University of Southampton
School of Electronics and Computer Science

Network Layer 2

ELEC3227/ELEC6255

Alex Weddell
asw@ecs.soton.ac.uk

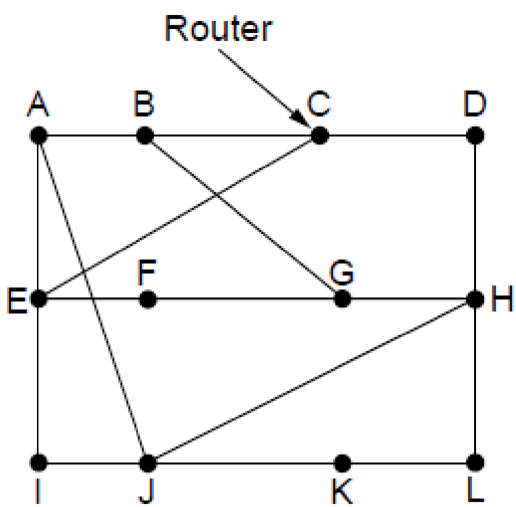
Source: ThinkStock

UNIVERSITY OF Southampton
School of Electronics and Computer Science

Overview

- Shortest path algorithm
- Congestion/admission control/throttling
- Application requirements
- Traffic shaping

Distance Vector Routing



Network

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		

$\underbrace{\text{JA, JI, JH, JK}}$ Vectors received at J from Neighbors A, I, H and K

New vector for J

The "Count-to-Infinity" Problem

- Failures can cause DV to "count to infinity" while seeking a path to an unreachable node

A	B	C	D	E	
	•	•	•	•	Initially
	1	•	•	•	After 1 exchange
	1	2	•	•	After 2 exchanges
	1	2	3	•	After 3 exchanges
	1	2	3	4	After 4 exchanges

Good news of a path to *A* spreads quickly

A	B	C	D	E	
	1	2	3	4	Initially
	3	2	3	4	After 1 exchange
	3	4	3	4	After 2 exchanges
	5	4	5	4	After 3 exchanges
	5	6	5	6	After 4 exchanges
	7	6	7	6	After 5 exchanges
	7	8	7	8	After 6 exchanges
		⋮			
	•	•	•	•	

Bad news of no path to *A* is learned slowly

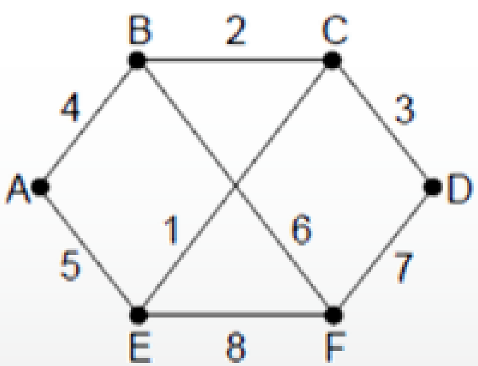
Distance Vector Routing

- Safeguards against “count to infinity”…

1. **Route poisoning:** advertise a cost of infinity for links that are *known* to have failed. *This means news of a failed link spreads more quickly.*
2. **Split horizon:** do not advertise route involving a neighbour back to the same neighbour. *This means that routes involving bad links will die out more quickly.*

Link State Routing

- LSP (Link State Packet) for a node lists neighbors and weights of links to reach them



Network

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

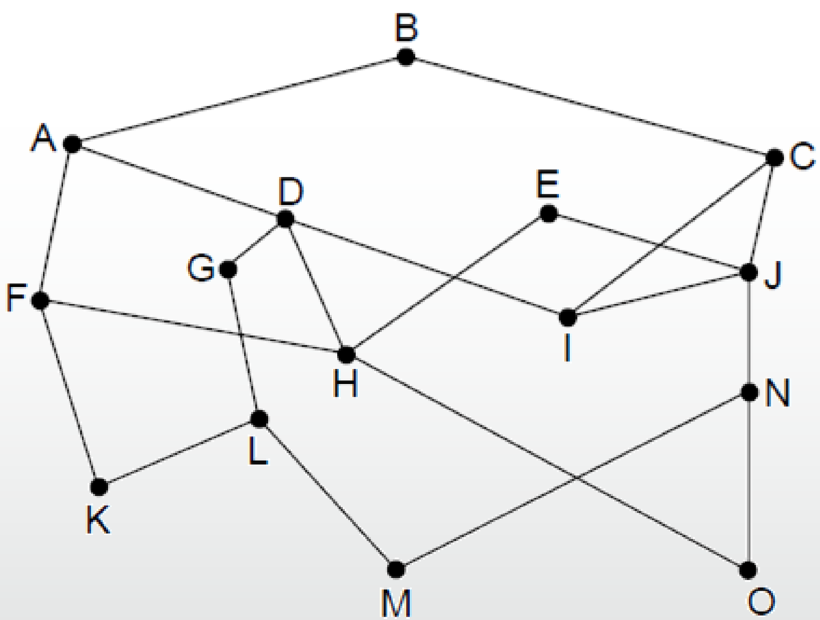
LSP for each node

Link State Routing

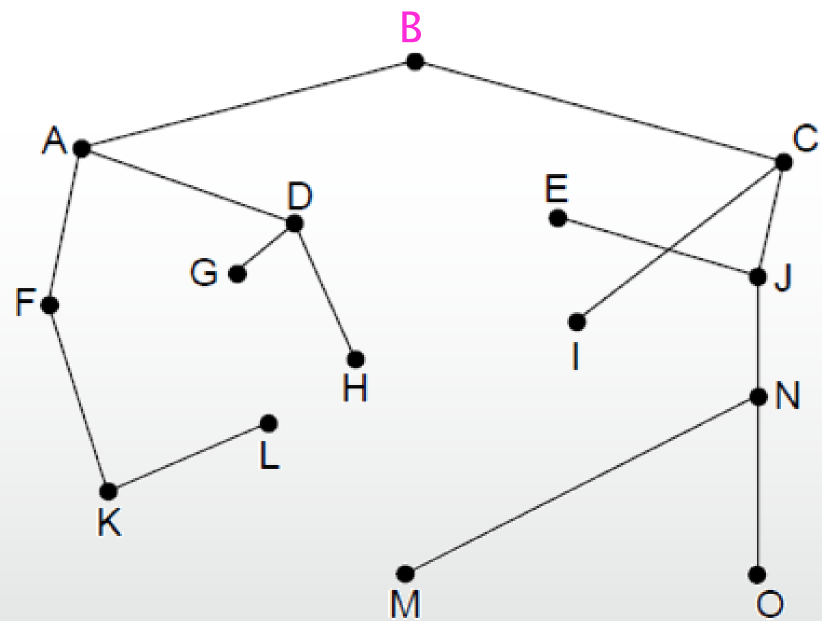
- Link State Packet propagation
  - Reliable flooding – uses acknowledgments and retransmissions
  - Ensures all link state packets transferred to neighbours.
- Issues
  - Not working from same ‘map’
    - Routing loops can form if two nodes each think their neighbour is on the best route
  - Unreliable links
    - ‘Hello’ protocol – periodically tests connectivity with neighbours.
  - Restarting links
    - Recognise ‘age’ field of LSP: re-initialise.

The Optimality Principle

- Each portion of a best path is also a best path; the union of them to a router is a tree called the **sink tree**
  - "Best" means fewest hops in the example



Network



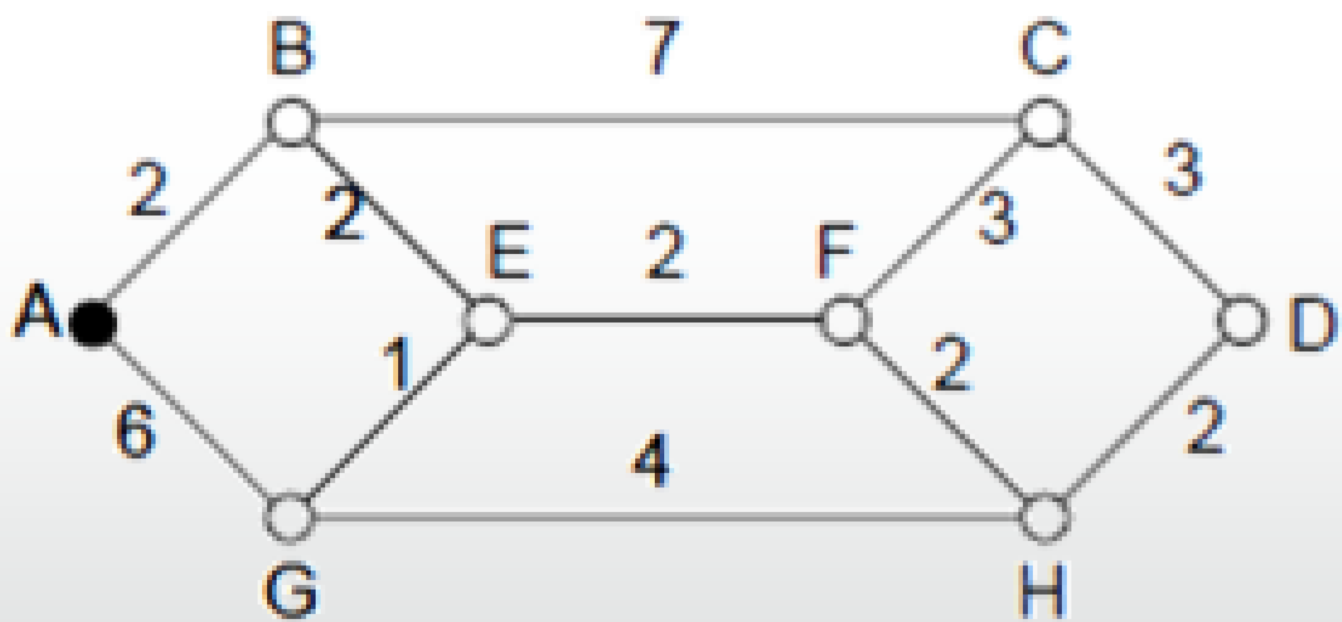
Sink tree of best paths to router B

Shortest Path Algorithm

- **Dijkstra's algorithm** computes a sink tree on the graph:
  - Each link is assigned a non-negative weight/distance
  - Shortest path is the one with lowest total **weight**
  - Using weights of 1 gives paths with fewest **hops**

- Algorithm:
  - Start with sink, set **distance** at other nodes to **infinity**
  - **Relax** distance to other nodes
  - Pick the **lowest distance** node, add it to sink tree
  - **Repeat** until all nodes are in the sink tree

Shortest Path Algorithm



UNIVERSITY OF Southampton
School of Electronics
and Computer Science

Shortest Path Algorithm

. . .

```
for (p = &state[0]; p < &state[n]; p++) {
    p->predecessor = –1;
    p->length = INFINITY;
    p->label = tentative;
}
state[t].length = 0;  state[t].label = permanent;
k = t;
do {
    for (i = 0; i < n; i++)
        if (dist[k][i] != 0 && state[i].label == tentative) {
            if (state[k].length + dist[k][i] < state[i].length) {
                state[i].predecessor = k;
                state[i].length = state[k].length + dist[k][i];
            }
        }
```

. . .

Start with the sink, all other nodes are unreachable

Relaxation step. Lower distance to nodes linked to newest member of the sink tree

Shortest Path Algorithm

. . .

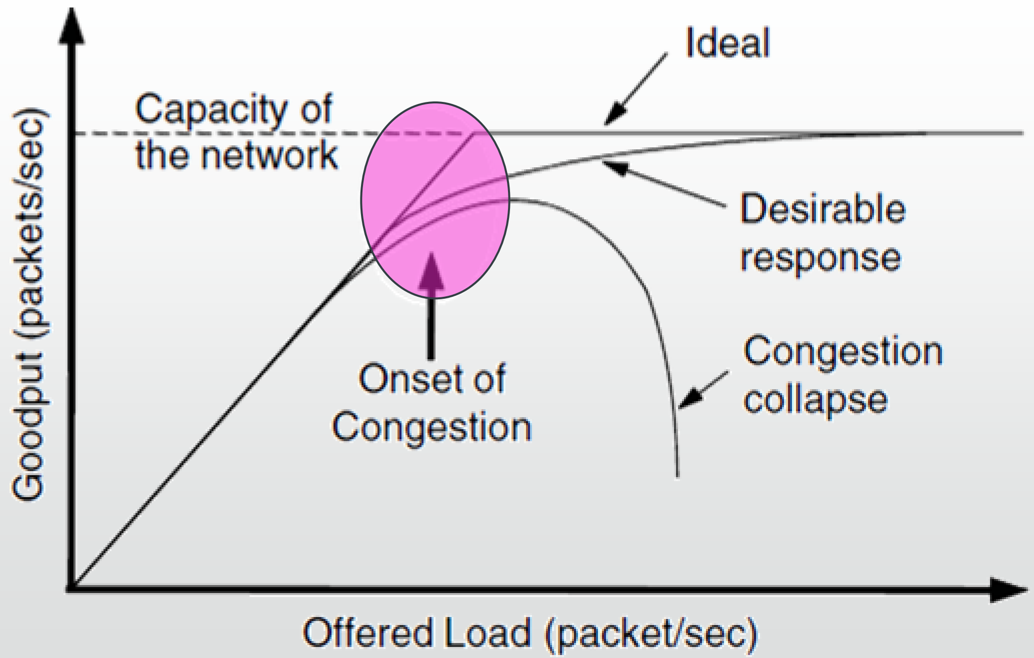
```
        k = 0; min = INFINITY;
        for (i = 0; i < n; i++)
                if (state[i].label == tentative && state[i].length < min) {
                        min = state[i].length;
                        k = i;
                }
        state[k].label = permanent;
} while (k != s);
```

Find the lowest distance, add it to the sink tree, and repeat until done

UNIVERSITY OF Southampton
School of Electronics
and Computer Science

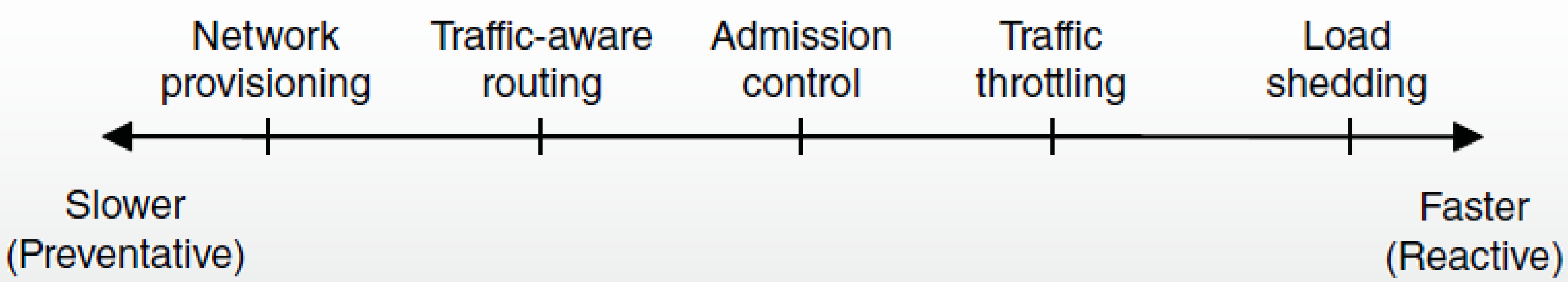
Congestion Control

- Handling congestion is the responsibility of the Network and Transport layers working together
- Congestion results when too much traffic is offered; performance degrades due to loss/retransmissions
  - Goodput (=useful packets) trails offered load



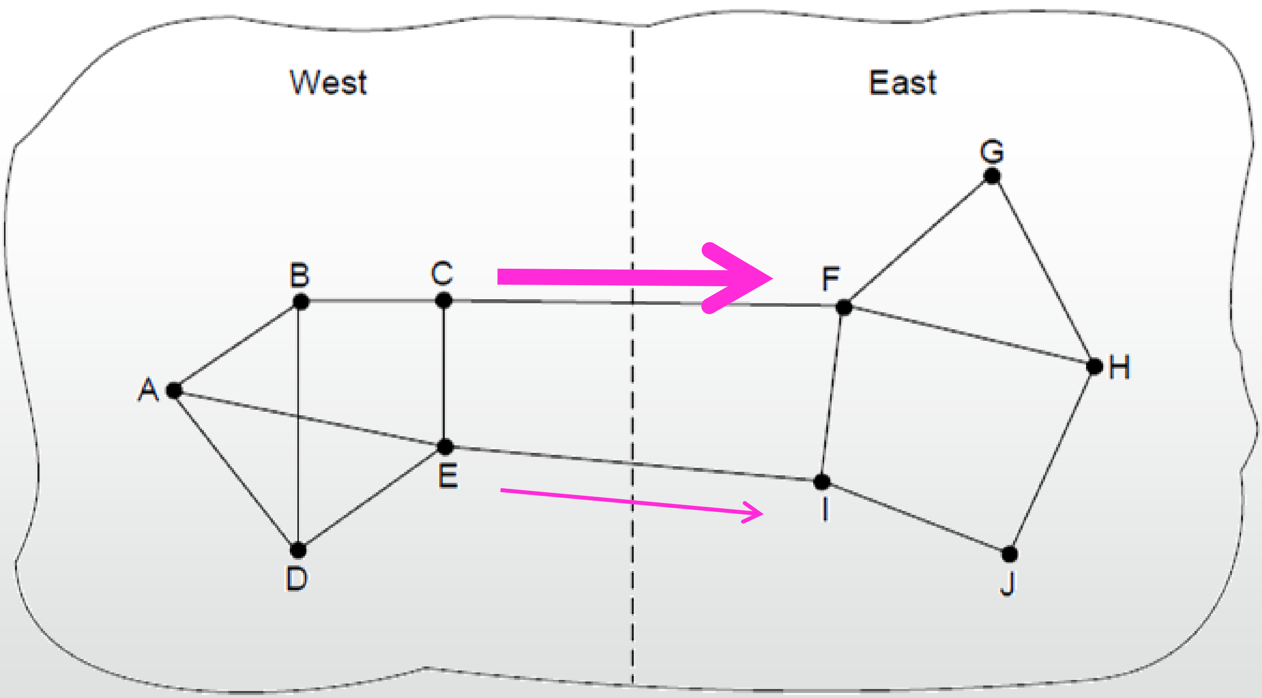
Congestion Control Approaches

- Network must do its best with the offered load
  - Different approaches at different timescales
  - Nodes should also reduce offered load (Transport)



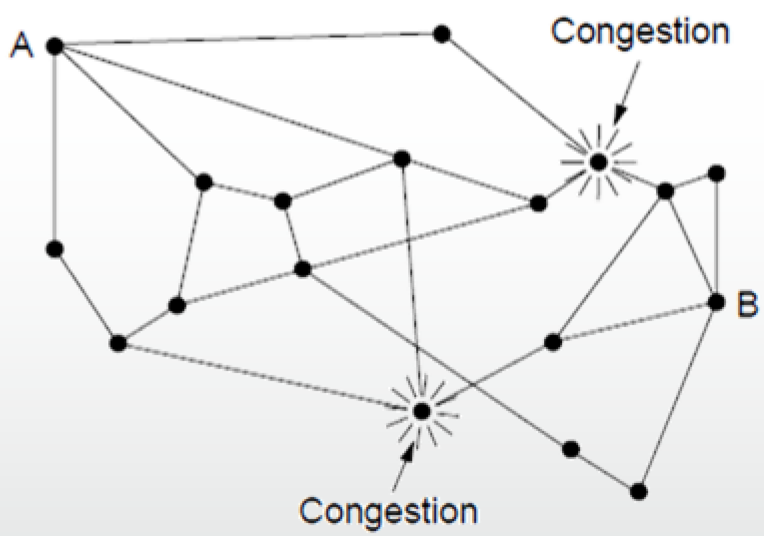
Traffic-Aware Routing

- Choose routes depending on traffic, not just topology
  - E.g., use *EI* for West-to-East traffic if *CF* is loaded
  - But take care to avoid oscillations

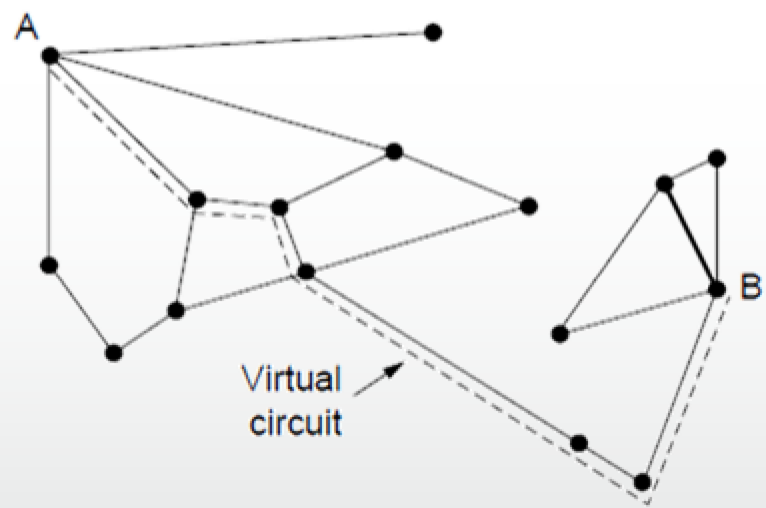


Admission Control

- Admission control allows a new traffic load only if the network has sufficient capacity, e.g., with virtual circuits
  - Can combine with looking for an uncongested route



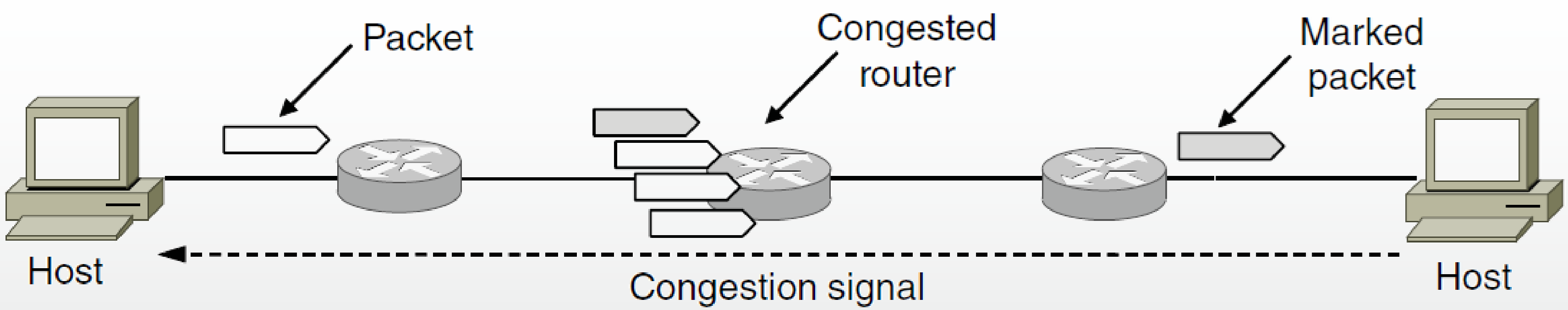
Network with some congested nodes



Uncongested portion and route *AB* around congestion

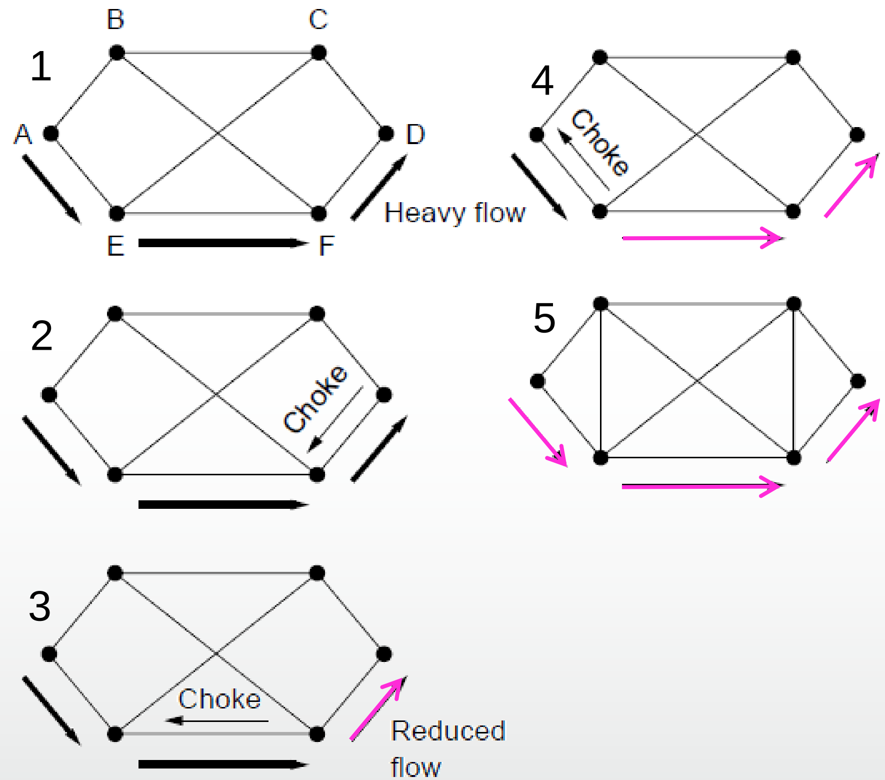
Traffic Throttling

- Congested routers signal hosts to slow down traffic
  - ECN (Explicit Congestion Notification) marks packets and receiver returns signal to sender



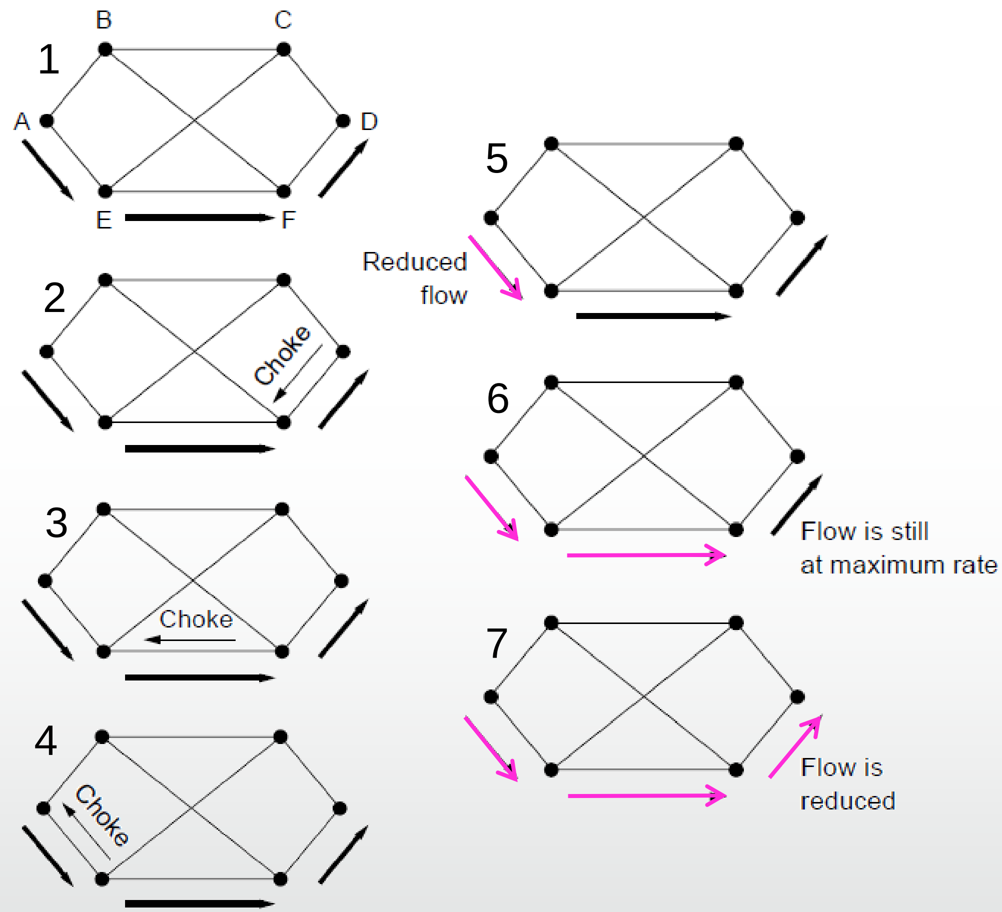
Traffic Throttling

- 'Choke' notification can be end-to-end or link-by-link
- **Link-by-link** (right) produces rapid relief, but requires larger buffering capability by intermediate routers



Traffic Throttling

- **End-to-end** (right) takes longer to have an effect, but can better target the cause of congestion



UNIVERSITY OF Southampton
School of Electronics and Computer Science

Load Shedding

- When all else fails, network will drop packets (shed load)
- “Wine” policy: older is better
  - Better for file transfer
  - Receiver needs to buffer fewer packets for incomplete transmission
- “Milk” policy: newer is better
  - Better for media streaming
  - After a certain amount of time, media packet is useless, so should discard oldest packets
- Intelligent load-shedding
  - Routing information packets may have a higher priority – less likely to be ‘shed’
  - Similar for some media packets: MPEG transmits a complete image frame periodically, and then differences from that frame… So needs to prioritise the periodic complete image transmission.

Application Requirements

- Different applications care about different properties
  - We want all applications to get what they need

Application	Bandwidth	Delay	Jitter	Loss
Email	Low	Low	Low	Medium
File sharing	High	Low	Low	Medium
Web access	Medium	Medium	Low	Medium
Remote login	Low	Medium	Medium	Medium
Audio on demand	Low	Low	High	Low
Video on demand	High	Low	High	Low
Telephony	Low	High	High	Low
Videoconferencing	High	High	High	Low

"High" means a demanding requirement, e.g., low delay

Application Requirements

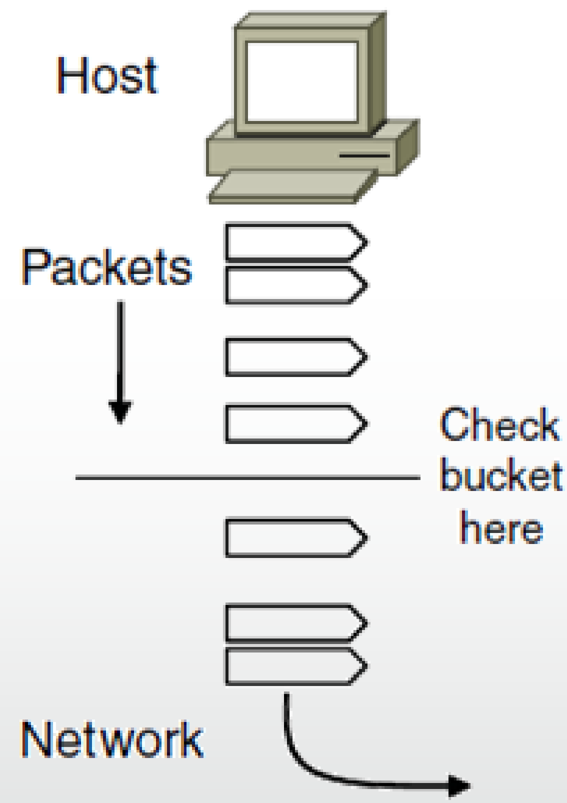
- Network provides service with different kinds of QoS (Quality of Service) to meet application requirements
- QoS is part of the routing policy decisions of an Autonomous System (AS). Routers within an AS are configured to reflect the policies of that specific AS. QoS provides an optional mechanism for routers to tailor their behaviour based upon the differing needs of specific applications.

Network Service	**Application**
Constant bit rate	Telephony
Real-time variable bit rate	Videoconferencing
Non-real-time variable bit rate	Streaming a movie
Available bit rate	File transfer

Example of QoS categories from ATM networks

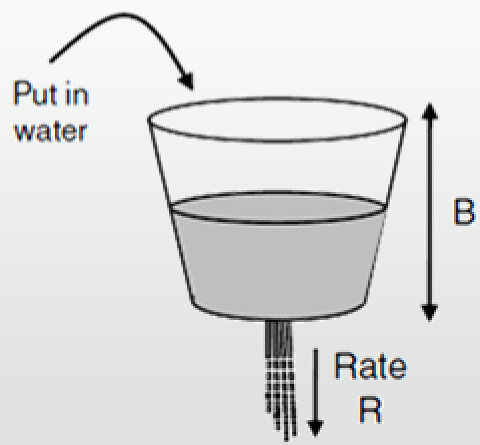
Traffic Shaping

- Traffic shaping regulates the average rate and burstiness of data entering the network
  - Lets us make Service Level Agreement (SLA) guarantees
  - For example, packets in excess of the agreed upon pattern might be dropped by the network or marked as having a lower priority
  - Traffic Policing = monitoring Traffic Flow
- Algorithms are used to limit long-term rate of flow, but allow short-term bursts up to a certain length.

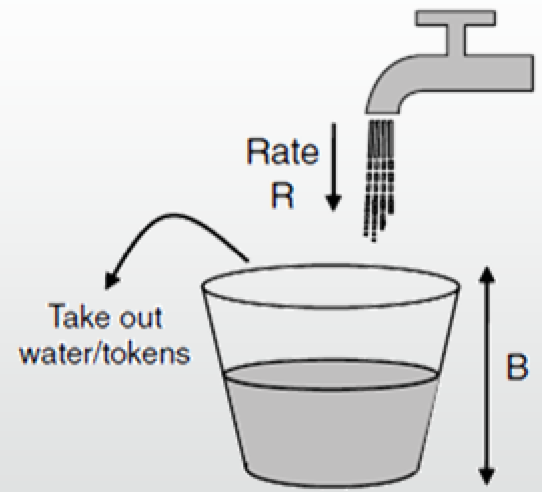


Traffic Shaping

- Token/Leaky bucket limits both the average rate (R) and short-term burst (B) of traffic
  - Leaky bucket algorithm – No matter what rate the packets enter the bucket, the outflow is at a constant rate (R) or less
  - Token bucket algorithm – to send a packet must be able to take tokens out of bucket. No more than a fixed # of tokens (B) can accumulate in bucket.
- For token, bucket size is B, water enters at rate R and is removed to send; opposite for leaky.



Leaky bucket
(to send, must not be full)



Token bucket
(need some water to send)

Traffic Shaping

- Token buckets

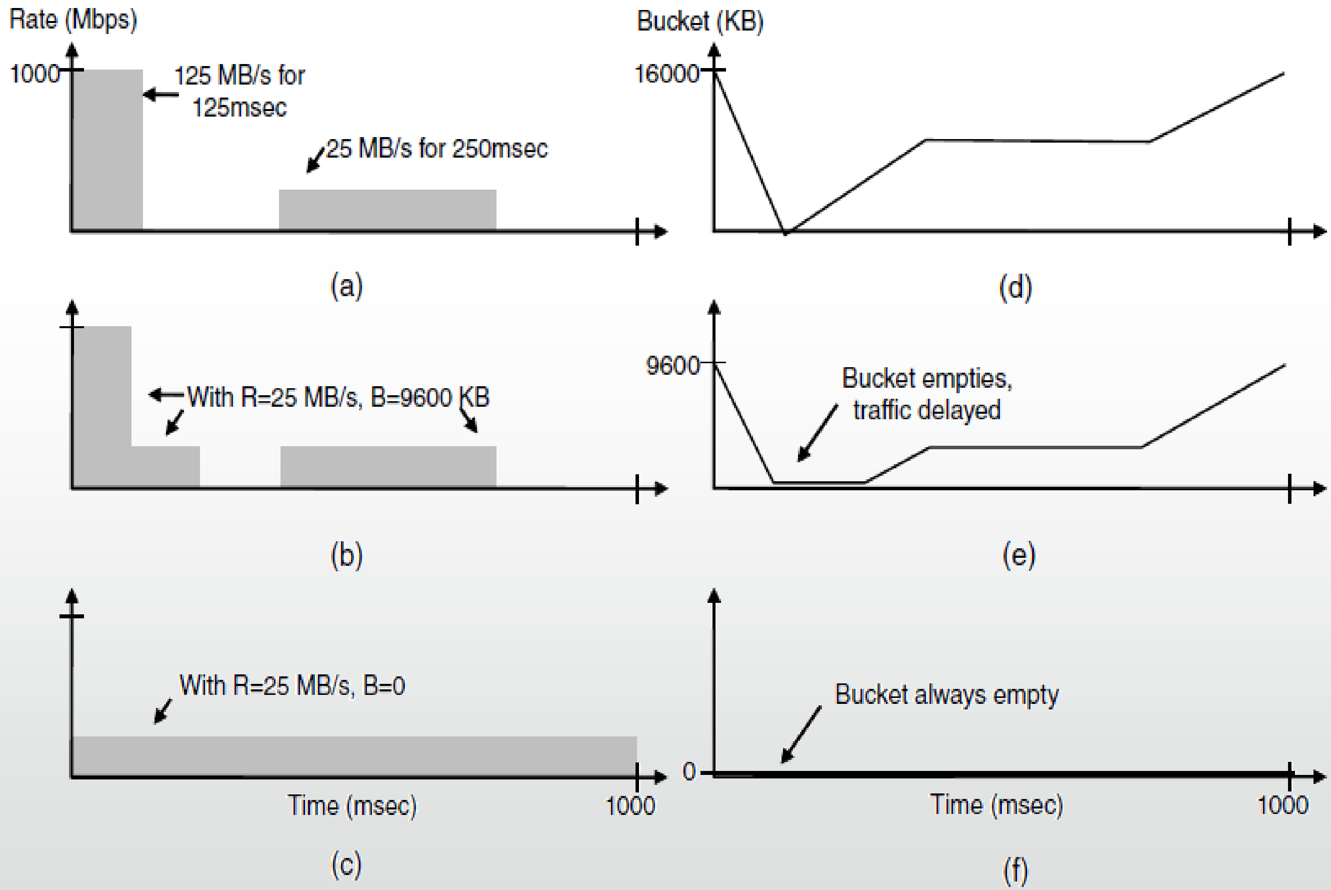
Host traffic
R=200 Mbps
B=16000 KB

↓

Shaped by
R=200 Mbps
B=9600 KB

↓

Shaped by
R=200 Mbps
B=0 KB



Smaller bucket size delays traffic and reduces burstiness

UNIVERSITY OF Southampton
School of Electronics and Computer Science

Overview

- Shortest path algorithm
- Congestion/admission control/throttling
- Application requirements
- Traffic shaping