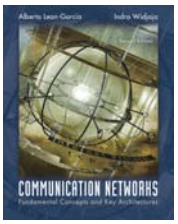
Chapter 7
Packet-Switching Networks

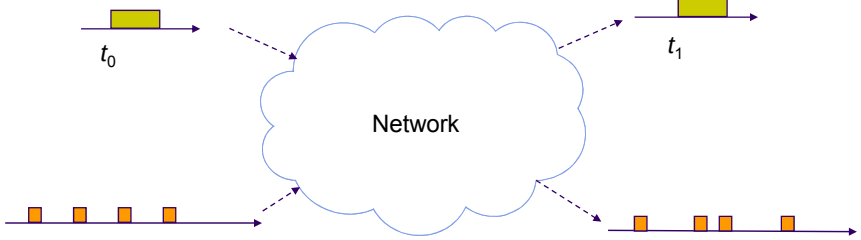


Network Services and Internal Network Operation
Packet Network Topology
Datagrams and Virtual Circuits
Routing in Packet Networks
Shortest Path Routing
ATM Networks
Traffic Management

Network Layer

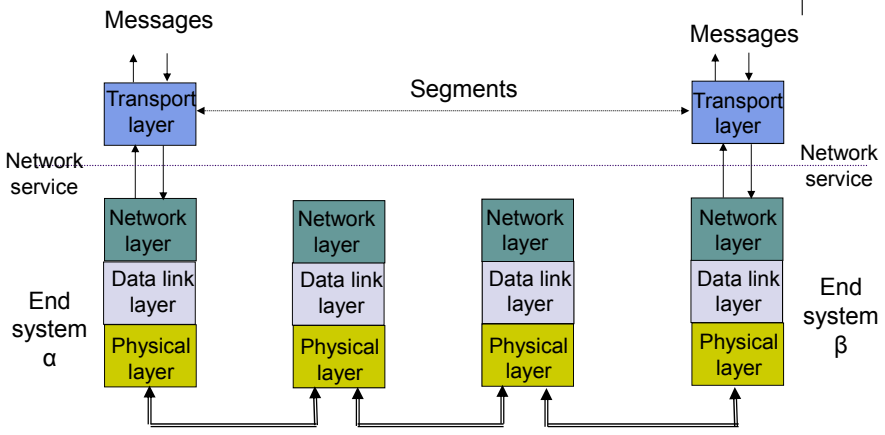
- Network Layer: the most complex layer
  - Requires the coordinated actions of multiple, geographically distributed network elements (switches & routers)
  - Must be able to deal with very large scales
    - Billions of users (people & communicating devices)
  - Biggest Challenges
    - Addressing: where should information be directed to?
    - Routing: what path should be used to get information there?

Packet Switching



- Transfer of information as payload in data packets
- Packets undergo random delays & possible loss
- Different applications impose differing requirements on the transfer of information

Network Service



- Network layer can offer a variety of services to transport layer
- Connection-oriented service or connectionless service
- Best-effort or delay/loss guarantees

Network Service vs. Operation

Network Service

- Connectionless
  - Datagram Transfer
- Connection-Oriented
  - Reliable and possibly constant bit rate transfer

Internal Network Operation

- Connectionless
  - IP
- Connection-Oriented
  - Telephone connection
  - ATM

Various combinations are possible

- Connection-oriented service over Connectionless operation
- Connectionless service over Connection-Oriented operation
- Context & requirements determine what makes sense

Network Layer Functions

What are essentials?

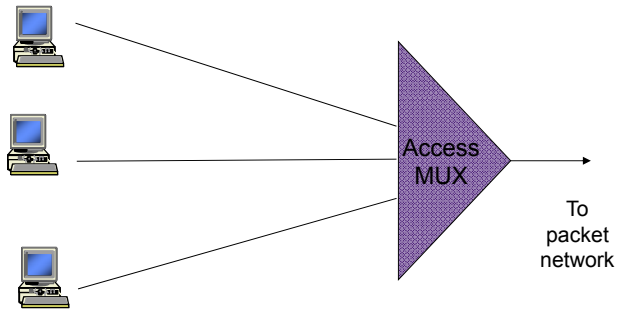
- **Routing**: mechanisms for determining the set of best paths for routing packets requires the collaboration of network elements
- **Forwarding**: transfer of packets from NE inputs to outputs
- **Priority & Scheduling**: determining order of packet transmission in each NE

  Optional: congestion control, segmentation & reassembly, security

End-to-End Packet Network

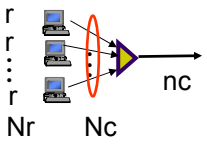
- Packet networks very different than telephone networks
- Individual packet streams are highly bursty
  - Statistical multiplexing is used to concentrate streams
- User demand can undergo dramatic change
  - Peer-to-peer applications stimulated huge growth in traffic volumes
- Internet structure highly decentralized
  - Paths traversed by packets can go through many networks controlled by different organizations
  - No single entity responsible for end-to-end service

Access Multiplexing



- Packet traffic from users multiplexed at access to network into aggregated streams
- DSL traffic multiplexed at DSL Access Mux
- Cable modem traffic multiplexed at Cable Modem Termination System

Oversubscription



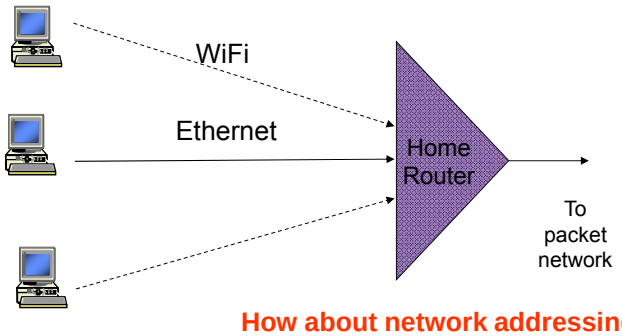
- Access Multiplexer
  - N subscribers connected @ c bps to mux
  - Each subscriber active r/c of time (ave. rate r)
  - Mux has C=nc bps to network
  - Oversubscription rate: N/n
  - Find n so that at most 1% overflow probability

Feasible oversubscription rate increases with size

N	*r/c*	*n*	*N/n*	
10	0.01	1	10	**10 extremely lightly loaded users**
10	0.05	3	3.3	10 very lightly loaded user
10	0.1	4	2.5	10 lightly loaded users
20	0.1	6	3.3	20 lightly loaded users
40	0.1	9	4.4	40 lightly loaded users
100	0.1	18	5.5	100 lightly loaded users

What is the probability that k users transmit at the peak rate?

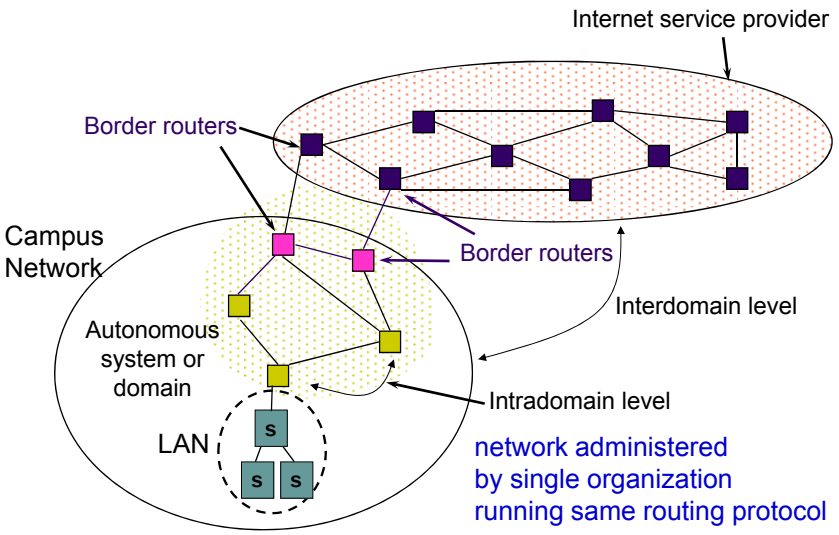
Home LANs



How about network addressing?

- Home Router
  - LAN Access using Ethernet or WiFi (IEEE 802.11)
  - Private IP addresses in Home (192.168.0.x) using Network Address Translation (NAT)
  - Single global IP address from ISP issued using Dynamic Host Configuration Protocol (DHCP)

Connecting to ISP

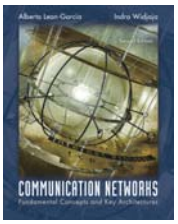


Key Role of Routing

How to get packet from here to there?

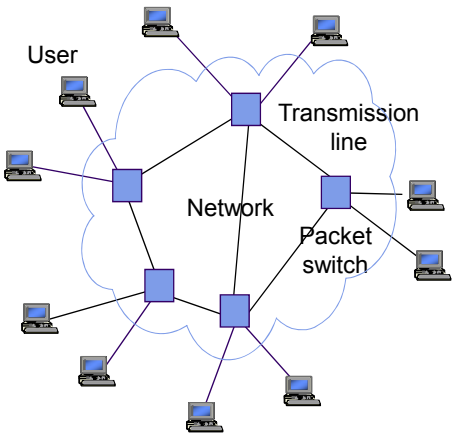
- Decentralized nature of Internet makes routing a major challenge
  - Interior gateway protocols (IGPs) are used to determine routes within a domain
  - Exterior gateway protocols (EGPs) are used to determine routes across domains
  - Routes must be consistent & produce stable flows
- Scalability required to accommodate growth
  - Hierarchical structure of IP addresses essential to keeping size of routing tables manageable

Chapter 7
Packet-Switching Networks



Datagrams and Virtual Circuits

Packet Switching Network



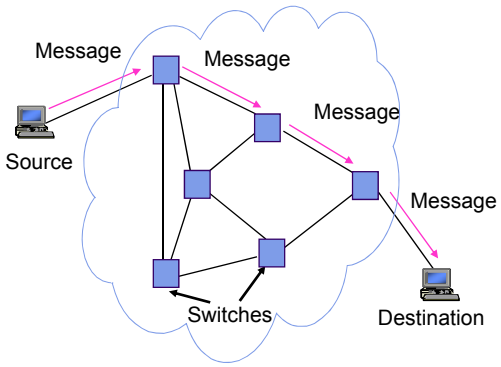
Packet switching network

- Transfers packets between users
- Transmission lines + packet switches (routers)
- Origin in message switching

Two modes of operation:

- Connectionless
- Virtual Circuit

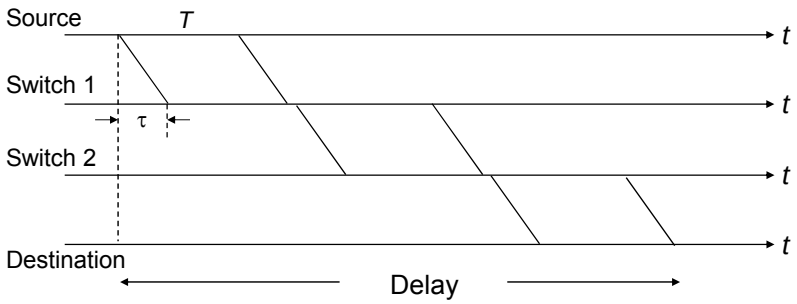
Message Switching



- Message switching invented for telegraphy
- Entire messages multiplexed onto shared lines, stored & forwarded
- Headers for source & destination addresses
- Routing at message switches
- Connectionless

- Transmission delay vs. propagation delay
  - Transmit a 1000B from LA to DC via a 1Gbps network, signal speed 200Km/sec.

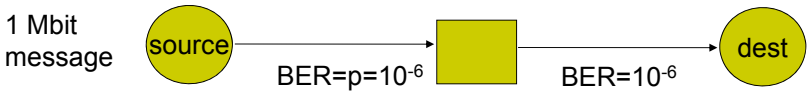
Message Switching Delay



Minimum delay = $3\tau + 3T$

Additional queueing delays possible at each link

Long Messages vs. Packets



How many bits need to be transmitted to deliver message?

- Approach 1: send 1 Mbit message
- Probability message arrives correctly

$$P_c = (1-10^{-6})^{10^6} \approx e^{-10^6 10^{-6}} = e^{-1} \approx 1/3$$

- On average it takes about 3 transmissions/hop
- Total # bits transmitted ≈ 6 Mbits

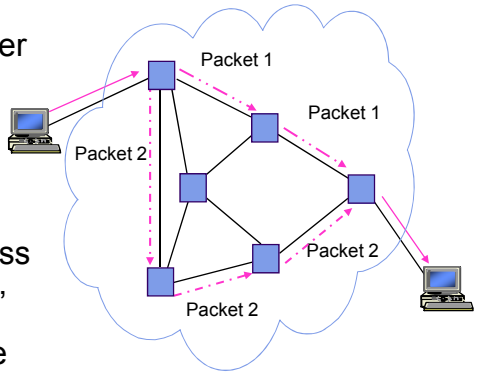
- Approach 2: send 10 100-kbit packets
- Probability packet arrives correctly

$$P'_c = (1-10^{-6})^{10^5} \approx e^{-10^5 10^{-6}} = e^{-0.1} \approx 0.9$$

- On average it takes about 1.1 transmissions/hop
- Total # bits transmitted ≈ 2.2 Mbits

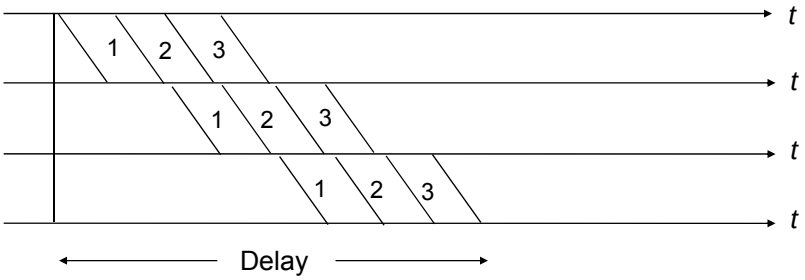
Packet Switching - Datagram

- Messages broken into smaller units (packets)
- Source & destination addresses in packet header
- Connectionless, packets routed independently (datagram)
- Packet may arrive out of order
- Pipelining of packets across network can reduce delay, increase throughput
- Lower delay that message switching, suitable for interactive traffic



Packet Switching Delay

Assume three packets corresponding to one message traverse same path

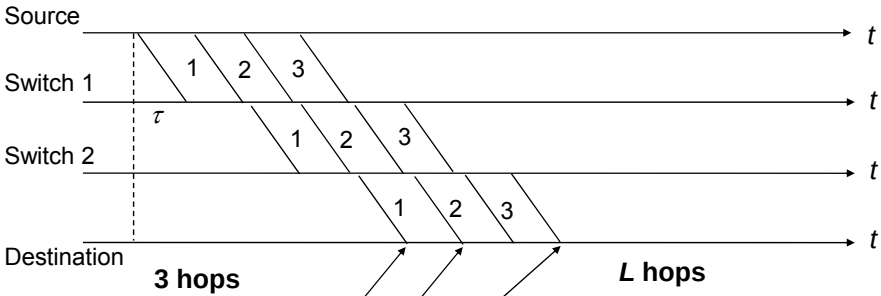


Minimum Delay = $3\tau + 5(T/3)$ (single path assumed)

Additional queueing delays possible at each link

Packet pipelining enables message to arrive sooner

Delay for k-Packet Mess. over L Hops



3 hops

$3\tau + 2(T/3)$ first bit received

$3\tau + 3(T/3)$ first bit released

$3\tau + 5(T/3)$ last bit released

***L* hops**

$L\tau + (L\text{-}1)P$ first bit received

$L\tau + LP$ first bit released

$L\tau + LP + (k\text{-}1)P$ last bit released

where $T = k\,P$

Routing Tables in Datagram Networks

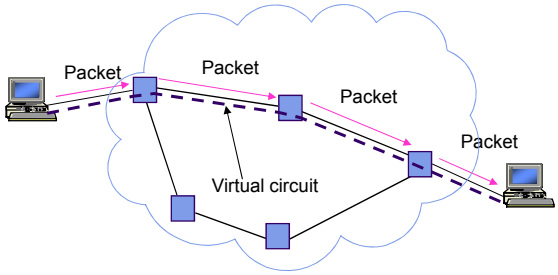
Destination address	Output port
0785	7
1345	12
1566	6
2458	12

- Route determined by table lookup
- Routing decision involves finding next hop in route to given destination
- Routing table has an entry for each destination specifying output port that leads to next hop
- Size of table becomes impractical for very large number of destinations

Example: Internet Routing

- Internet protocol uses datagram packet switching *across networks*
  - Networks are treated as data links
- Hosts have two-part IP address:
  - Network address + Host address
- Routers do table lookup on network address
  - This reduces size of routing table
- In addition, network addresses are assigned so that they can also be aggregated
  - Discussed as CIDR in Chapter 8

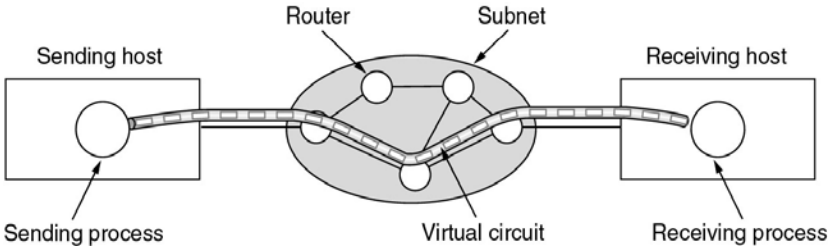
Packet Switching – Virtual Circuit



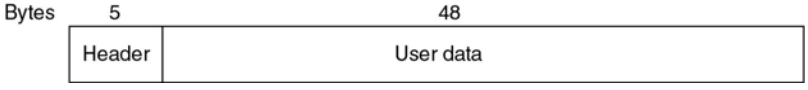
- Call set-up phase sets ups pointers in fixed path along network
- All packets for a connection follow the same path
- Abbreviated header identifies connection on each link
- Packets queue for transmission
- Variable bit rates possible, negotiated during call set-up
- Delays variable, cannot be less than circuit switching

ATM Virtual Circuits

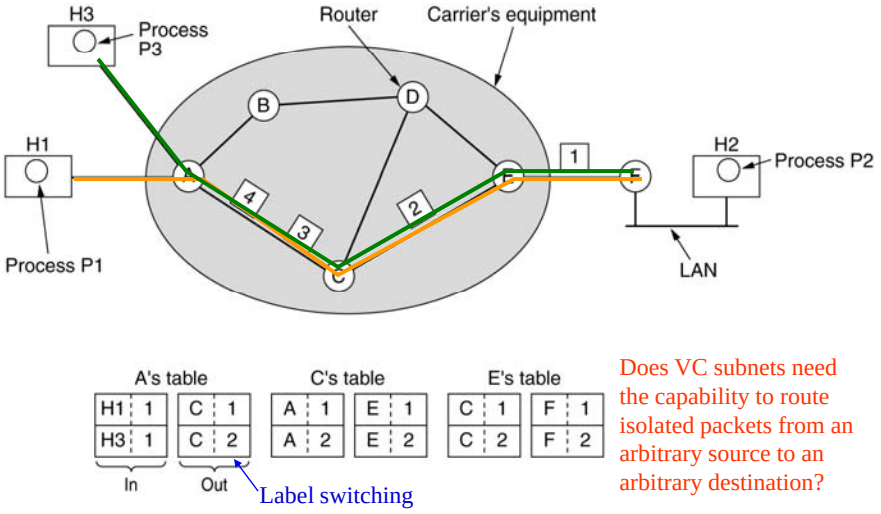
- A VC is a connection with resources reserved.



- A Cell is a small and fixed-size packet, delivered in order.

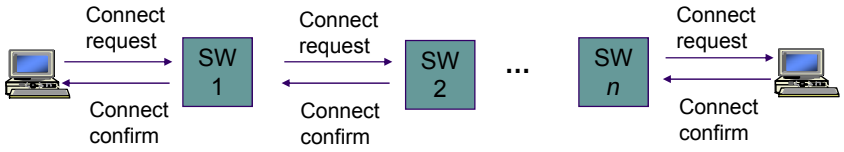


Routing in VC Subnet



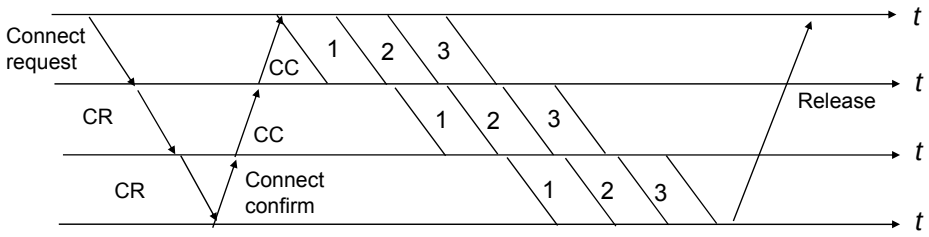
Does VC subnets need the capability to route isolated packets from an arbitrary source to an arbitrary destination?

Connection Setup



- Signaling messages propagate as route is selected
- Signaling messages identify connection and setup tables in switches
- Typically a connection is identified by a *local* tag, Virtual Circuit Identifier (VCI)
- Each switch only needs to know how to relate an incoming tag in one input to an outgoing tag in the corresponding output
- Once tables are setup, packets can flow along path

Connection Setup Delay



- Connection setup delay is incurred before any packet can be transferred
- Delay is acceptable for sustained transfer of large number of packets
- This delay may be unacceptably high if only a few packets are being transferred

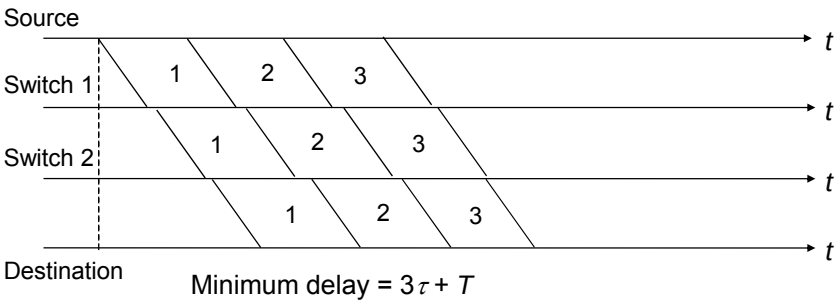
Virtual Circuit Forwarding Tables

Input VCI	Output port	Output VCI
12	13	44
15	15	23
27	13	16
58	7	34

- Each input port of packet switch has a forwarding table
- Lookup entry for per-link/port VCI of incoming packet
- Determine output port (next hop) and insert VCI for next link
- Very high speeds are possible (HW-based)
- Table can also include priority or other information about how packet should be treated

What are key pros & cons of VC switching vs. datagram switching?

Cut-Through switching



Minimum delay = $3\tau + T$

- Some networks perform error checking on header only, so packet can be forwarded as soon as header is received & processed
- Delays reduced further with cut-through switching

Message vs. Packet Min. Delay

- Message:

  $L\ \tau\ +\ L\ T$ $= L\ \tau + (L - 1)\ T\ +\ T$

- Packet

  $L\ \tau + L\ P + (k - 1)\ P$ $= L\ \tau + (L - 1)\ P\ +\ T$

- Cut-Through Packet (Immediate forwarding after header) $= L\ \tau + T$

Above neglect header processing delays

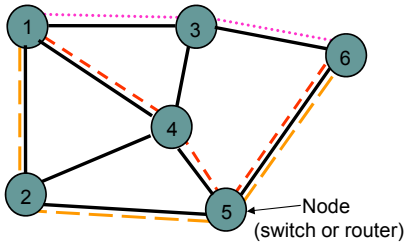
Comparison of VC and Datagram Subnets

Issue	Datagram subnet	Virtual-circuit subnet
Circuit setup	Not needed	Required
Addressing	Each packet contains the full source and destination address	Each packet contains a short VC number
State information	Routers do not hold state information about connections	Each VC requires router table space per connection
Routing	Each packet is routed independently	Route chosen when VC is set up; all packets follow it
Effect of router failures	None, except for packets lost during the crash	All VCs that passed through the failed router are terminated
Quality of service	Difficult	Easy if enough resources can be allocated in advance for each VC
Congestion control	Difficult	Easy if enough resources can be allocated in advance for each VC

Chapter 7
Packet-Switching Networks

Routing in Packet Networks

Routing in Packet Networks



- Three possible (loopfree) routes from 1 to 6:
  - 1-3-6, 1-4-5-6, 1-2-5-6
- Which is "best"? **What is the objective function for optimization?**
  - Min delay? Min hop? Max bandwidth? Min cost? Max reliability?

Creating the Routing Tables

- Need information on state of links
  - Link up/down; congested; delay or other metrics
- Need to distribute link state information using a routing protocol
  - What information is exchanged? How often?
  - How to exchange with neighbors?
- Need to compute routes based on information
  - Single metric; multiple metrics
  - Single route; alternate routes

Routing Algorithm Requirements

- Correctness
- Responsiveness
  - Topology or bandwidth changes, congestion
- Optimality
  - Resource utilization, path length
- Robustness
  - Continues working under high load, congestion, faults, equipment failures, incorrect implementations
- Simplicity
  - Efficient software implementation, reasonable processing load
- Fairness

Centralized vs Distributed Routing

- Centralized Routing
  - All routes determined by a central node
  - All state information sent to central node
  - Problems adapting to frequent topology changes
  - What is the problem? Does not scale
- Distributed Routing
  - Routes determined by routers using distributed algorithm
  - State information exchanged by routers
  - Adapts to topology and other changes
  - Better scalability, but maybe inconsistent due to loops

Static vs Dynamic Routing

- Static Routing
  - Set up manually, do not change; requires administration
  - Works when traffic predictable & network is simple
  - Used to override some routes set by dynamic algorithm
  - Used to provide default router
- Dynamic Routing
  - Adapt to changes in network conditions
  - Automated
  - Calculates routes based on received updated network state information

Flooding

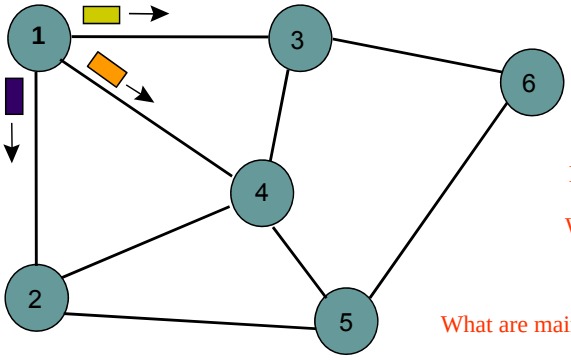
Send a packet to all nodes in a network

- Useful in starting up network or broadcast
- No routing tables available
- Need to broadcast packet to all nodes (e.g. to propagate link state information)

Approach

- Send packet on all ports except one where it arrived
- Exponential growth in packet transmissions

Flooding Example



Is flooding static or adaptive?

What is the major problem?

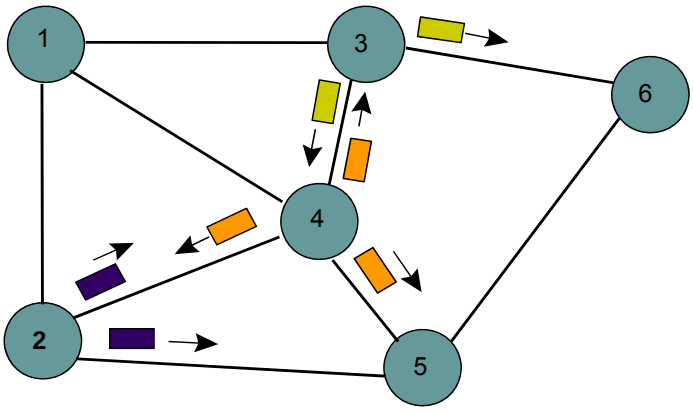
How to handle the problem?

What are main nice properties of flooding?

How flooding can be terminated?

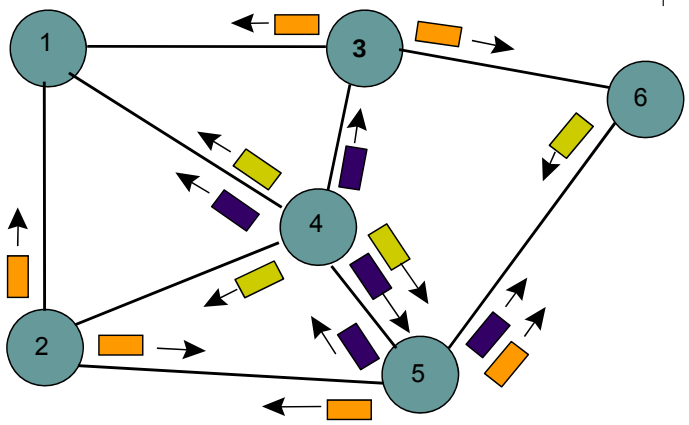
Duplicates (infinite number due to loops)
Hop Count; Sequence number with a counter per a source router.
Robustness; always follow shortest path
TTL is a way to terminate flooding.

Flooding Example



Flooding is initiated from Node 1: Hop 2 transmissions

Flooding Example



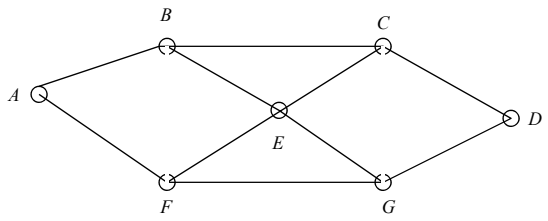
Flooding is initiated from Node 1: Hop 3 transmissions

Limited Flooding

- Time-to-Live field in each packet limits number of hops to certain diameter
- Each switch adds its ID before flooding; discards repeats
- Source puts sequence number in each packet; switches records source address and sequence number and discards repeats

Limited Flooding Example

- Suppose the following network uses flooding as the routing algorithm. If a packet sent by A to D has a maximum hop of 3, list all the routes it will take. Also tell how many hops worth of bandwidth it consumes. Assume the bandwidth weight of the lines is the same.



Shortest Paths & Routing

- Many possible paths connect any given source and to any given destination
- Routing involves the selection of the path to be used to accomplish a given transfer
- Typically it is possible to attach a cost or distance to a link connecting two nodes
- Routing can then be posed as a shortest path problem

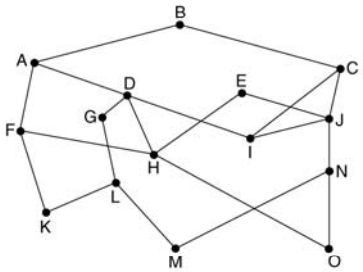
Routing Metrics

Means for measuring desirability of a path

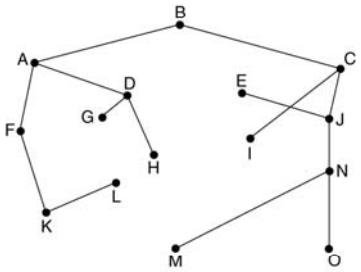
- Path Length = sum of costs or distances
- Possible metrics
  - Hop count: rough measure of resources used
  - Reliability: link availability
  - Delay: sum of delays along path; complex & dynamic
  - Bandwidth: "available capacity" in a path
  - Load: Link & router utilization along path
  - Cost: $$$

An Example: Sink Tree

- Sink tree: minimum # of hops to a root.



(a) A subnet.



(b) A *sink tree* for router B.

Q1: must a sink tree be unique? An example?

Q2: each packet will be delivered within a finite # of hops?

Shortest Path Approaches

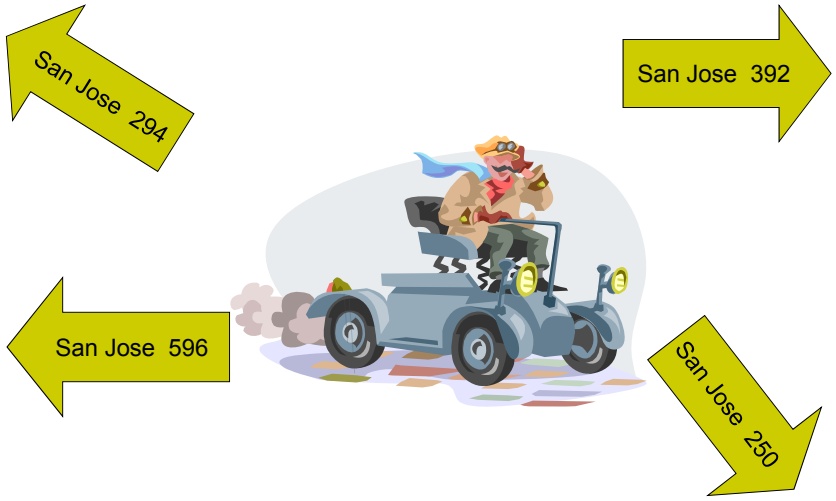
Distance Vector Protocols

- Neighbors exchange list of distances to destinations
- Best next-hop determined for each destination
- Ford-Fulkerson (distributed) shortest path algorithm

Link State Protocols

- Link state information flooded to all routers
- Routers have complete topology information
- Shortest path (& hence next hop) calculated
- Dijkstra (centralized) shortest path algorithm

Distance Vector
Do you know the way to San Jose?



Distance Vector

Local Signpost

- Direction
- Distance

Routing Table

For each destination list:

- Next Node
- Distance

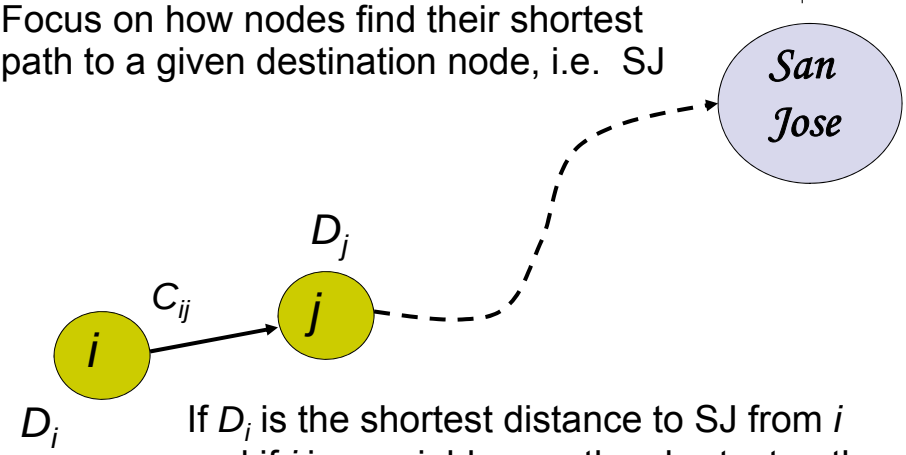
dest	next	dist

Table Synthesis

- Neighbors exchange table entries
- Determine current best next hop
- Inform neighbors
  - Periodically
  - After changes

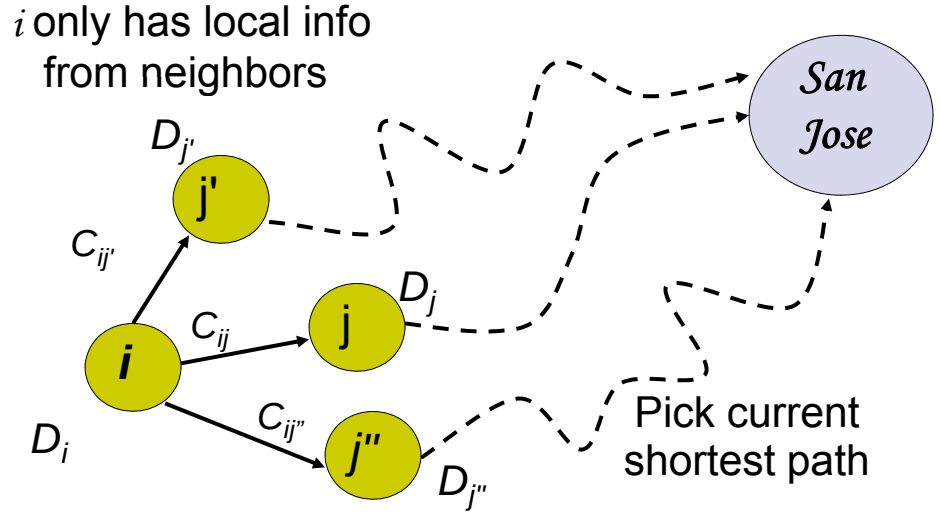
Shortest Path to SJ

Focus on how nodes find their shortest path to a given destination node, i.e. SJ

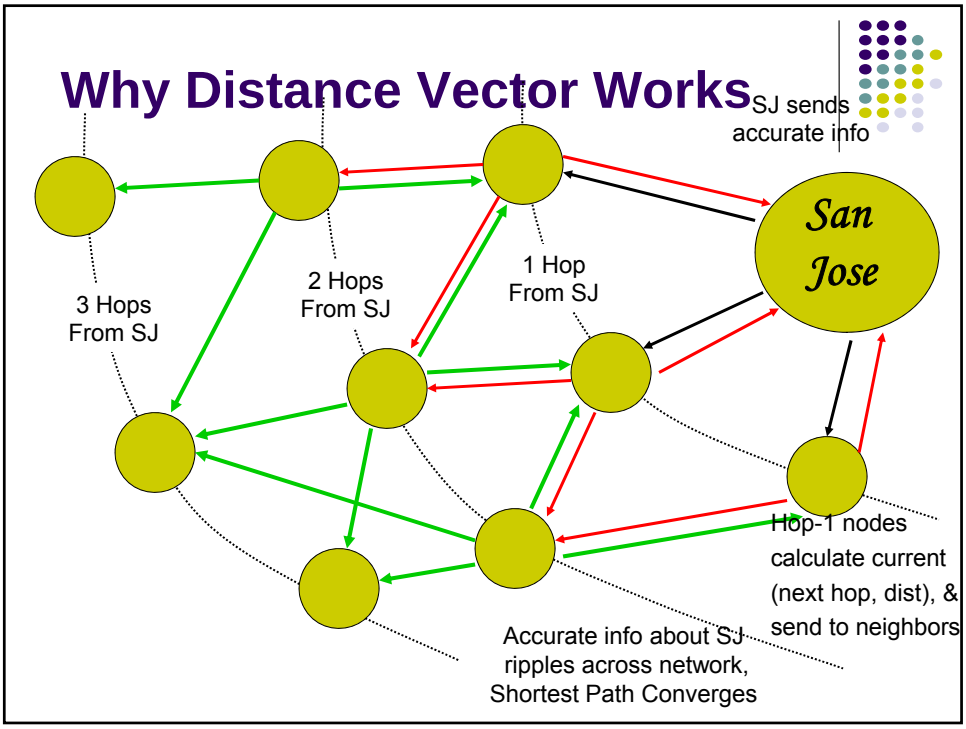


If D_i is the shortest distance to SJ from *i* and if *j* is a neighbor on the shortest path, then $D_i = C_{ij} + D_j$

But we don't know the shortest paths

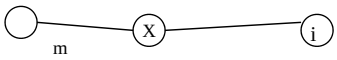


Why Distance Vector Works

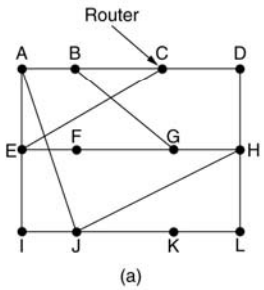


Distance Vector Routing (RIP)

- Routing operates by having each router maintain a vector table giving the best known distance to each destination and which line to use to get there. The tables are updated by exchanging information with the neighbors.
- Vector table: one entry for each router in the subnet; each entry contains two parts: preferred outgoing line to use for that destination and an estimate of the time or distance to the destination.
- The router is assumed to know the distance/cost to each neighbor and update the vector table *periodically* by changing it with neighbors.
  - # hops
  - Delay (ECHO)
  - Capacity
  - Congestion



An Example



(a)

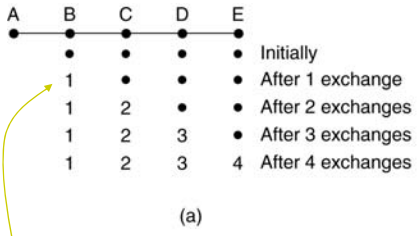
To	A	I	H	K	New estimated delay from J	Line
A	0	24	20	21	8	A
B	12	36	31	28	20	A
C	25	18	19	36	28	I
D	40	27	8	24	20	H
E	14	7	30	22	17	I
F	23	20	19	40	30	I
G	18	31	6	31	18	H
H	17	20	0	19	12	H
I	21	0	14	22	10	I
J	9	11	7	10	0	–
K	24	22	22	0	6	K
L	29	33	9	9	15	K
	JA delay is 8	JI delay is 10	JH delay is 12	JK delay is 6	New routing table for J	

Vectors received from J's four neighbors

(a) A subnet. (b) Input from A, I, H, K, and the new routing table for J.

Count-to-Infinity Problem

- It converges to the correct answer quickly to good news but slowly to bad news.

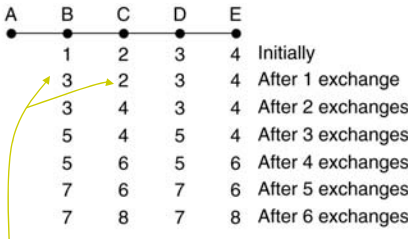


(a)

B knows A is 1 hop away while all other routers still think A is down, why?

What is the spreading rate of good news?

How many exchanges needed in a N-hop subnet?



(b)

Does B know that C's path runs through B?

Why spreading rate of bad news so slow?

What is the core problem?

Problem: Bad News Travels Slowly

Remedies

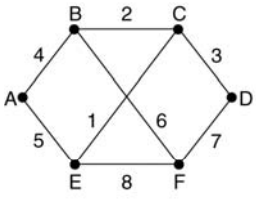
- Split Horizon
  - Do not report route to a destination to the neighbor from which route was learned
- Poisoned Reverse
  - Report route to a destination to the neighbor from which route was learned, but with infinite distance
  - Breaks erroneous direct loops immediately
  - Does not work on some indirect loops

Link-State Algorithm

- Basic idea: two stage procedure
  - Each source node gets a map of all nodes and link metrics (link state) of the entire network
    - Learning who the neighbors are and what are delay to them
    - Construct a link state packet, and deliver it to others
  - Find the shortest path on the map from the source node to all destination nodes; Dijkstra's algorithm
- Broadcast of link-state information
  - Every node *i* in the network broadcasts to every other node in the network:
    - ID's of its neighbors: $\mathcal{N}_i$=set of neighbors of i
    - Distances to its neighbors: $\{C_{ij} \mid j \in N_i\}$
  - Flooding is a popular method of broadcasting packets

Building Link State Packets

- A state packet starts with the ID of the sender, a seq#, age, and a list of neighbors with delay information.



(a) A subnet.

Link State Packets

A	B	C	D	E	F
Seq.	Seq.	Seq.	Seq.	Seq.	Seq.
Age	Age	Age	Age	Age	Age
B 4	A 4	B 2	C 3	A 5	B 6
E 5	C 2	D 3	F 7	C 1	D 7
	F 6	E 1		F 8	E 8

(b) The link state packets for this subnet.

When to build the link state packets?

Periodically, or when significant event occurs.

Distributing the Link State Packets

- Flooding is used to distribute the link state packets.

What is the major problem with flooding?

How to handle the problem?

(source router, sequence number)

How to make the sequence number unique?

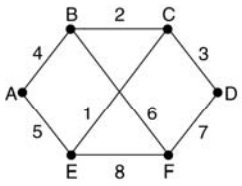
32-bit sequence number

What happens if a router crashes, losing its track, and starts again?

What happens if sequence number is corrupted, say 65,540, not 4.

***Age* field (in sec; usually a packet comes in 10 sec.)**
All packets should be ACKed.

A Packet Buffer



(a)

Link State Packets

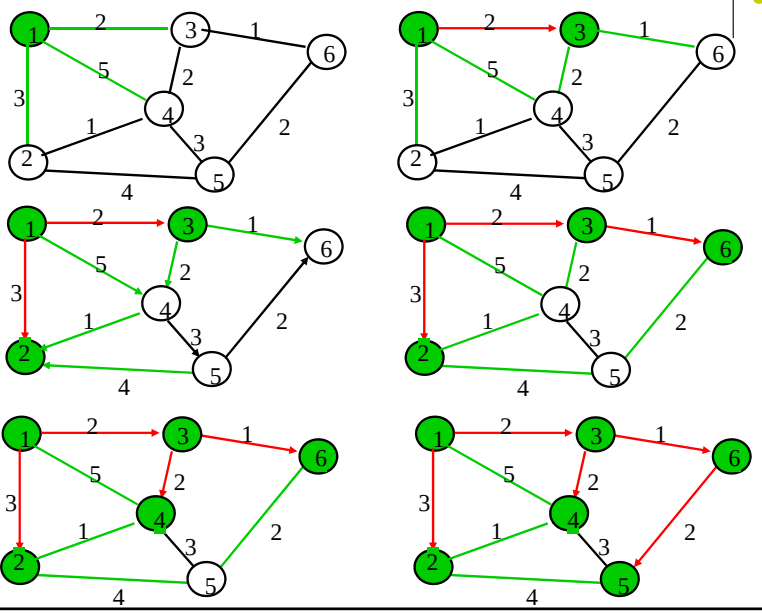
A	B	C	D	E	F
Seq.	Seq.	Seq.	Seq.	Seq.	Seq.
Age	Age	Age	Age	Age	Age
B 4	A 4	B 2	C 3	A 5	B 6
E 5	C 2	D 3	F 7	C 1	D 7
	F 6	E 1		F 8	E 8

(b)

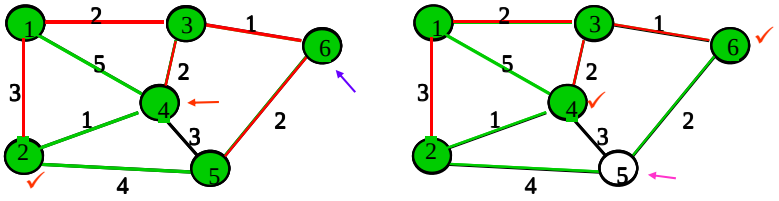
flooding

Source	Seq.	Age	Send flags A	Send flags C	Send flags F	ACK flags A	ACK flags C	ACK flags F	Data
A	21	60	0	1	1	1	0	0	
F	21	60	1	1	0	0	0	1	
E	21	59	0	1	0	1	0	1	
C	20	60	1	0	1	0	1	0	
D	21	59	1	0	0	0	1	1	

Shortest Paths in Dijkstra's Algorithm



Execution of Dijkstra's algorithm



Iteration	N	D_2	D_3	D_4	D_5	D_6
Initial	{1}	3	2 ✓	5	∞	∞
1	{1,3}	3 ✓	2	4	∞	3
2	{1,2,3}	3	2	4	7	3 ✓
3	{1,2,3,6}	3	2	4 ✓	5	3
4	{1,2,3,4,6}	3	2	4	5 ✓	3
5	{1,2,3,4,5,6}	3	2	4	5	3

Dijkstra's algorithm

- N: set of nodes for which shortest path already found
- Initialization: (*Start with source node s)*
  - $N = \{s\}$, $D_s = 0$, "s is distance zero from itself"
  - $D_j = C_{sj}$ for all $j \neq s$, distances of directly-connected neighbors
- Step A: (*Find next closest node i*)
  - Find $i \notin N$ such that
  - D_i = min Dj for $j \notin N$
  - Add i to N
  - If N contains all the nodes, stop
- Step B: (*update minimum costs)*
  - For each node $j \notin N$
  - D_j = min (D_j, $D_i + C_{ij}$) ← *Minimum distance from s to j through node **i** in N*
  - Go to Step A

Reaction to Failure

- **If a link fails,**
  - Router sets link distance to infinity & floods the network with an update packet
  - All routers immediately update their link database & recalculate their shortest paths
  - Recovery very quick
- **But watch out for old update messages**
  - Add time stamp or sequence # to each update message
  - Check whether each received update message is new
  - If new, add it to database and broadcast
  - If older, send update message on arriving link

Why is Link State Better?

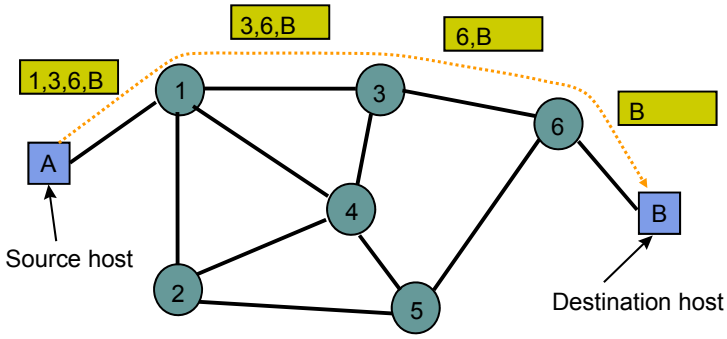
- Fast, loopless convergence
- Support for precise metrics, and multiple metrics if necessary (throughput, delay, cost, reliability)
- Support for multiple paths to a destination
  - algorithm can be modified to find best two paths
  - More flexible, e.g., source routing

What is the memory required to store the input data for a subnet with *n* routers – each of them has *k* neighbors? But not critical today!

Source Routing vs. H-by-H

- Source host selects path that is to be followed by a packet
  - Strict: sequence of nodes in path inserted into header
  - Loose: subsequence of nodes in path specified
- Intermediate switches read next-hop address and remove address
  - Or maintained for the reverse path
- Source routing allows the host to control the paths that its information traverses in the network
- Potentially the means for customers to select what service providers they use

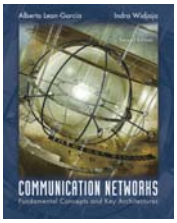
Example



How path learned?

Source host needs link state information

Chapter 7
Packet-Switching Networks



Traffic Management

Packet Level

Flow Level

Flow-Aggregate Level

Traffic Management

Vehicular traffic management

- Traffic lights & signals control flow of traffic in city street system
- Objective is to maximize flow with tolerable delays
- Priority Services
  - Police sirens
  - Cavalcade for dignitaries
  - Bus & High-usage lanes
  - Trucks allowed only at night

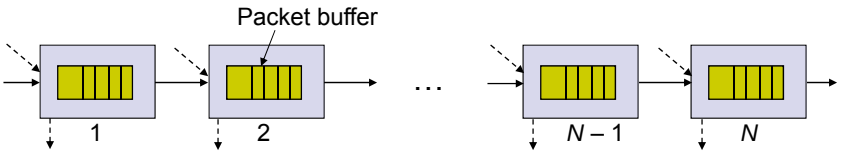
Packet traffic management

- Multiplexing & access mechanisms to control flow of packet traffic
- Objective is make efficient use of network resources & deliver QoS
- Priority
  - Fault-recovery packets
  - Real-time traffic
  - Enterprise (high-revenue) traffic
  - High bandwidth traffic

Time Scales & Granularities

- Packet Level
  - Queueing & scheduling at multiplexing points
  - Determines relative performance offered to packets over a short time scale (microseconds)
- Flow Level
  - Management of traffic flows & resource allocation to ensure delivery of QoS (milliseconds to seconds)
  - Matching traffic flows to resources available; congestion control
- Flow-Aggregate Level
  - Routing of aggregate traffic flows across the network for efficient utilization of resources and meeting of service levels
  - “Traffic Engineering”, at scale of minutes to days

End-to-End QoS



- A packet traversing network encounters delay and possible loss at various multiplexing points
- End-to-end performance is accumulation of per-hop performances

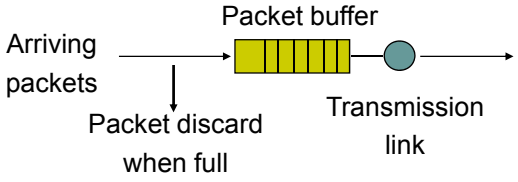
How to keep end-to-end delay under some upper bound? Jitter, loss?

Scheduling & QoS

- End-to-End QoS & Resource Control
  - Buffer & bandwidth control → Performance
  - Admission control to regulate traffic level
- Scheduling Concepts
  - fairness/isolation
  - priority, aggregation,
- Fair Queueing & Variations
  - WFQ, PGPS
- Guaranteed Service
  - WFQ, Rate-control
- Packet Dropping
  - aggregation, drop priorities

72

FIFO Queueing



- All packet flows share the same buffer
- Transmission Discipline: First-In, First-Out
- Buffering Discipline: Discard arriving packets if buffer is full (Alternative: random discard; pushout head-of-line, i.e. oldest, packet)

How about aggressiveness vs. fairness?

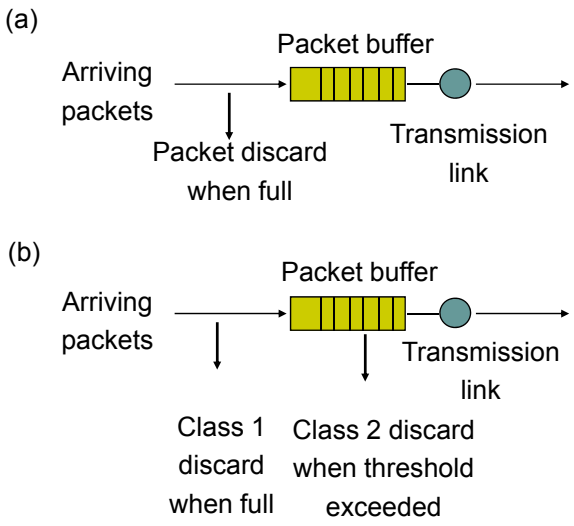
73

FIFO Queueing

- Cannot provide differential QoS to different packet flows
  - Different packet flows interact strongly
- Statistical delay guarantees via load control
  - Restrict number of flows allowed (connection admission control)
  - Difficult to determine performance delivered
- Finite buffer determines a maximum possible delay
- Buffer size determines loss probability
  - But depends on arrival & packet length statistics
- Variation: packet enqueueing based on queue thresholds
  - some packet flows encounter blocking before others
  - higher loss, lower delay

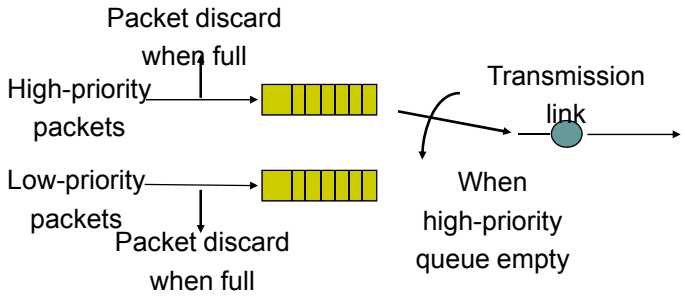
74

FIFO w/o and w/ Discard Priority



75

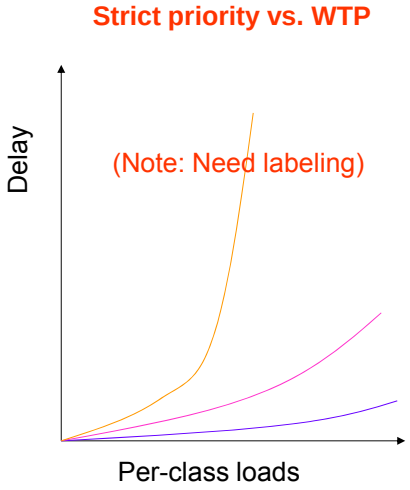
HOL Priority Queueing



- High priority queue serviced until empty
- High priority queue has lower waiting time
- Buffers can be dimensioned for different loss probabilities
- Surge in high priority queue can cause low priority queue to saturate

76

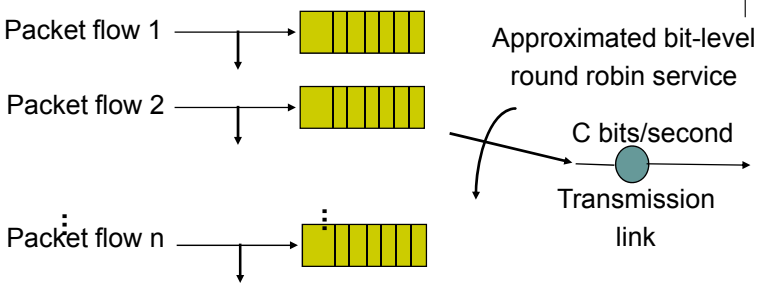
HOL Priority Features



- Provides differential QoS
- Pre-emptive priority: lower classes invisible
- Non-preemptive priority: lower classes impact higher classes through residual service times
- High-priority classes can hog all of the bandwidth & starve lower priority classes
- Need to provide some isolation between classes

77

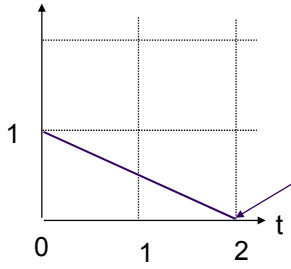
Fair Queuing / Generalized Processor Sharing (GPS)



- Each flow has its own logical queue: prevents hogging; allows differential loss probabilities
- C bits/sec allocated equally among non-empty queues
  - transmission rate = C / n(t), where n(t)=# non-empty queues
- Idealized system assumes *fluid flow* from queues
- Implementation requires approximation: simulate fluid system; sort packets according to completion time in ideal system

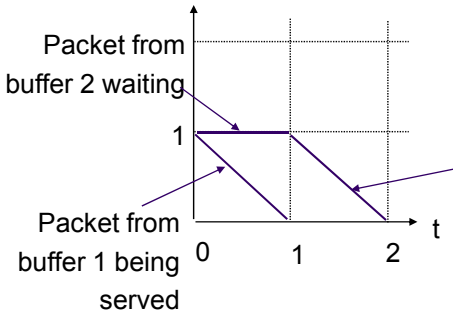
78

Fair Queuing – Example 1



Fluid-flow system: both packets served at rate ½ (overall rate : 1 unit/second)

Both packets complete service at t = 2

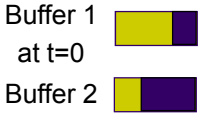


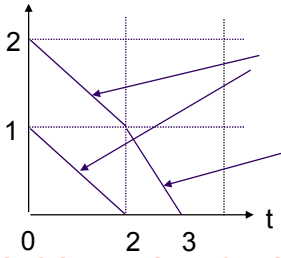
Packet-by-packet system: buffer 1 served first at rate 1; then buffer 2 served at rate 1.

Packet from buffer 2 being served

79

Fair Queuing – Example 2



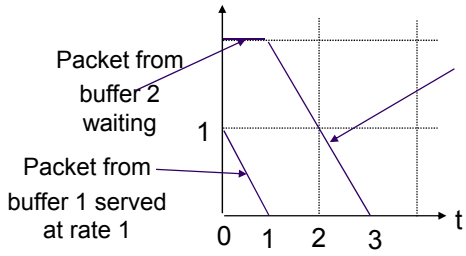


Fluid-flow system: both packets served at rate 1/2

Packet from buffer 2 served at rate 1

Service rate = reciprocal of the number of active buffers at the time.

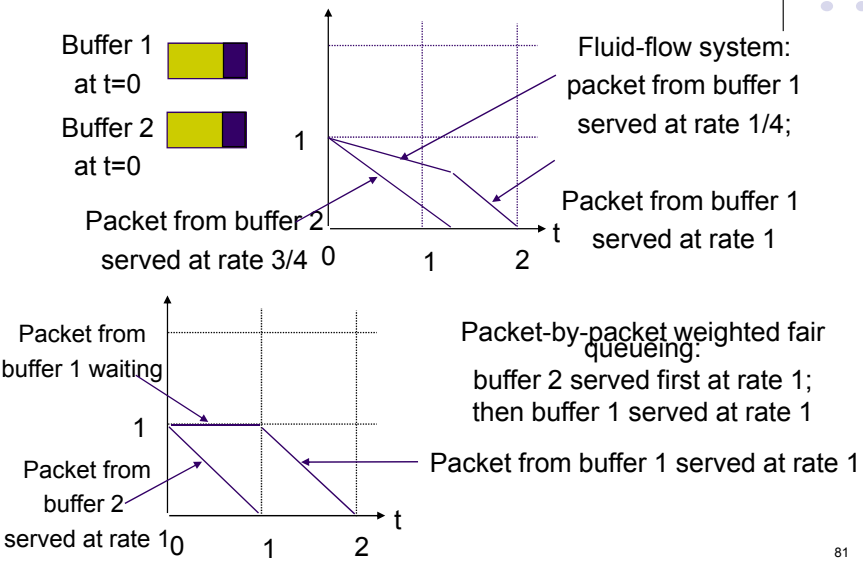
*** Within a buffer, FIFO still though!**



Packet-by-packet fair queueing: buffer 2 served at rate 1

80

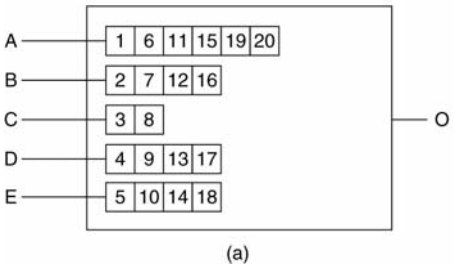
Weighted Fair Queueing (WFQ)



81

Example

- FIFO -> Fair queueing
  - packet-by-packet vs. bit-by-bit



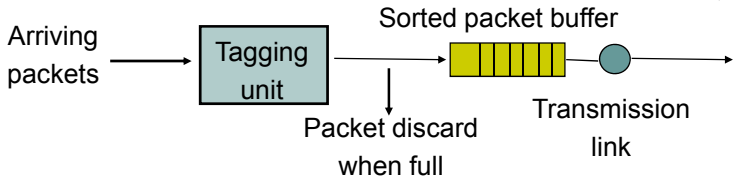
Packet	Finishing time
C	8
B	16
D	17
E	18
A	20

(b)

(a) A router with five packets queued for line O.

(b) Finishing times for the five packets.

Packetized GPS/WFQ



- Compute packet completion time in ideal system
  - add tag to packet
  - sort packet in queue according to tag
  - serve according to HOL
- WFQ and its many variations form the basis for providing QoS in packet networks

83

Buffer Management

- Packet drop strategy: Which packet to drop when buffers full
- Fairness: protect behaving sources from misbehaving sources
- Aggregation:
  - Per-flow buffers protect flows from misbehaving flows
  - Full aggregation provides no protection
  - Aggregation into classes provided intermediate protection
- Drop priorities:
  - Drop packets from buffer according to priorities
  - Maximizes network utilization & application QoS
  - Examples: layered video, policing at network edge
- Controlling sources at the edge

Early or Overloaded Drop

Random early detection (RED):

- drop pkts if short-term avg of queue exceeds threshold
- pkt drop probability increases linearly with queue length
- mark offending pkts
- improves performance of cooperating TCP sources
- increases loss probability of misbehaving sources

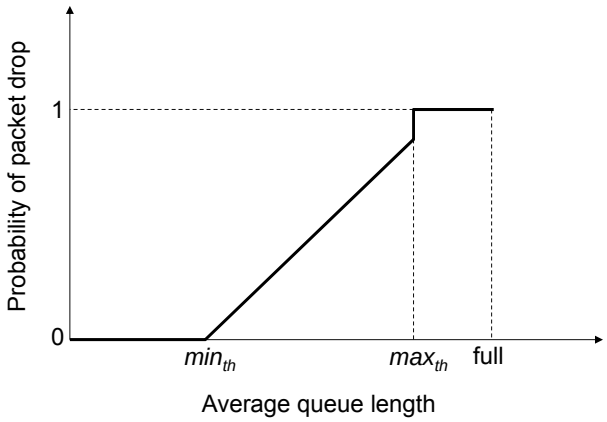
Random Early Detection (RED)

- Packets produced by TCP will reduce input rate in response to network congestion
- Early drop: discard packets before buffers are full
- Random drop causes some sources to reduce rate before others, causing gradual reduction in aggregate input rate

Algorithm:

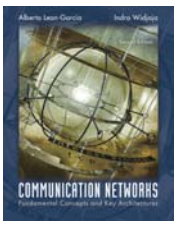
- Maintain running average of queue length
- If Q_{avg} < minthreshold, do nothing
- If Q_{avg} > maxthreshold, drop packet
- If in between, drop packet according to probability
- Flows that send more packets are more likely to have packets dropped

Packet Drop Profile in RED



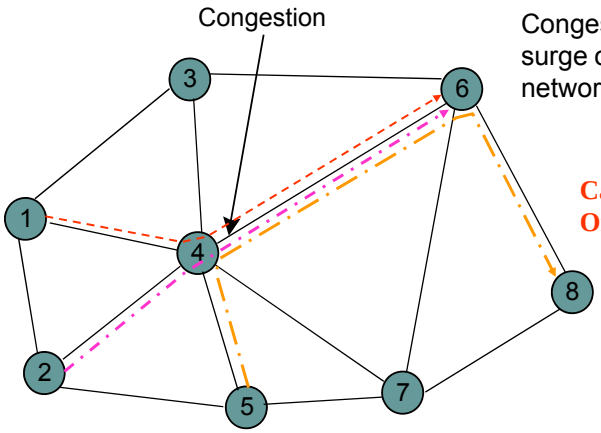
The method of "early" is used to notify some source to reduce its transmission rate before the buffer becomes full.

Chapter 7
Packet-Switching Networks



Traffic Management at the Flow Level

Why Congestion?



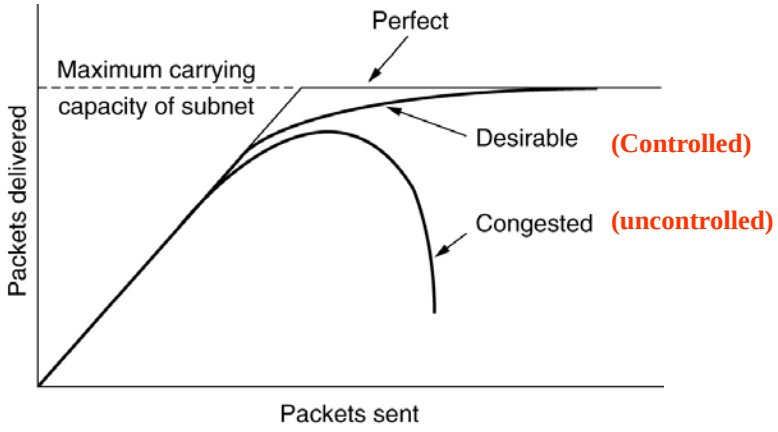
Congestion occurs when a surge of traffic overloads network resources

Can a large buffer help? Or even worse?

Approaches to Congestion Control:

- Preventive Approaches (open –loop): Scheduling & Reservations
- Reactive Approaches (closed-loop): Detect & Throttle/Discard

Ideal Effect of Congestion Control

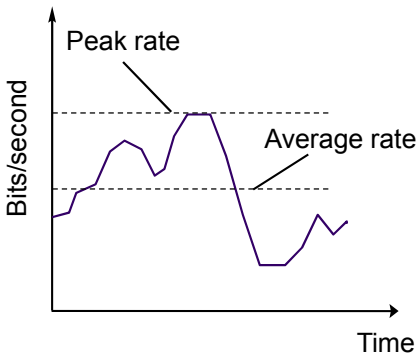


Resources used efficiently up to capacity available

Open-Loop Control

- Network performance is guaranteed to all traffic flows that have been admitted into the network
- Initially for connection-oriented networks
- Key Mechanisms
  - Admission Control
  - Policing
  - Traffic Shaping
  - Traffic Scheduling

Admission Control



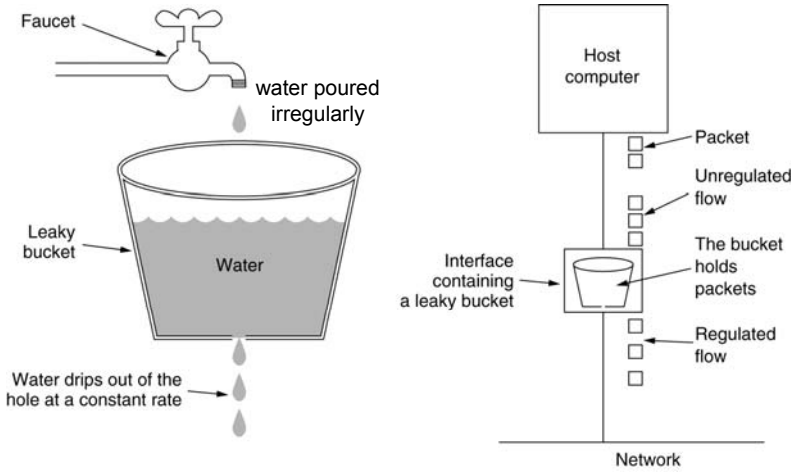
Typical bit rate demanded by a variable bit rate information source

- Flows negotiate contract with network
- Specify requirements:
  - Peak, Avg., Min Bit rate
  - Maximum burst size
  - Delay, Loss requirement
- Network computes resources needed
  - "Effective" bandwidth
- If flow accepted, network allocates resources to ensure QoS delivered as long as source conforms to contract

Policing

- Network monitors traffic flows continuously to ensure they meet their traffic contract
- When a packet violates the contract, network can discard or tag the packet giving it lower priority
- If congestion occurs, tagged packets are discarded first
- *Leaky Bucket Algorithm* is the most commonly used policing mechanism
  - Bucket has specified leak rate for average contracted rate
  - Bucket has specified depth to accommodate variations in arrival rate
  - Arriving packet is *conforming* if it does not result in overflow

The Leaky Bucket



(a) A leaky bucket with water. (b) a leaky bucket with packets.

The Leaky Bucket Example

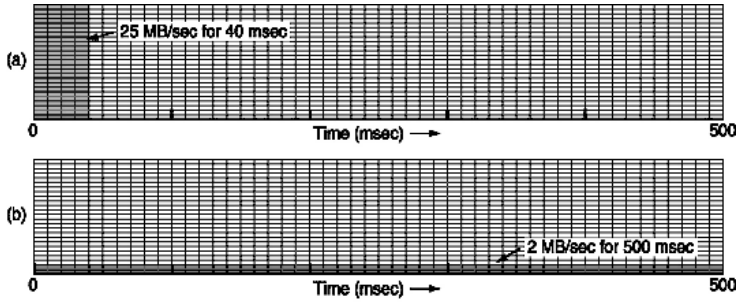
- Data comes to a router in 1 MB bursts, that is, an input runs at 25 MB/s (burst rate) for 40 msec. The router is able to support 2 MB/s output (leaky) rate. The router uses a leaky bucket for traffic shaping.

  (1) How large the bucket should be so there is no data loss (assuming fluid system)?

  (2) Now, if the leaky bucket size is 1MB, how long the *maximum burst interval* can be?

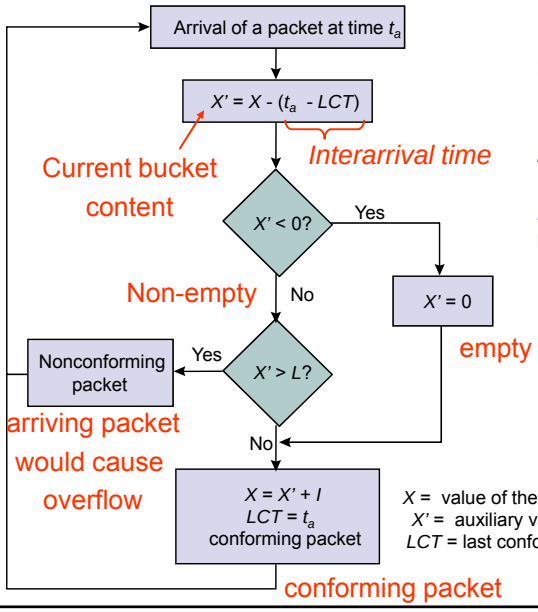
The Leaky Bucket Example

° **Example: data comes to a router in 1 MB bursts, that is, an input runs at 25 MB/s for 40 msec. The router is able to support 2 MB/s outgoing (leaky) rate. The leaky bucket size is 1MB.**



(a) Input to a leaky bucket. (b) Output from a leaky bucket.

Leaky Bucket Algorithm



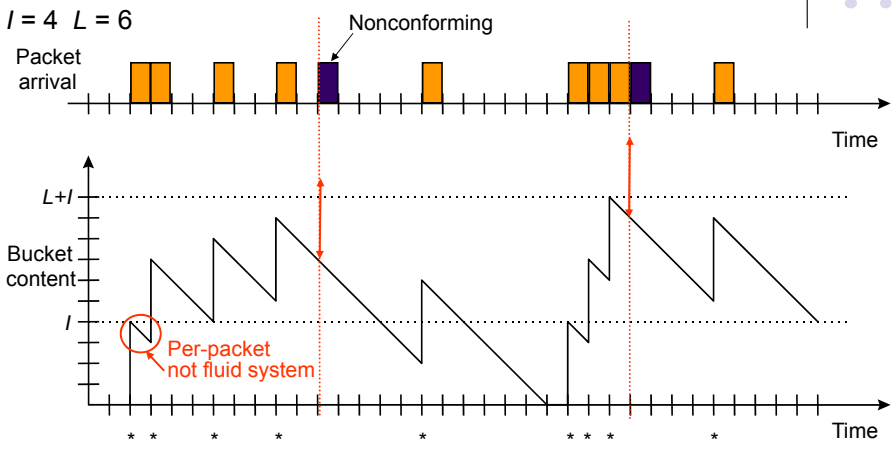
Depletion rate:
1 packet per unit time

$L+I$ = Bucket Depth

I = increment per arrival, nominal interarrival time

X = value of the leaky bucket counter
X' = auxiliary variable
LCT = last conformance time

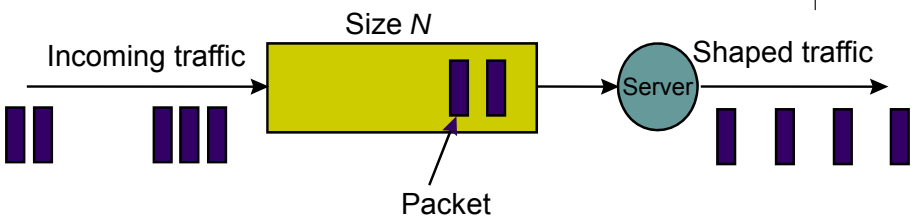
Leaky Bucket Example



Non-conforming packets not allowed into bucket & hence not included in calculations

maximum burst size (MBS = 3 packets)

Leaky Bucket Traffic Shaper

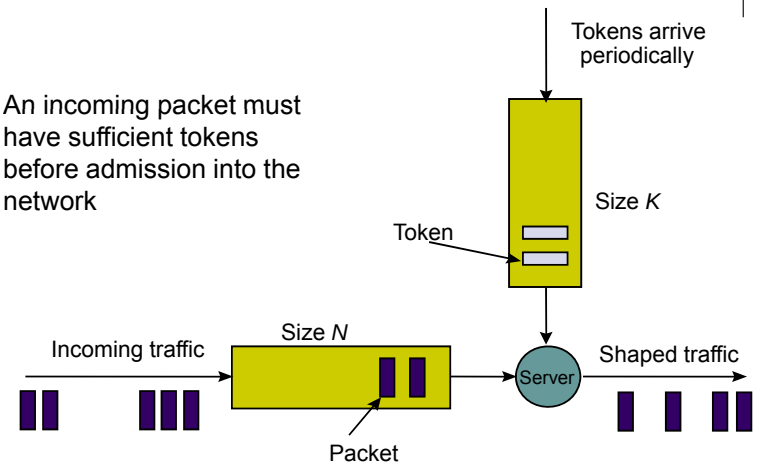


- Buffer incoming packets
- Play out periodically to conform to parameters
- Surges in arrivals are buffered & smoothed out
- Possible packet loss due to buffer overflow

Too restrictive, since conforming traffic does not need to be completely smooth, how to allow some burstiness?

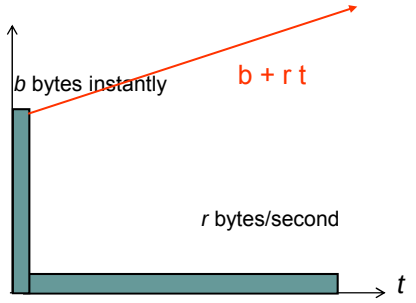
Token Bucket Traffic Shaper

An incoming packet must have sufficient tokens before admission into the network



- Token rate regulates transfer of packets
- If sufficient tokens available, packets enter network without delay
- K determines how much burstiness allowed into the network

Token Bucket Shaping Effect



The token bucket constrains the traffic from a source to be limited to $b + r\,t$ bits in an interval of length t

Q1: what are two main differences of a leaky bucket and a token bucket?

Allow saving for burst spending; packet discarding or not.

Q2: When a token bucket is the same as a leaky bucket?

b = 0; but still different indeed: packet discarding or not

The Token Bucket Example 1

- **A network uses a token bucket for traffic shaping. A new token is put into the bucket every 1 msec. Each token is good for one packet, which contains 100 bytes of data. What is the maximum sustainable (input) data rate?**

The Token Bucket Example 2

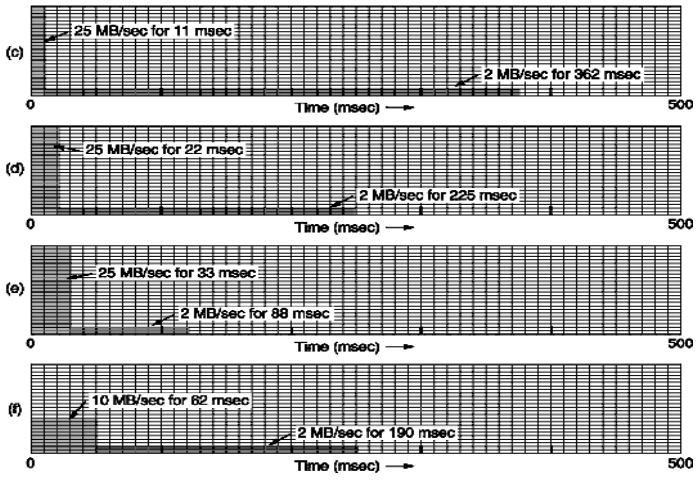
- Given: the token bucket capacity C, the token arrival rate p, and the maximum output rate M, calculate the maximum burst interval S

$$C + pS = MS$$

- Example 2: data comes to a router in 1 MB bursts, that is, an input runs at 25 MB/s (burst rate) for 40 msec. The router uses a token bucket with capacity of 250KB for traffic shaping. Initially, the bucket is full of tokens. And, the tokens are generated and put into the bucket in a rate of 2 MB/s.

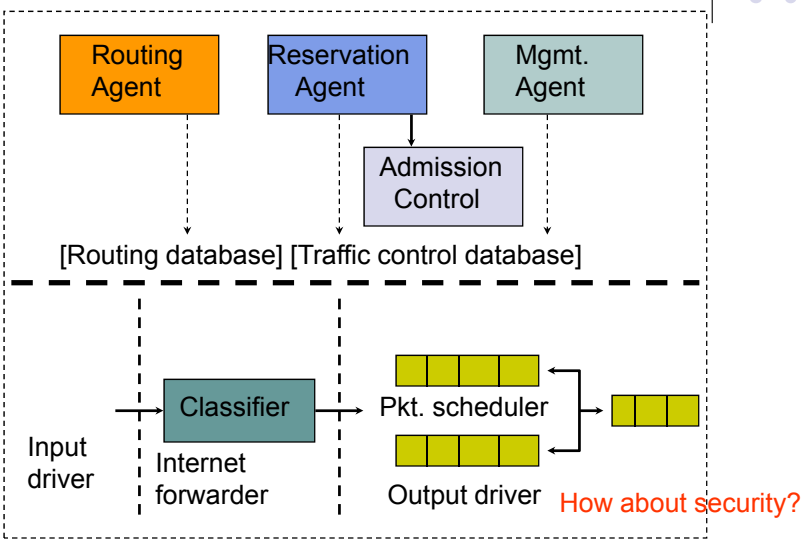
What will be the output from the token bucket?

Fluid Examples



Output from a token bucket with capacities of (c) 250 KB, (d) 500 KB, (e) 750 KB, (f) **Output from a 500KB token bucket feeding a 10-MB/sec leaky bucket of 1MB.**

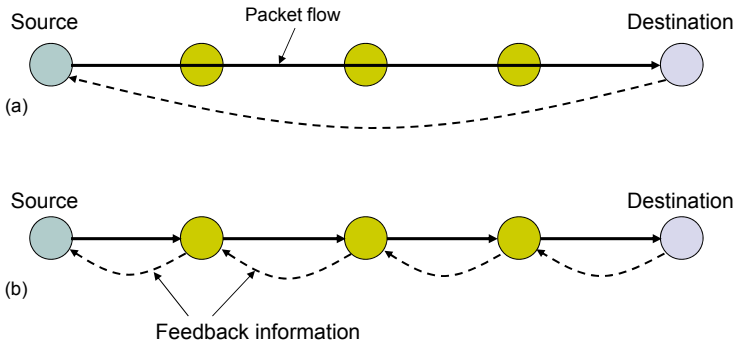
Current View of Router Function



Closed-Loop Flow Control

- Congestion control
  - feedback information to regulate flow from sources into network
  - Based on buffer content, link utilization, etc.
  - Examples: TCP at transport layer; congestion control at ATM level
- End-to-end vs. Hop-by-hop
  - Delay in effecting control
- Implicit vs. Explicit Feedback
  - Source deduces congestion from observed behavior
  - Routers/switches generate messages alerting to congestion

E2E vs. H2H Congestion Control



TCP vs. ATM

Congestion Warning

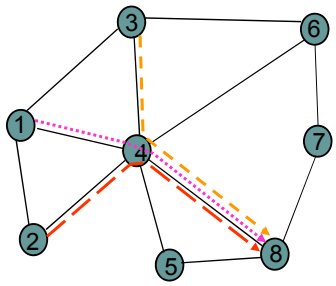
- Threshold-based utilization warning
  - Which factor used for threshold calculation?
  - How to measure the utilization? Instantaneously or smoothed?
  - How to set the threshold?
  - How many threshold levels?
- The Warning Bit in ACKs
- Choke packets to the source

Isn't this approach too slow in reaction?

Traffic Engineering

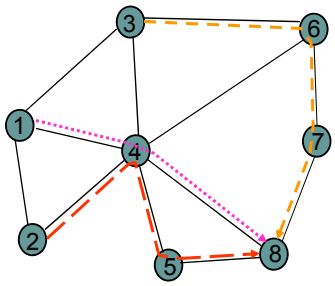
- Management exerted at flow aggregate level
- Distribution of flows in network to achieve efficient utilization of resources (bandwidth)
- Shortest path algorithm to route a given flow not enough
  - Does not take into account requirements of a flow, e.g. bandwidth requirement
  - Does not take account interplay between different flows
- Must take into account aggregate demand from all flows

Effect of Traffic Engineering



(a)

Shortest path routing congests link 4 to 8



(b)

Better flow allocation distributes flows more uniformly