

# COMP2211 - Networks and Systems

## Networks: Network Layer

Anne Reinarz

[anne.k.reinarz@durham.ac.uk](mailto:anne.k.reinarz@durham.ac.uk)

Department of Computer Science

# Outline



- Routing algorithm: routing packets from source to destination.
- Optimality principle: finding the optimal path from source to destination.
- Various routing algorithms
  - Dijkstra's algorithm
  - Dynamic routing algorithms:
    - Distance vector algorithm
    - Link state routing
  - Hierarchical routing

# Routing Algorithms

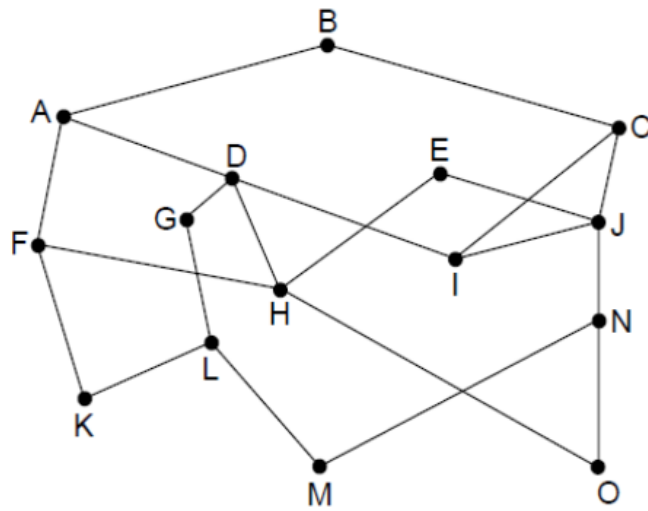
- Routing is the process of discovering network paths
  - Decide what to optimize (e.g., fairness vs efficiency)
  - Model the network as a graph of nodes and links
  - Update routes for changes in topology (e.g., failures)

# Optimality Principle

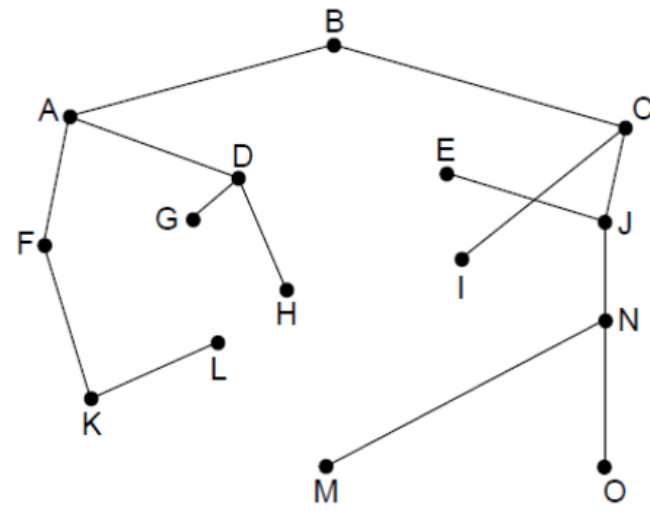
If router J is on the optimal path from I to K, then the optimal path from J to K is on the same route

- Identify the optimal path from source to destination.
- Sink tree: optimal routes from all sources to a given destination.
- Distance metrics:
  - the number of hops, or
  - time delay.

# Optimality Principle



Example network



Sink tree for B

# Shortest Path Algorithm

Dijkstra's algorithm computes a sink tree on the graph:

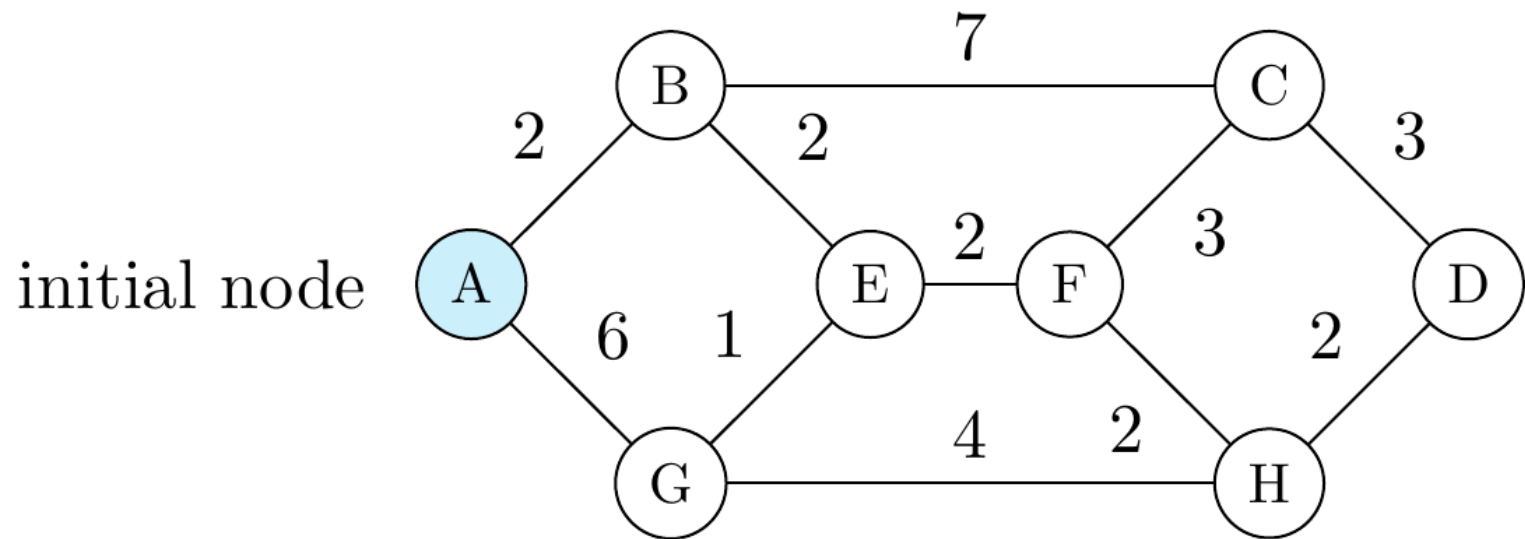
- Each node is labelled with its distance from the source node to the best known path
- Initially no paths are known
- Each link is assigned a non-negative weight/distance
- Shortest path is the one with lowest total weight

# Shortest Path Algorithm

## Algorithm:

- Start with sink, set distance at other nodes to infinity
- Labels tentative (○) or permanent ( • ), initially all tentative
- Pick lowest distance non-permanent node, make perm.
- Repeat from this node, until all nodes are permanent

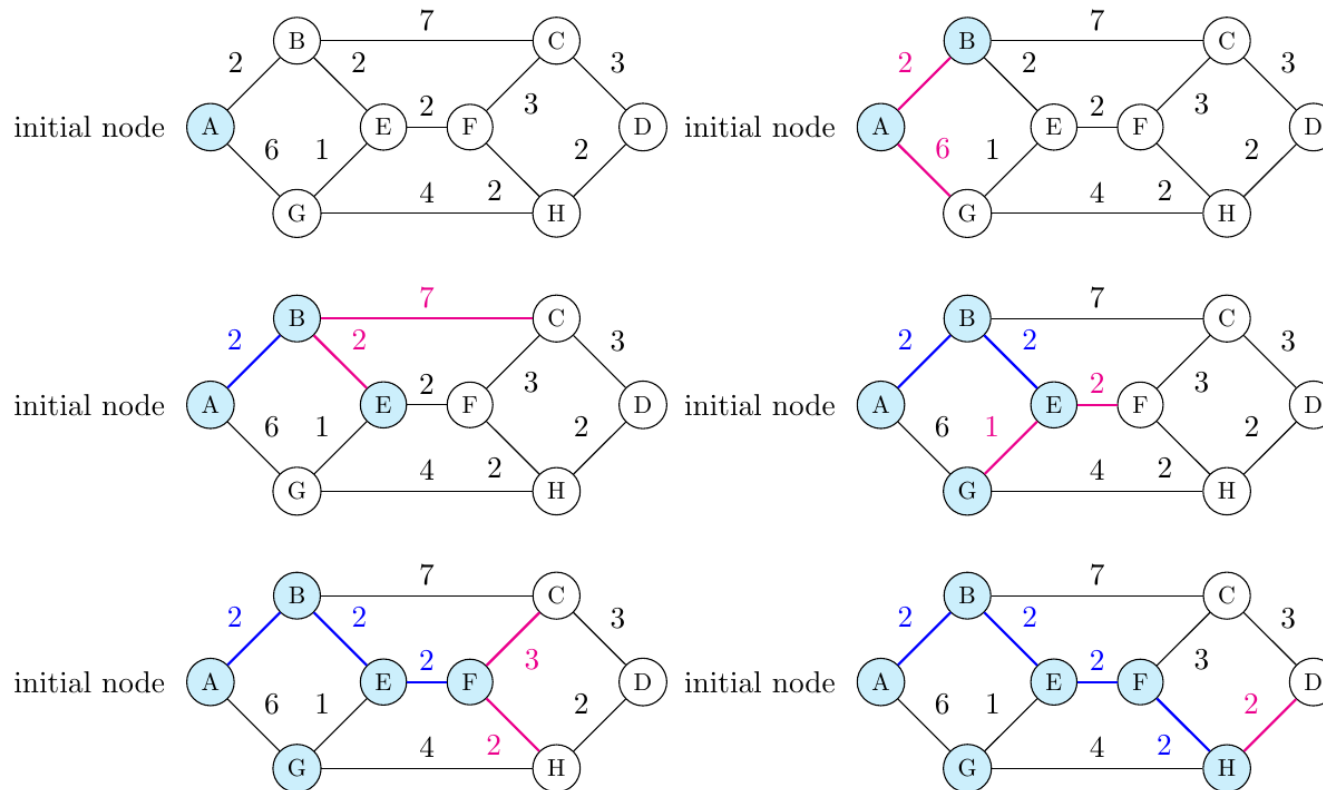
# Shortest Path Algorithm



- weights represent cost/distance



# Shortest Path Algorithm



- A network and first five steps in finding the shortest paths from A to D. Pink arrows show the sink tree so far

# Distance Vector Routing

Two types of dynamic routing algorithm:

1. Distance vector, and
2. link state routing algorithms

# Distance Vector Routing

## Distance vector:

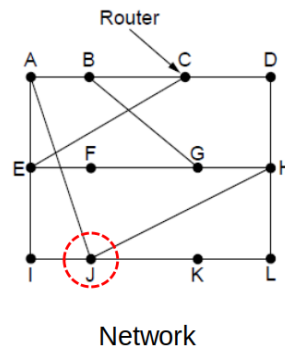
- Each node maintains a table (vector of best distance to dest.)
- Tables are updated by exchanging information between nodes
- Tables have 2 entries: outgoing line and estimated distance (#hops or propagation delay)

# Distance Vector Routing

## Algorithm:

- Each node knows distance of links to its immediate neighbours
- Each node advertises a vector of the lowest known distances to all nodes
- Each node uses received vectors to update its own
- Repeat periodically

# Distance Vector Routing



Compute the route from J to G.

J → A in 8 msec

A → G 18 msec

J → 8 + 18 = 26 msec.

					New estimated delay from J	
					↓ Line	
To	A	I	H	K		
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

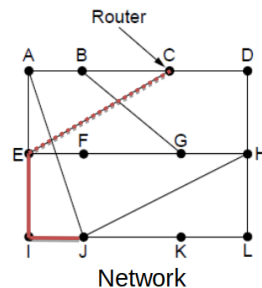
  

JA delay is 8	JI delay is 10	JH delay is 12	JK delay is 6
---------------	----------------	----------------	---------------

New vector for J

Vectors received at J from Neighbours A, I, H and K

# Distance Vector Routing



1. Compute the route from J to C.

J to A = 8 msec , A to C = 25 msec

J to C = 33 msec

J to I = 10 msec , I to C = 18 msec

J to C = 28 msec

J to H = 12 msec , H to C = 19 msec

J to C = 31 msec

J to K = 6 msec , K to C = 36 msec

J to C = 42 msec

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 vector for J

Vectors received at J from  
Neighbours A, I, H and K

# 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	
•	✗	•	•	•	Initially
	1	2	3	4	After 1 exchange
	3	2	3	4	After 2 exchanges
	3	4	3	4	After 3 exchanges
	5	4	5	4	After 4 exchanges
	5	6	5	6	After 5 exchanges
	7	6	7	6	After 6 exchanges
	7	8	7	8	After 7 exchanges
	•	•	•	•	...

Bad news of no path to A  
is learned slowly

# Distance Vector Routing

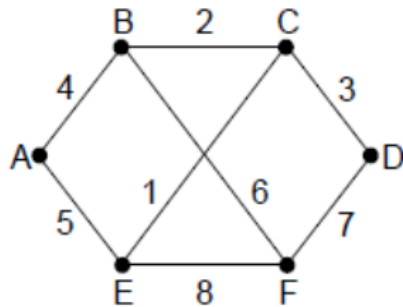
## Link State Routing

1. Learn the network address of the neighbouring routers by sending HELLO packet, record name
2. Set the distance to each neighbour
3. Construct a packet telling all other routers what it has just learned
4. Send the packet to and receive packets from all other routers
5. Compute the shortest path by using Dijkstra's algorithm



# Link State Routing: LSPs

- LSP (Link State Packet) for a node lists neighbours 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

Challenge: getting these packets to *all* routers

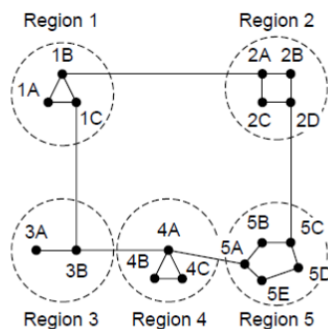
Solution: "flooding"

# Flooding

- A simple method to send a packet to all network nodes
- Each node floods a new packet received on an incoming link by sending it out on all of the other links
- Nodes need to keep track of flooded packets to stop the flood
  - E.g. by using sequence numbers
- Does not rely on routing tables being built first

# Hierarchical Routing

- Hierarchical routing reduces the work of route computation but may result in slightly longer paths than flat routing



a

What router should we use  
from 2A to 4B (i.e. min hops)?

Full table for 1A

Dest.	Line	Hops
1A	—	—
1B	1B	1
1C	1C	1
2A	1B	2
2B	1B	3
2C	1B	3
2D	1B	4
3A	1C	3
3B	1C	2
4A	1C	3
4B	1C	4
4C	1C	4
5A	1C	4
5B	1C	5
5C	1B	5
5D	1C	6
5E	1C	5

b

Hierarchical table for 1A

Dest.	Line	Hops
1A	—	—
1B	1B	1
1C	1C	1
2	1B	2
3	1C	2
4	1C	3
5	1C	4

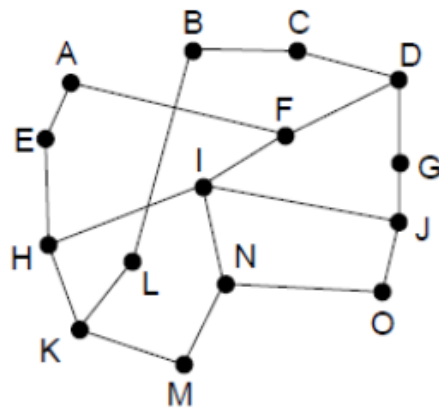
Best choice to  
reach nodes in 5  
except for 5C

c

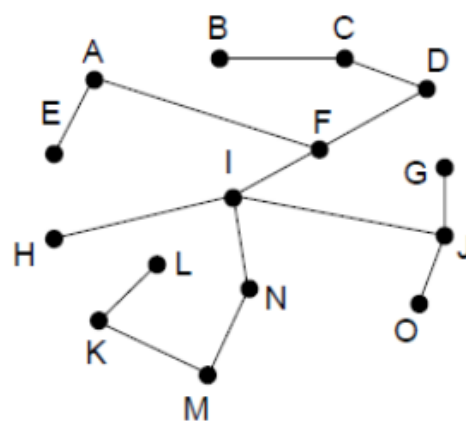
# Broadcast Routing

- Broadcast sends a packet to all nodes
- RPF (Reverse Path Forwarding): Arrived packets are checked to see if they arrived from a preferred link, which is the link that is normally used for sending packets towards the source of the broadcast.
- 1st hop: I sends packets to F, H, J, and N. Packets arrive on the same link that is used to send to I
- 2nd hop: 8 packets are generated, two by each router. 5 of them arrive on the preferred link
- ...

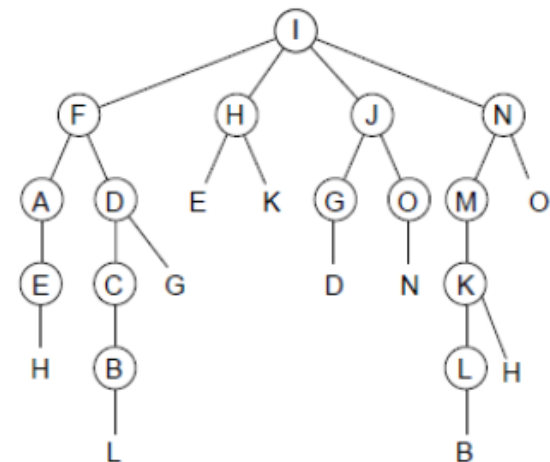
# Broadcast Routing



Network



Sink tree for I is efficient  
broadcast



RPF from I is larger than  
sink tree

# Summary

- Packet routing algorithms
- Different routing algorithms
  - Dijkstra's algorithm
  - Shortest path algorithm
  - Distance vector routing
  - Link state routing
  - Hierarchical routing
- Flooding
- Broadcasting

