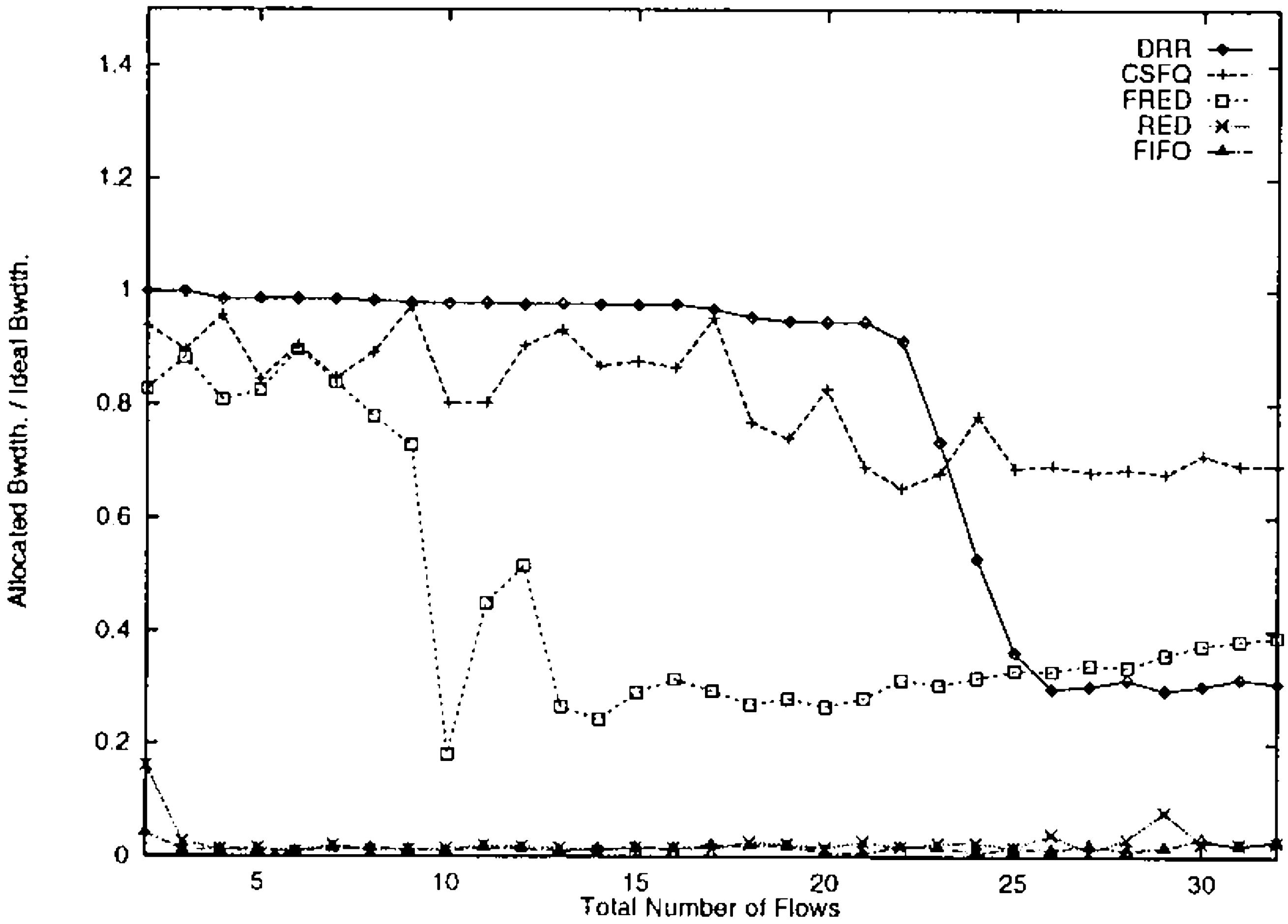


(a)

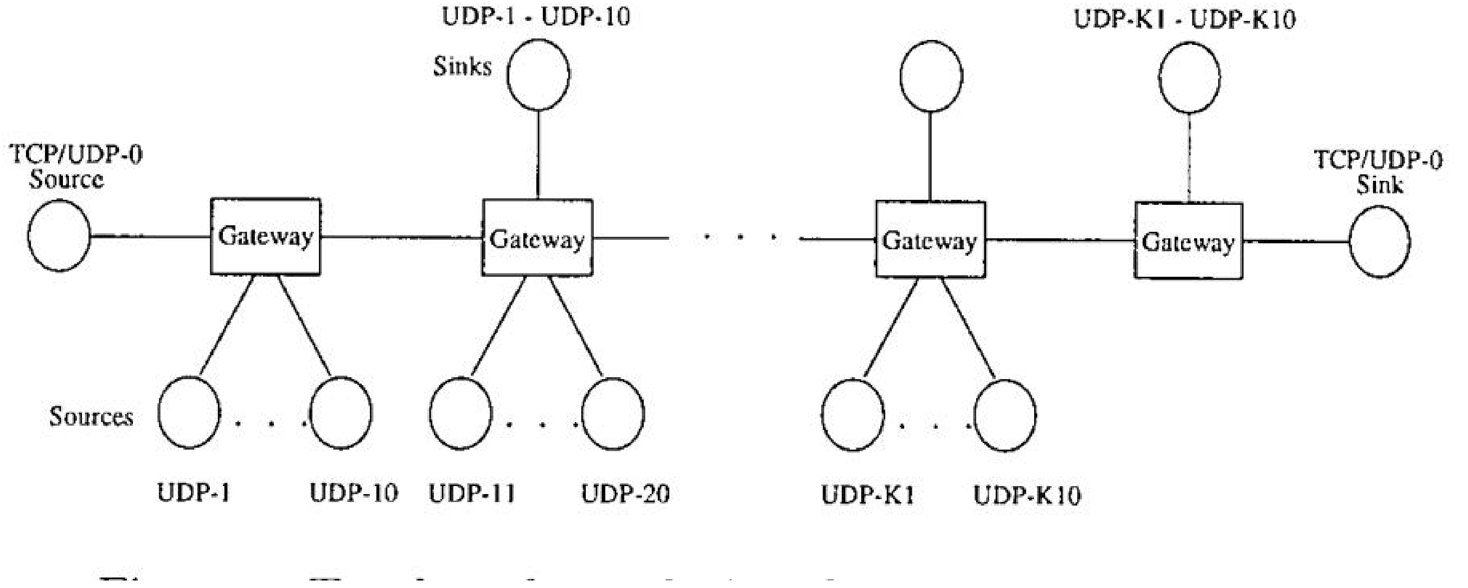


(b)

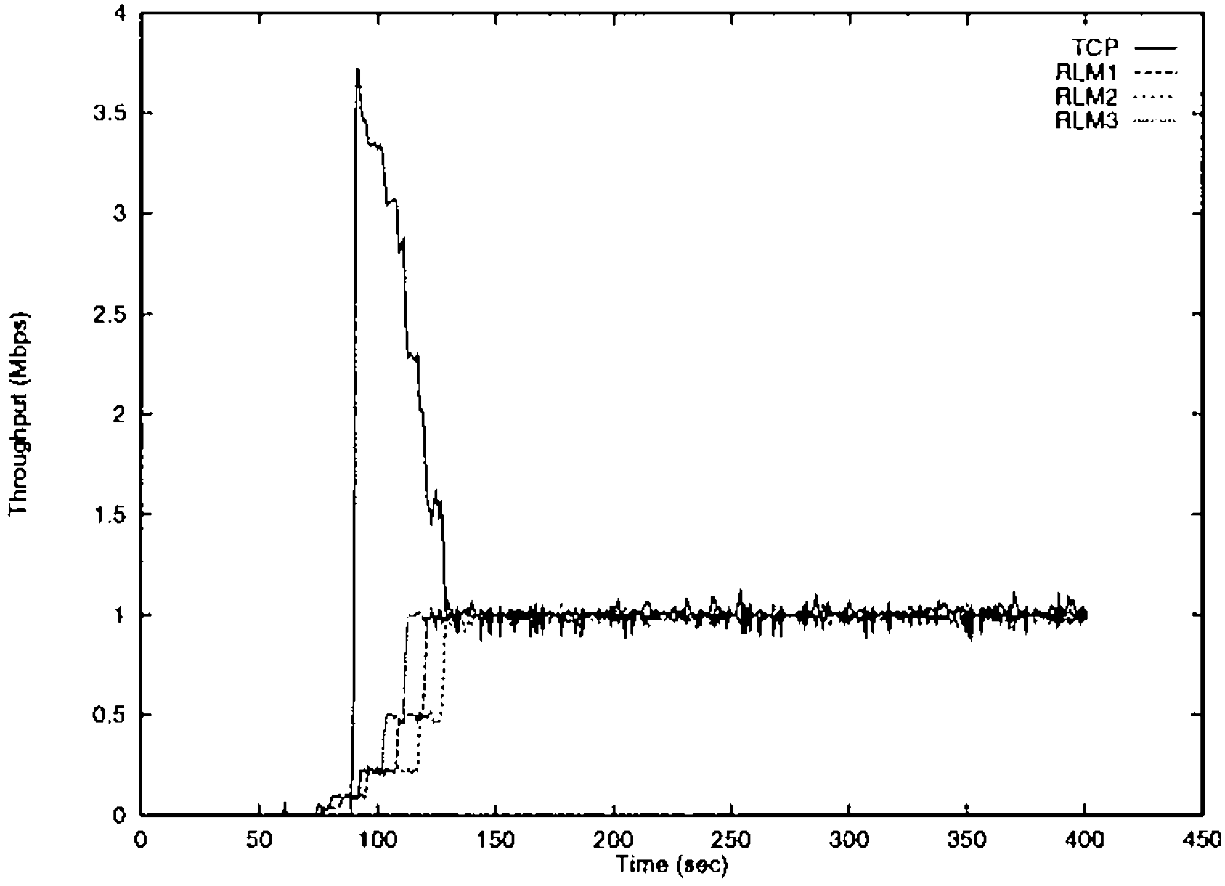
Single TCP competing with up to 31 UDPs



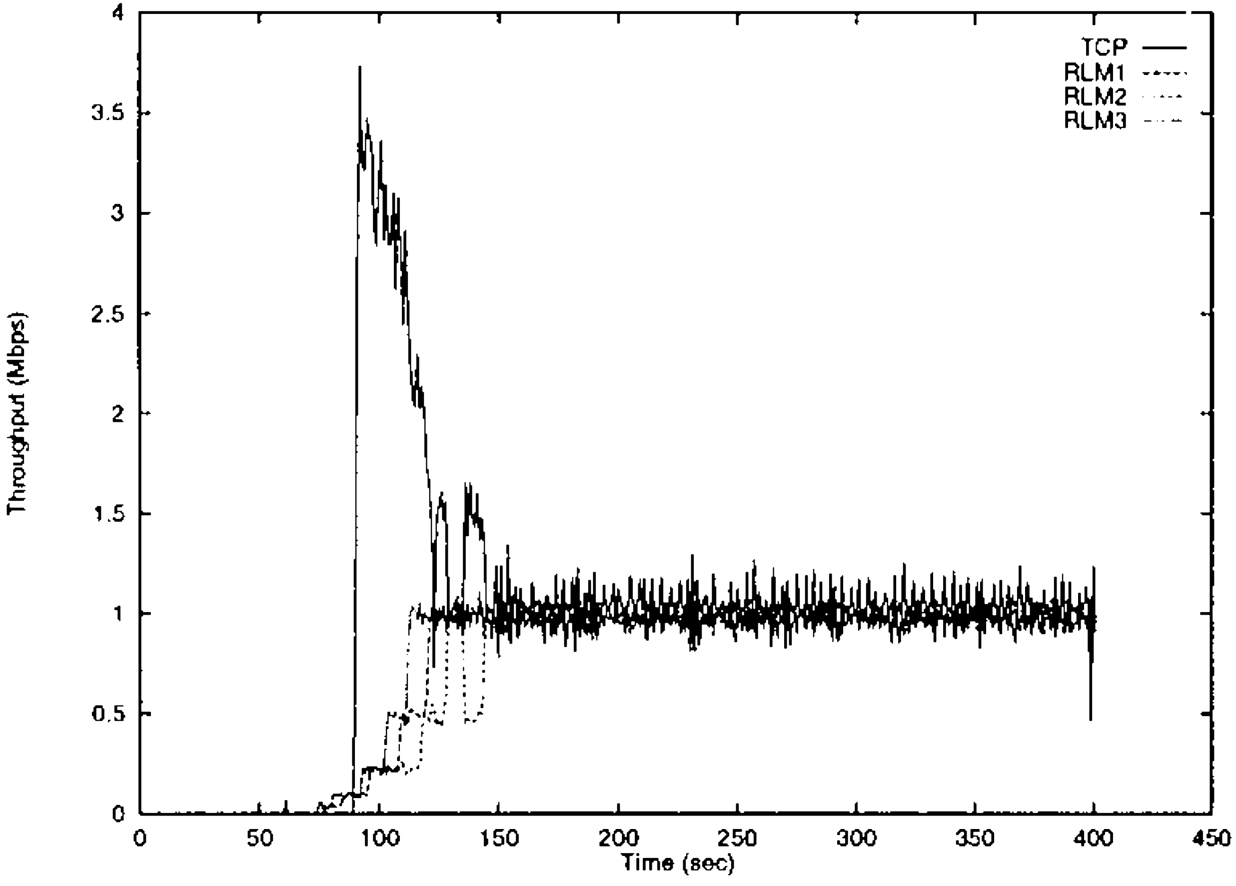
Multiple congested links



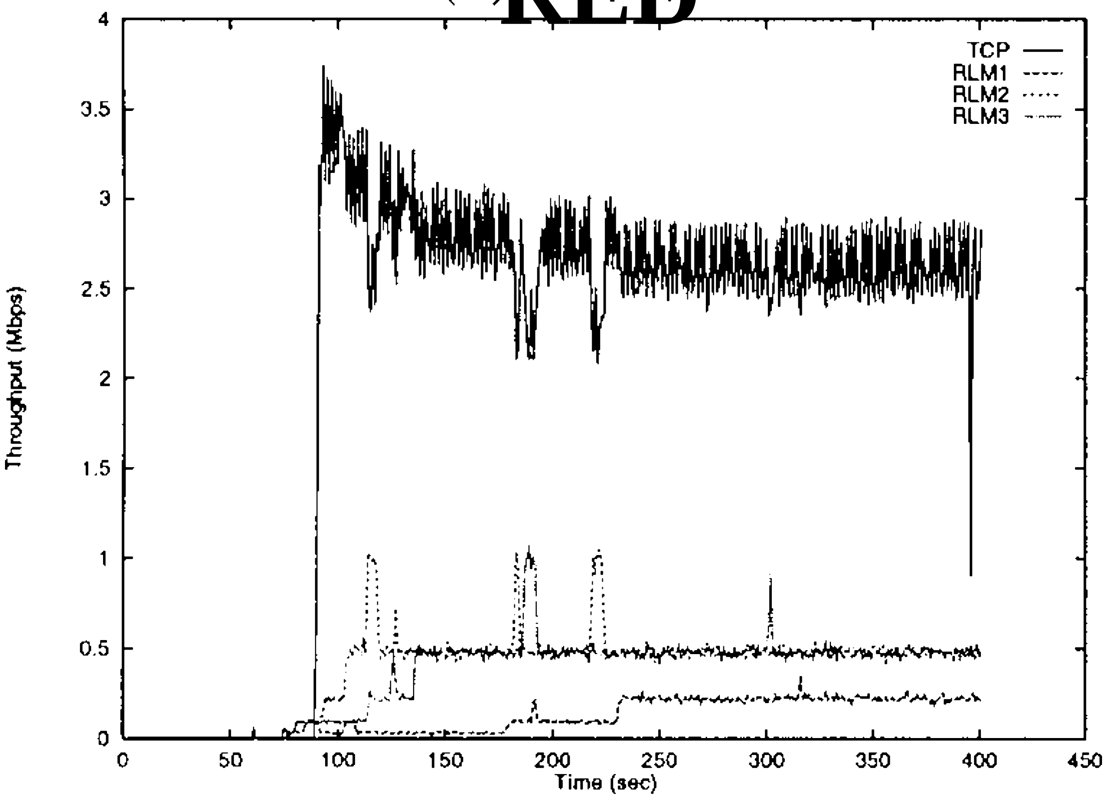
Coexistence of TCP and Receiver Layered Multicast: DRR



Coexistence of TCP and Receiver Layered Multicast: CSFQ



Coexistence of TCP and Receiver Layered Multicast: RED



(d) RED

Conclusions

- **CSFQ does not require per flow state within the core**
- **CSFQ performance comparable to DRR (which however requires per flow state)**
- **superior to FRED ("partial" per flow state)**
- **much better than RED, FIFO (no per flow state)**
- **large latency and propagation delay effects (such as on a cross country connection or on a satellite segment) still to be explored**
- **use of TOS field (ie,packet state) potentially controversial**