

Increase/Decrease of $cwnd$ Revisited

By how much should $cwnd$ (w) be changed?

Increase: $w' = b_i w + a_i$

Decrease: $w' = b_d w + a_d$

Alternatives:

1. Additive increase, additive decrease:

$$a_i > 0, a_d < 0, b_i = b_d = 0$$

2. Additive increase, multiplicative decrease:

$$a_i > 0, 0 < b_d < 1, a_d = b_i = 0$$

3. Multiplicative increase, additive decrease:

$$b_i > 1, a_d < 0, a_i = b_d = 0$$

4. Multiplicative increase, multiplicative decrease:

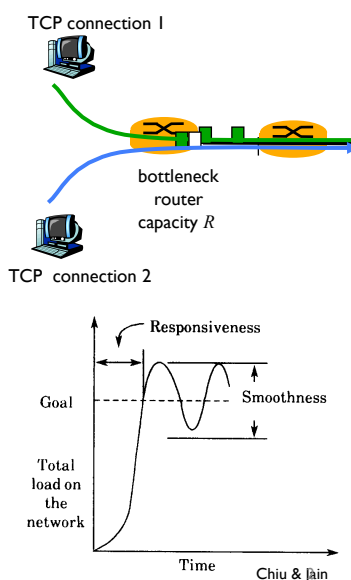
$$b_i > 1, 0 < b_d < 1, a_d = a_i = 0$$

1

Goals of Congestion Control

1. Efficiency: resources are fully utilized
2. Fairness: if k TCP connections share the same bottleneck link of bandwidth R , each connection should get an average rate of R/k
3. Responsiveness: fast convergence, quick adaptation to current capacity
4. Smoothness: little oscillation
 - larger change step increases responsiveness but decreases smoothness
5. Distributed: no (explicit) coordination between sources

Guidelines for congestion control (as in routing):
be skeptical of good news, react fast to bad news

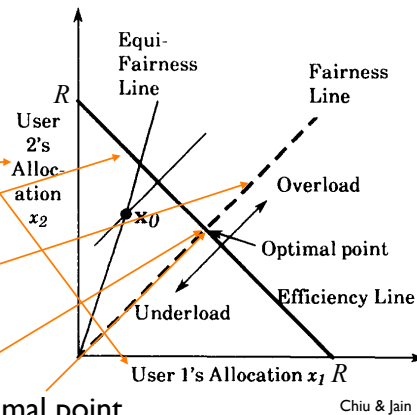


Resource Allocation

View resource allocations as a trajectory through an n -dimensional vector space, one dimension per user

A 2-user resource allocations trajectory:

- x_1, x_2 : the two users' allocations
- Efficiency Line: $x_1 + x_2 = x_i = R$
 - below this line, system is underloaded
 - above, overloaded
- Fairness Line: $x_1 = x_2$
- Optimal point: Efficient and Fair
- Goal of control: to operate at optimal point



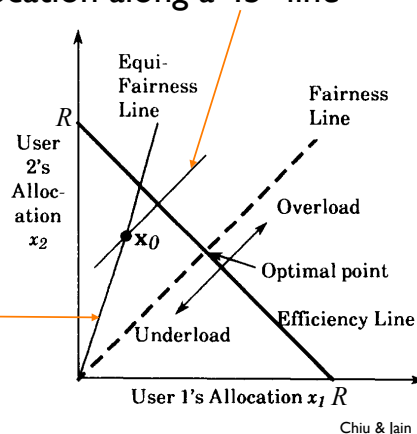
3

Additive/Multiplicative Factors

Additive factor: increasing both users' allocation by the same amount moves an allocation along a 45° line

Multiplicative factor: multiplying both users' allocation by the same factor moves an allocation on a line through the origin (the "equi-fairness," or rather, "equi-unfairness" line)

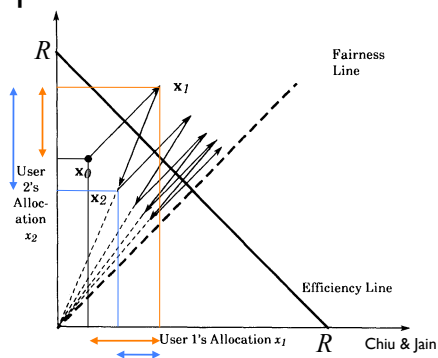
The slope of this line, not position on it, determines fairness



4

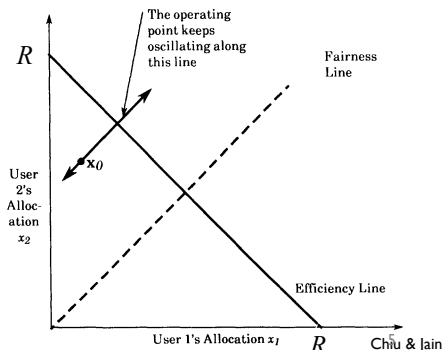
AIMD

**Additive Increase,
Multiplicative Decrease:**
system converges to an
equilibrium near the optimal
point



It can be shown that only AIMD
takes system near optimal point

**Additive Increase,
Additive Decrease:**
system converges to
efficiency, but not to fairness



Issues to Think About

What about short flows? (setting initial $cwnd$)

- most flows are short
- most bytes are in long flows

How does TCP congestion control performs over wireless links?

- packet reordering fools fast retransmit
- loss not a good indicator of congestion

High speed?

- to reach 10 Gbps requires packet loss rate of 1/90 minutes!

Fairness: how do flows with different RTTs share link?

Other Fairness Issues

Fairness and UDP

Multimedia apps often do not use TCP

- do not want rate throttled by congestion control

Instead use UDP:

- pump audio/video at constant rate, tolerate packet loss

TCP-friendly UDP?

Fairness and parallel TCP connections

- nothing prevents app from opening parallel connections between 2 hosts
 - Web browsers do this already
- Example: link of rate R currently supporting 9 connections
 - new app asks for 1 TCP, gets rate $R/10$
 - new app asks for 11 TCPs, gets rate $R/2$!

Problem: on the Internet, there's no incentive to play fair $\Rightarrow$ Tragedy of the Commons

7

Router Architecture and OS

Operating System:

- system resource allocation
- workload (traffic) management

Circuit-switched vs. Packet-switched

- Statistical Multiplexing
- Packet delay analysis

Router architecture:

- Switching and input-, output-queueing
- Scheduling and dropping algorithms

QoS and Virtual Circuit

8

Queue Management

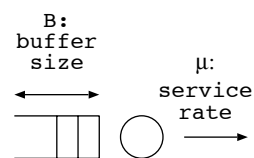
Queue management design issues:

- scheduling discipline: fifo, priority queue, fair queue
 - fairness
 - delay bound
- drop policy: tail drop, head drop, random drop
- cost of operation

Traditionally: FIFO with drop tail

First In First Out (FIFO):

- low cost
- not fair
- no delay bound



9

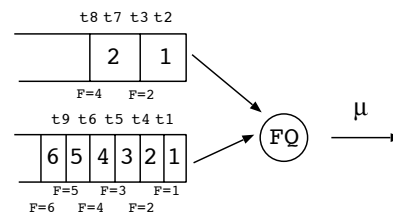
Queue Management

Priority Queue:

- fairness: gives priority to some connections
- delay bound: higher priority connections have lower delay
- but within the same priority, still operates as FIFO
- relatively cheap to operate ($O(\log N)$)

Fair Queueing:

- fair
- bounded delay
- expensive
- connections can be weighted differently (Weighted Fair Queueing, WFQ)



10

Approaches Towards Congestion Control

Network-assisted congestion control:

- routers provide feedback to end systems
- single bit indicating congestion (SNA, DECbit, TCP w/ Explicit Congestion Notification (ECN), ATM)
- explicit rate sender should send at

End-end congestion control:

- no explicit feedback from network
- congestion inferred from end-system observed loss, delay
- approach taken by TCP

11

Packet Dropping Policy

Drop Tail: drop newly arriving packet

Drop Head: long queued packet may be useless to the receiver already

Random drop: drop only on overflow or do early drop?

Traditionally, FIFO with drop tail, however:

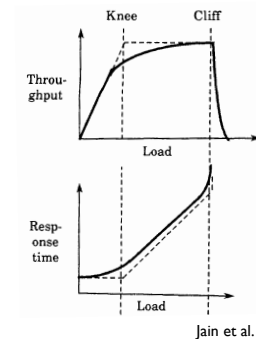
- TCP uses packet drop as congestion signal
- Congestion signal is time delayed in reaching sender
- Problems:
 - synchronized cwnd's
 - unfairness to long-RTT connections

12

Packet Dropping Policy

Goals:

- operate router at capacity “knee”: low delay, high throughput
- keep average queue size low
- but allow for fluctuations in actual queue size to accommodate *bursty traffic* and *transient congestion*



Early drop:

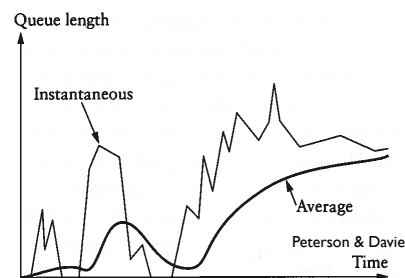
- watch the queue and start dropping before the queue is full
- give time for congestion signal to propagate back to source
- operate queue at knee capacity, prevent overflow

13

Packet Dropping Policy

Random drop:

- flows with more packets in queue have a higher probability of being dropped
- drop packet randomly if *average* queue length is above threshold
 - high *instantaneous* queue length could simply mean transient congestion
 - high *average* queue length indicates persistent congestion
 - what would be a good way to measure average queue length?



14

Random Early Drop

- define a minimum (min_th) and a maximum (max_th) queue occupancy thresholds
- monitor average queue length:

$$\text{aqlen} \leftarrow (1 - w)\text{aqlen} + wq,$$
 where :
 - aqlen : average queue length
 - w : weight
 - q : instantaneous queue length
- provide congestion avoidance by controlling average queue length
 - $\text{aqlen} < \text{min_th}$, no packet is dropped
 - $\text{aqlen} > \text{max_th}$, all packets are dropped
 - between the two thresholds, each packet has some probability $P() \leq \text{MaxP}$ to be dropped, depending on size and duration of queue occupancy

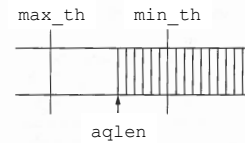


Figure 6.16 RED thresholds on a FIFO queue.

Peterson & Davie

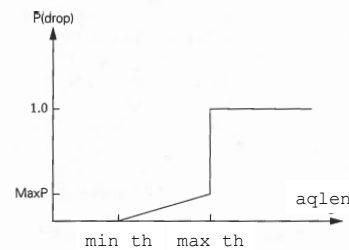


Figure 6.17 Drop probability function for RED.

15

Router Architecture and OS

Circuit-switched vs. Packet-switched

- Statistical Multiplexing
- Packet delay analysis

Router architecture:

- Switching and input-, output-queueing
- Scheduling and dropping algorithms

QoS and Virtual Circuit

16

Datagram Networks

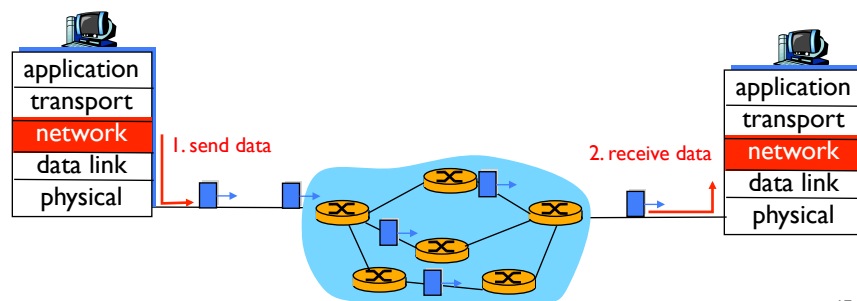
no call setup at network layer

routers: no state about end-to-end connections

- no network-level concept of “connection”

packets forwarded using destination host address

- packets between same source-destination pair may take different paths



17

Pros and Cons of Packet Switching

Advantages: great for bursty data

- resource sharing
- simpler, no call setup

Disadvantages: excessive congestion, packet delay and loss

- protocols needed for reliable data transfer
- congestion control

How to provide circuit-like behavior?

- bandwidth guarantees needed for audio/video apps
- still an unsolved problem

18

Network Service Model

What potential service model an application may ask from the “channel” transporting packets from sender to receiver?

Example services for individual packets:

- guaranteed delivery
- guaranteed delivery with less than 40 msec delay

Example services for a *flow* of packets:

- in-order datagram delivery
- guaranteed minimum bandwidth to flow
- restrictions on changes in inter-packet spacing

19

Virtual Circuits (VC)

Datagram network provides network-layer connectionless service

VC network provides network-layer **connection-oriented** service

Analogous to the transport-layer services, but:

- service: host-to-host
- implementation: in the core

“source-to-destination path behaves much like a telephone circuit”

- performance-wise
- network actions along source-to-destination path
 - call setup, teardown for each call *before* data can flow
 - each packet carries VC identifier (not destination host address)
 - every router on source-destination path maintains “state” for each passing connection
 - link, router resources (bandwidth, buffers) may be *allocated* to VC

20

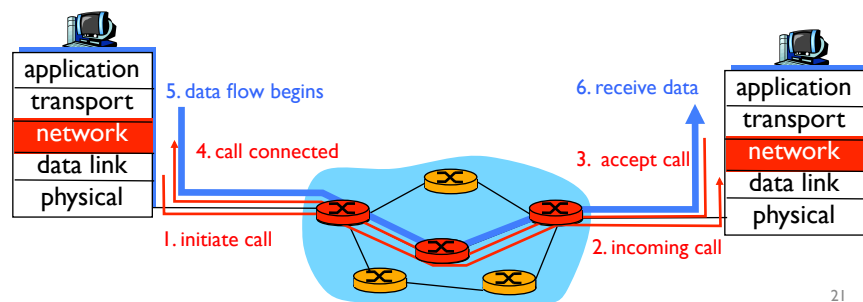
Virtual Circuits

Signalling protocol:

- used to setup, maintain, teardown VC
- e.g., ReSource reSeRvAtion Protocol (RSVP)

A VC consists of:

1. path from source to destination
2. VC numbers, one number for each link along path
3. entries in forwarding tables in routers along path



21

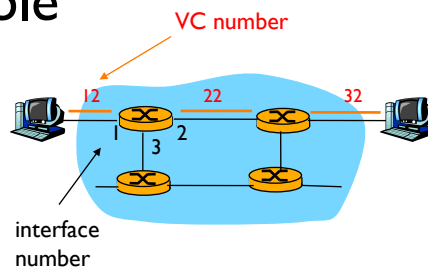
VC Forwarding Table

Packet belonging to VC carries a VC number

VC number must be changed for each link

New VC number obtained from forwarding table

Examples: MPLS, ATM, frame-relay, X.25



Forwarding table on northwest router:

Incoming interface	Incoming VC #	Outgoing interface	Outgoing VC #
1	12	2	22
2	63	1	18
3	7	2	17
1	97	3	87
...	...	...	...

Routers maintain connection state information!

22

Summary: Datagram vs. VC Networks

Datagram network:

- *destination address* in packet determines next hop
- routes may change during session
- analogy: driving, asking directions

Virtual circuit network:

- each packet carries tag (virtual circuit ID), tag determines next hop
- fixed path determined at *call setup time*, remains fixed through call
- *routers maintain per-call state*

23

Summary: Datagram vs. VC Networks

Internet:

- data exchanged among computers
- “elastic” service, no strict timing requirements
- “smart” end systems (computers)
 - can adapt, perform control, error recovery
- simple inside network, complexity at “edge”
- many link types
 - different characteristics
 - uniform service difficult

Virtual Circuits:

- evolved from telephony
- human conversation:
 - strict timing, reliability requirements
 - need for guaranteed service
- “dumb” end systems (telephones)
- complexity inside network

24

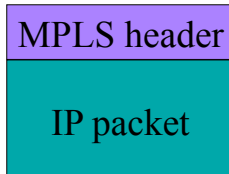
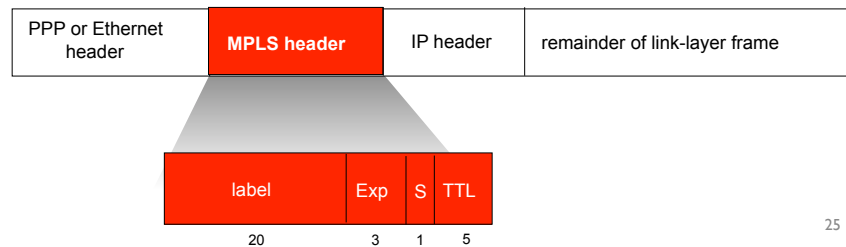
MultiProtocol Label Switching (MPLS)

Initial goal: speed up IP forwarding by using fixed length label (instead of IP address) to do forwarding

- borrowing ideas from Virtual Circuit (VC) approach
- but IP datagram still keeps IP address!

Label Switching

- encapsulate a data packet
 - put an MPLS header in front of the packet
- MPLS header includes a label
- label switching between MPLS-capable routers



25

MPLS Capable “Label-Switched” Routers

Forwards packets to outgoing interface based only on label value (don't inspect IP address)

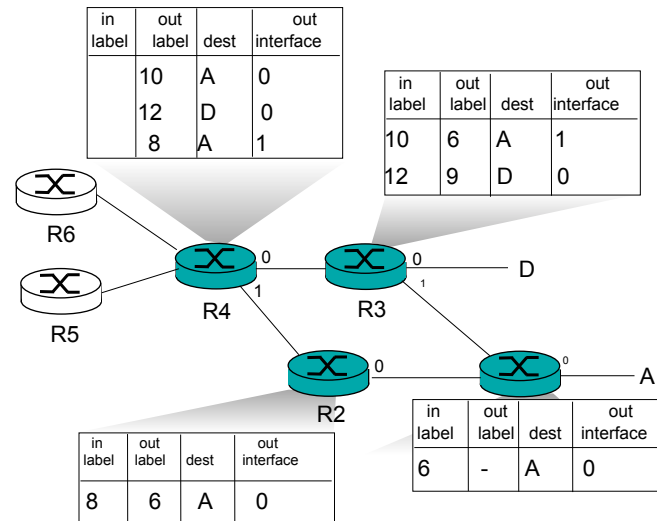
- MPLS forwarding table distinct from IP forwarding tables
- “downstream” MPLS router tells upstream neighbor the label it is using to identify a “flow”
- different labels used for each pair of MPLS routers
- a “flow” can range from a single connection to a pair of address prefixes or aggregated address prefixes, etc.

Signaling protocol needed to set up forwarding

- RSVP-TE
- forwarding possible along paths that IP alone would not allow (e.g., source-specific routing)
- use MPLS for traffic engineering
- must co-exist with IP-only routers

26

MPLS Forwarding Tables



27

Status of MPLS

Deployed in practice

- BGP-free backbone/core
- Virtual Private Networks
- Traffic engineering

Challenges

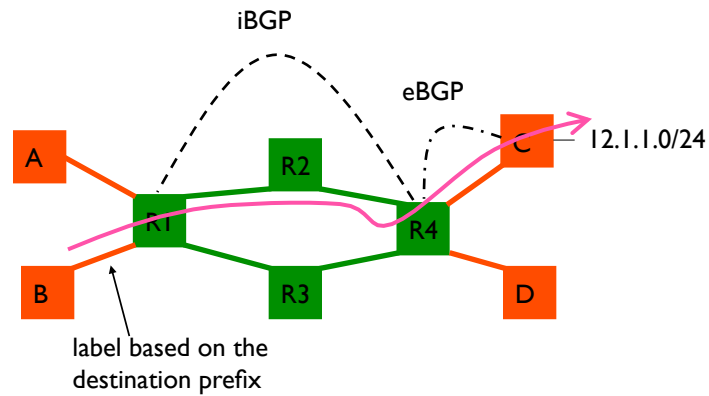
- protocol complexity
- configuration complexity
- difficulty of collecting measurement data

Continuing evolution

- standards
- operational practices and tools

28

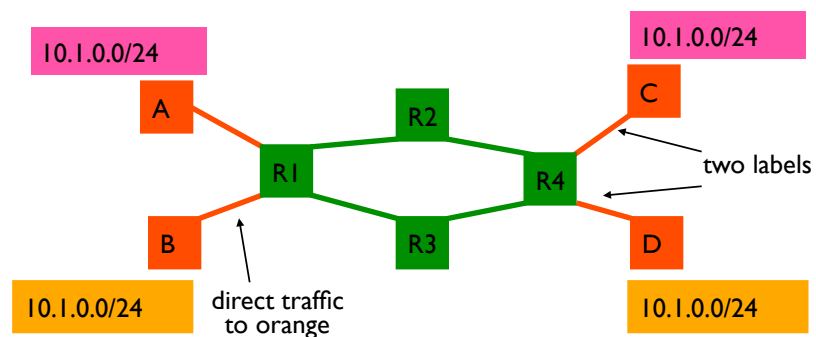
BGP-Free Backbone Core



Routers R2 and R3 don't need to speak BGP

29

VPNs With Private Addresses



MPLS tags can differentiate pink VPN from orange VPN

30