A Flow Table-Based Design to Approximate Fairness

Hot Interconnect X
August 21, 2002

Rong Pan
Stanford University

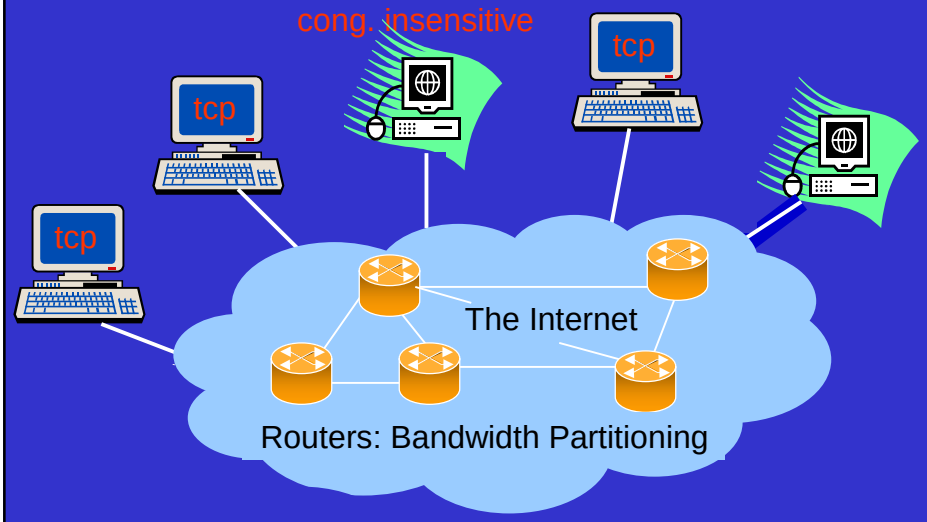
Joint Work with Lee Breslau, Balaji Prabhakar and Scott Shenker

Outline

- Background
- Goal
- AFD-FT Algorithm
- Simulation Results
- State Requirement and Implementation
- Summary

2

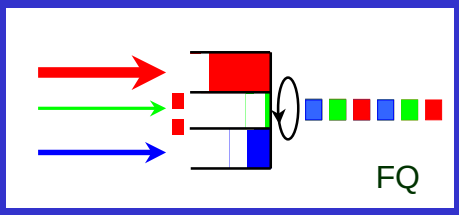
Background



3

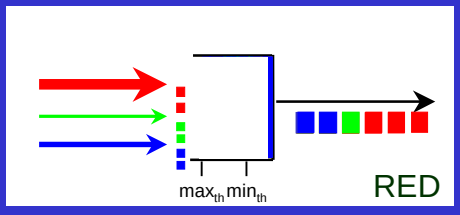
How to Partition the Bandwidth?

- Network-Centric (any type users)



+ Isolation among flows
- Needs queues for each flow (per flow state)

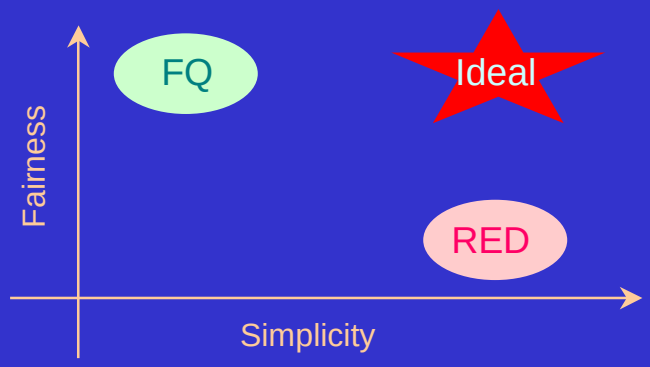
- User-Centric (congestion-aware users)



+ Easy to implement
- No isolation among flows

4

Goal



- Provide equal share to all users
- Be simple to implement

5

How to Achieve Fairness? - a simple analysis

R_i : the sending rate of flow i
D_i : the drop probability of flow i
Ideally, we want

$$R_i(1 - D_i) = R_{fair} \text{ (equal share)}$$

$$\Rightarrow D_i = (1 - \frac{R_{fair}}{R_i})_+$$

(that is, drop the excess)

6

Solution for Estimating R_{fair} and R_i

Record the ids of the *N* most recent arrivals

Let m_i: # of flow i's packets in these *N*
R: total arrival rate

$$R_i \approx R \frac{m_i}{N}$$

$$D_i \approx (1 - \frac{m_{fair}}{m_i})_+$$

$$\sum_i R_i (1 - D_i) \approx C$$

7

Approximate Fair Dropping (AFD)

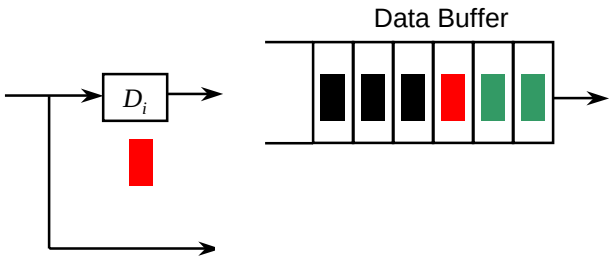
- AFD-SB (Theoretical Solution)

Control Plane

- Define a drop function as
  - $D_i = (1\text{-}m_{fair} / m_i)_+$

8

An Example



$$D_i \approx (1 - \frac{m_{fair}}{m_i})_+$$

$$m_{fair} = 2$$

5	0
2	4

Flow Table

9

Removing the Shadow Buffer

- Maintain a flow table
  - \+ *Easy to increment: the incoming flow i*
  - – *Difficult to decrement: choose which flow?*

- The drop function is the same as before
  - – $D_i = (1-m_{fair}/m_i)_+$

10

Goal for Mimicking AFD-SB

- Under AFD-SB, the probability for decrementing a flow i's packet: m_i/N
- On average, m_i flow i's packets replaced after N updates

The desired behavior

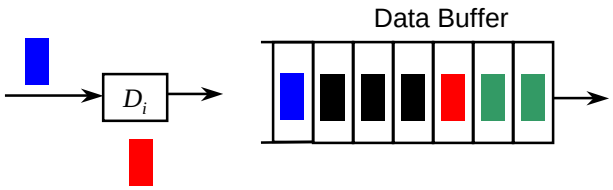
11

AFD Flow Table (AFD-FT)

- Keep a flow uniformly drawn from the flow table
  - *$P_i = 1/N_f$ (N_f : # of flows)*
- Once a flow is chosen, use it for the next N_u updates
  - *$N_u = a*m_i$ (a: a small constant ≤1)*

12

An Example (a = 0.5)



$$D_i \approx (1 - \frac{m_{fair}}{m_i})_+$$

$$m_{fair} = 2$$

$$N_u = ?$$

5	2
2	2

Flow Table

Choose another flow

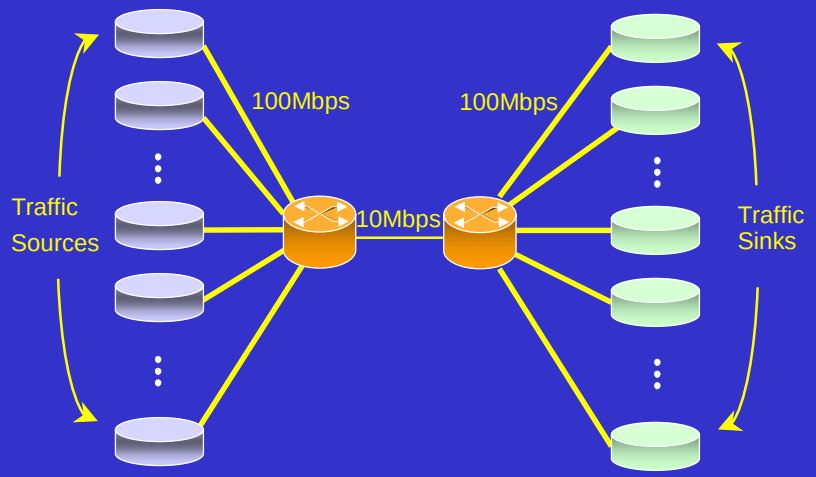
13

AFD-FT's Performance

- AFD-FT can achieve the performance of AFD-SB's performance
  - *has been proven theoretically*
  - *also has been verified through simulations*

14

Simulation Setup



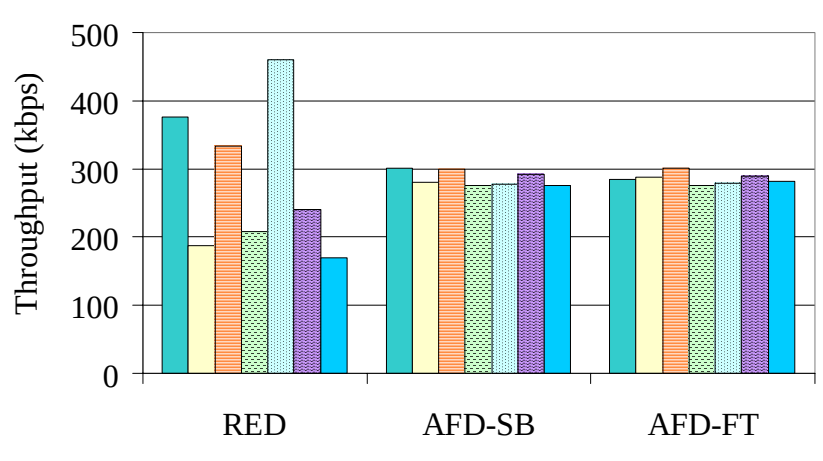
15

Network Setup Parameters

- All TCP's maximum window size = 300
- FIFO buffer size = 300 packets
- All packets sizes = 1 KByte
- N = 500, a = 0.06

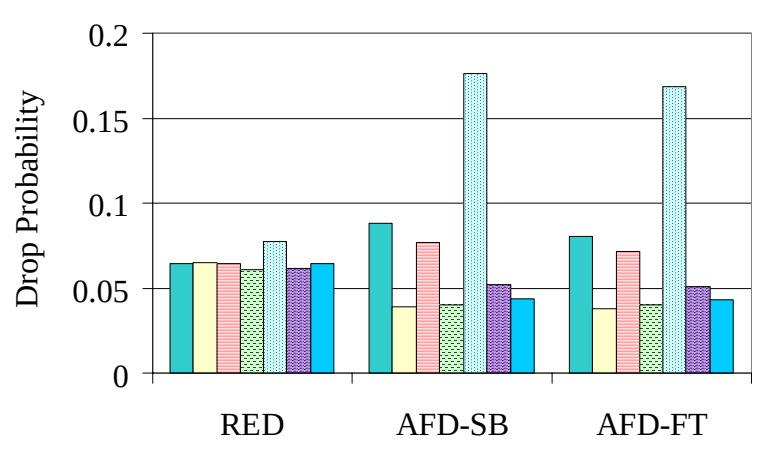
16

Mixed Traffic
with Different Levels of Unresponsiveness



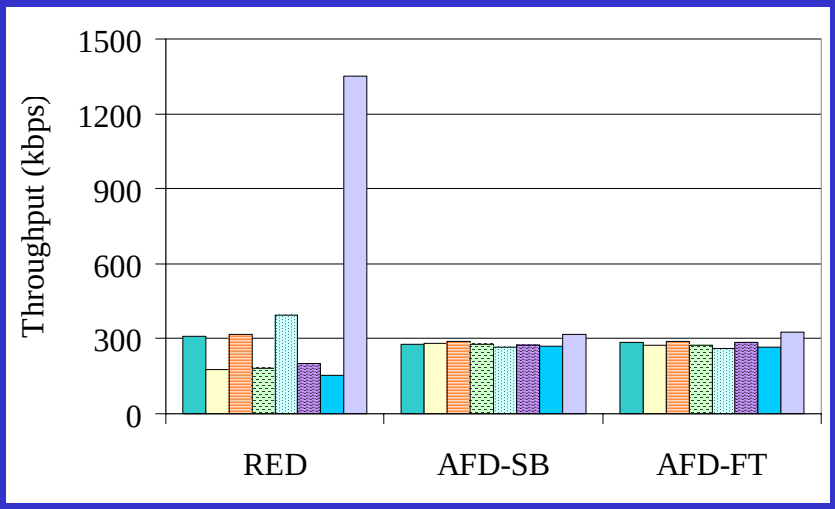
17

Drop Probabilities
(note differential dropping)



18

Add One CBR flow to the Mixed Traffic



19

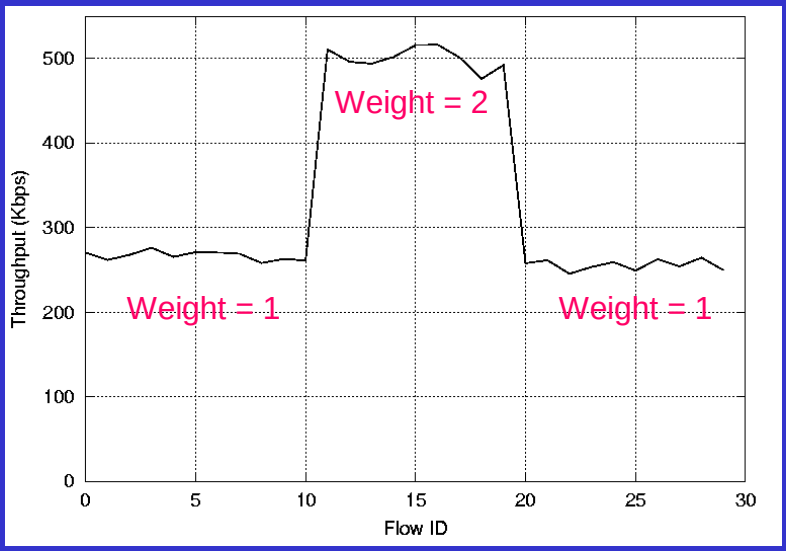
Simple to Handle Different Weights

(5, 1)	(0, 1)
(2, 2)	(4, 1)

$$D_i \approx (1 - \frac{m_{fair}}{m_i / w_i})_+$$

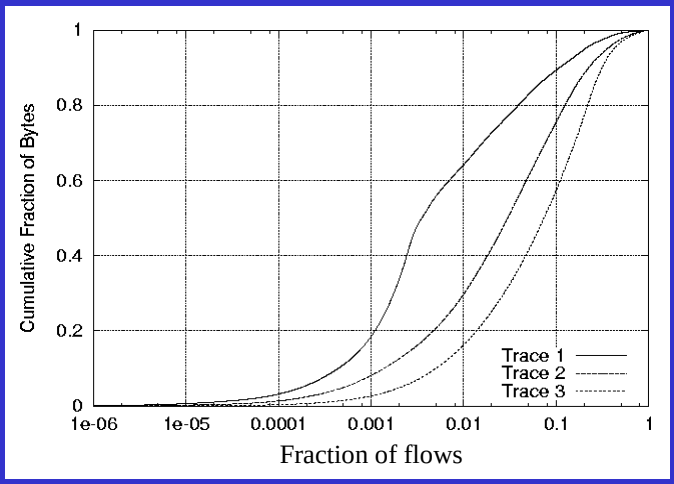
20

Fairness to Flows with Different Weights - 30 TCP flows



21

Not Per-Flow State



- State requirement on the order of # of unresponsive flows

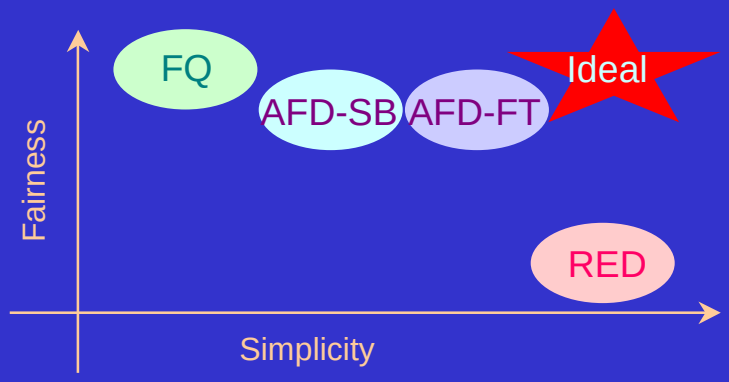
22

Implementation

- The entries of the flow table: limited
  - *For 1Gbps: the size is about tens of thousands, not millions*
- Implement the "Flow Table" using
  - *SRAM (Hash Table)*
  - *CAM*

23

AFD-FT - Summary



- Equal share is approximated in a wide variety of settings
- The state requirement is limited

24