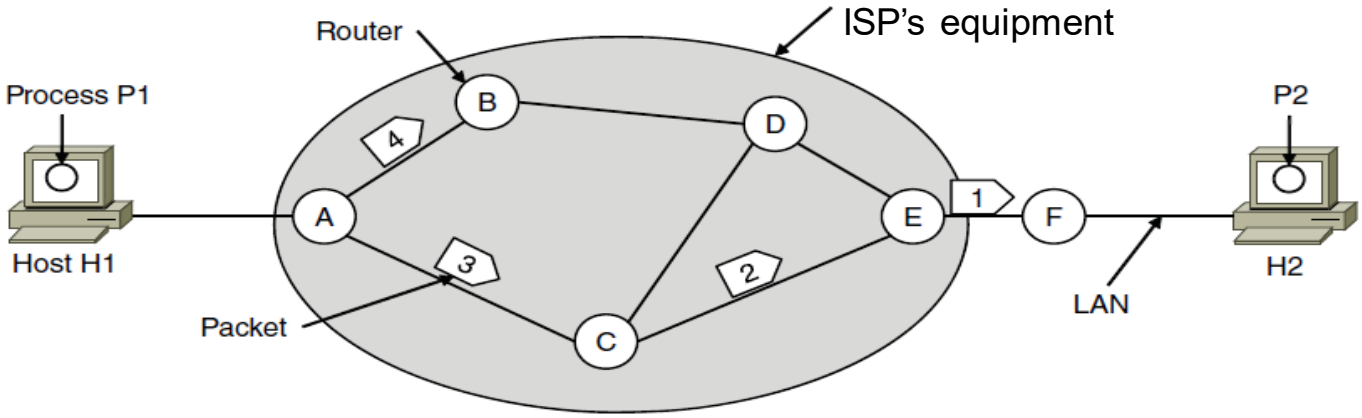
Week 6 – Network Layer Contd

Internet Technologies
COMP90007

Another Key Task: Routing



A's table (initially)

Dest.	Line
A	☒
B	B
C	C
D	B
E	C
F	C

A's table (later)

A	☒
B	B
C	C
D	B
E	B
F	B

C's Table

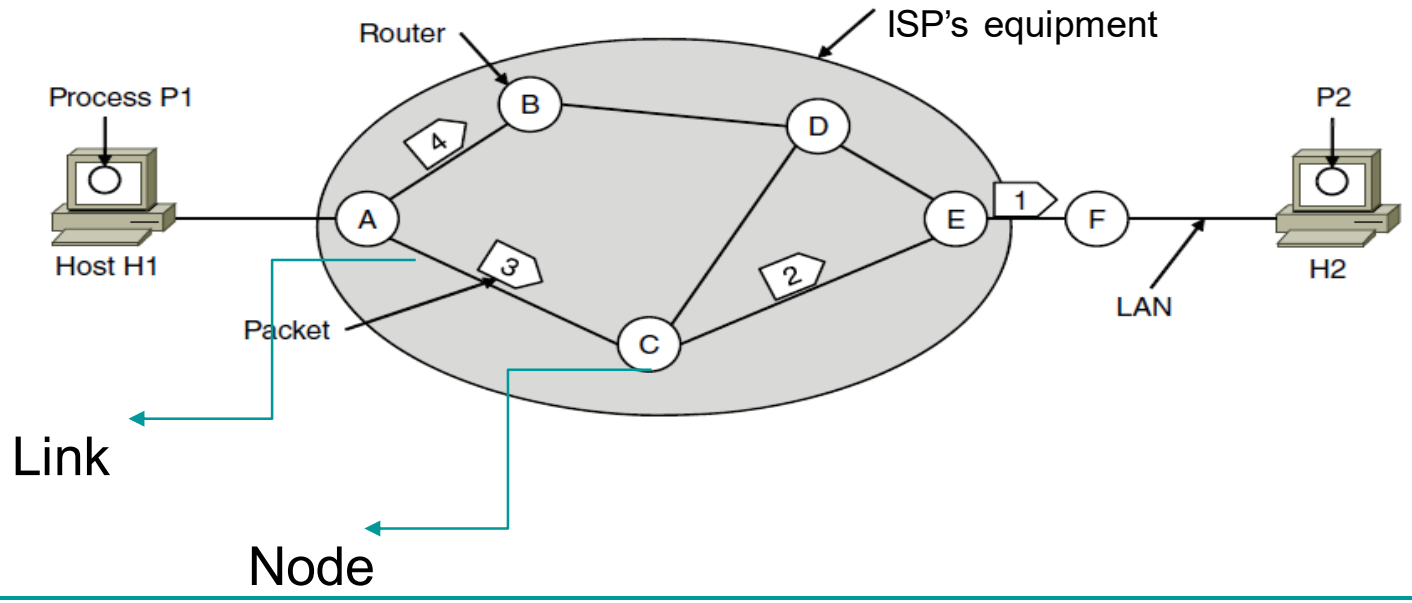
A	A
B	A
C	☒
D	E
E	E
F	E

E's Table

A	C
B	D
C	C
D	D
E	☒
F	F

Routing Algorithms Basics

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



Routing Algorithms and Types

- The **routing algorithm is responsible for deciding on which output line an incoming packet should be transmitted**

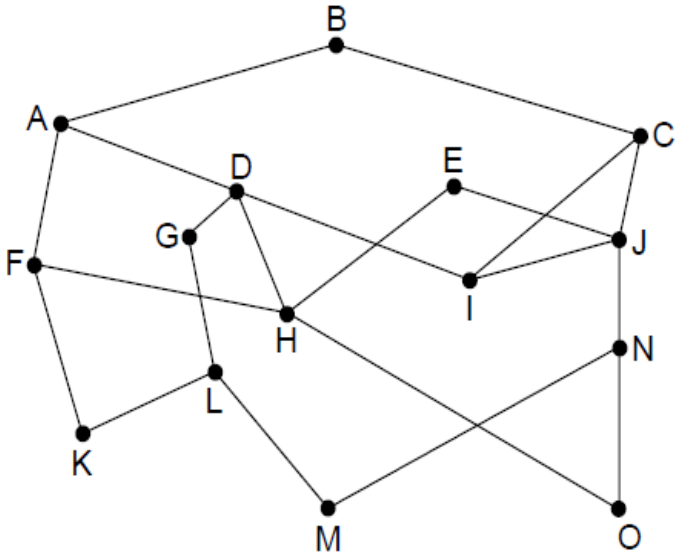
- Non-Adaptive Algorithms
  - Static decision making process (e.g., static routing)
- Adaptive Algorithms
  - Dynamic decision making process (e.g., dynamic routing)
  - Changes in network topology, traffic, etc.

A key principle: Optimality Principle

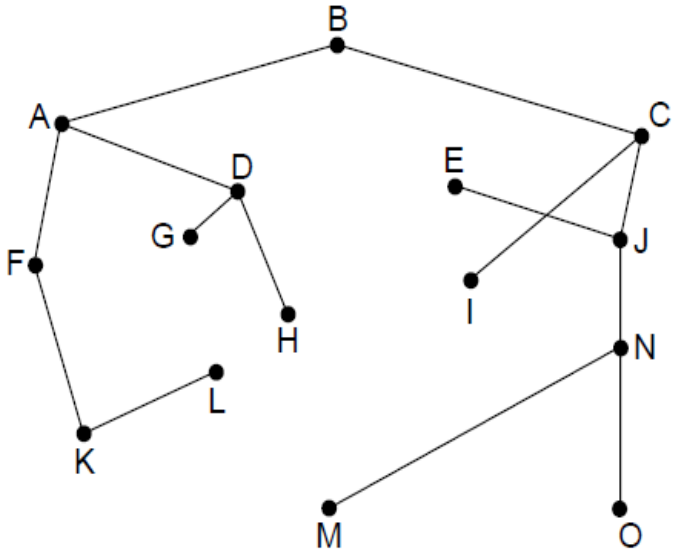
- “If router B is on the optimal path from router A to router C, then the optimal path for B to C also falls along the same route”.

A key graph type: Sink Tree

- The set of optimal routes from all sources to a given destination form a tree rooted at the destination -“sink tree”
- A key *goal of routing algorithms is to discover and utilise the sink trees*



Network



Sink tree of best paths to router B

Shortest Path Routing

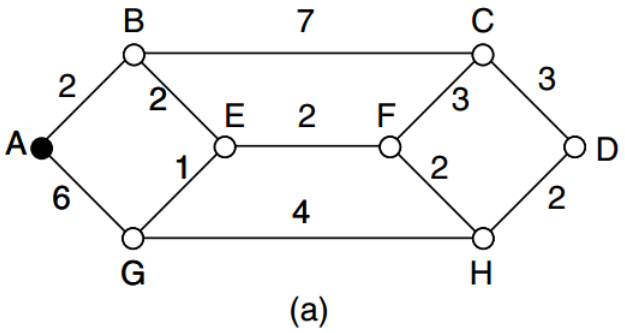
- We will see a non-adaptive algorithm here
- Shortest path can be determined by building a graph with each node representing a router, and each arc representing a communication link
- To choose a path between two routers, the algorithm needs to find the shortest path between them on the graph

Dijkstra's Algorithm

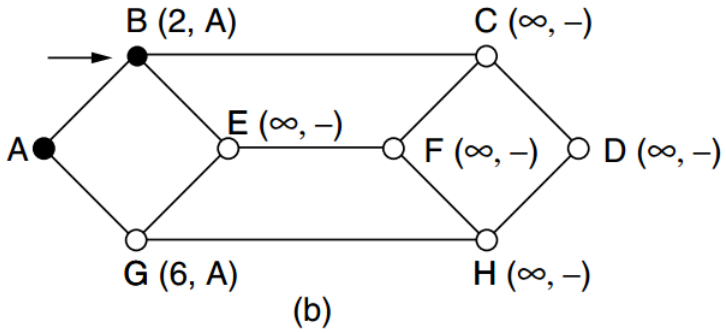
- Dijkstra's algorithm can be used to basically compute a sink tree on a graph through shortest path computation:
  - Each link is assigned a non-negative weight/distance
  - Shortest path is the one with lowest total weight
  - Using weights of 1 on arcs gives paths with fewest hops

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

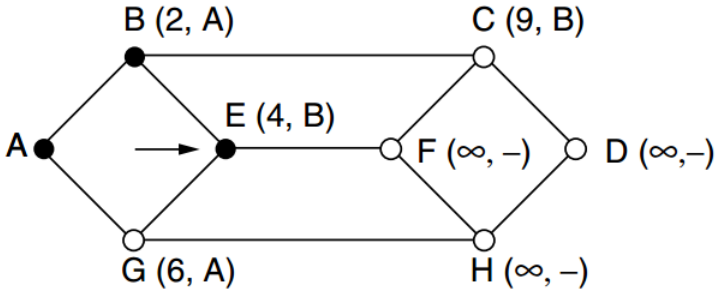
Example



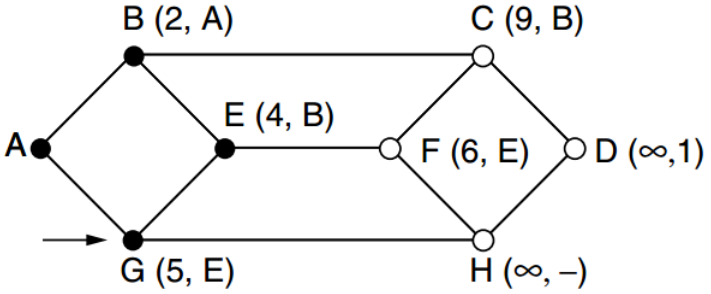
(a)



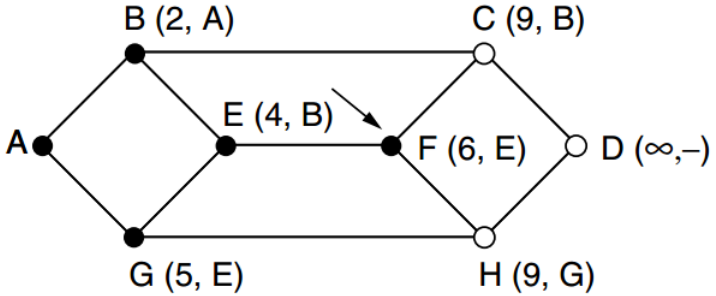
(b)



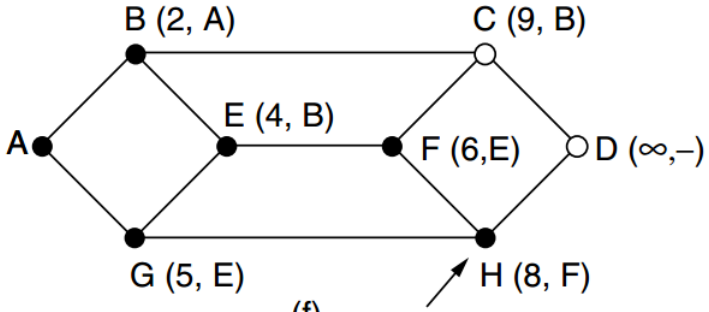
(c)



(d)



(e)



(f)

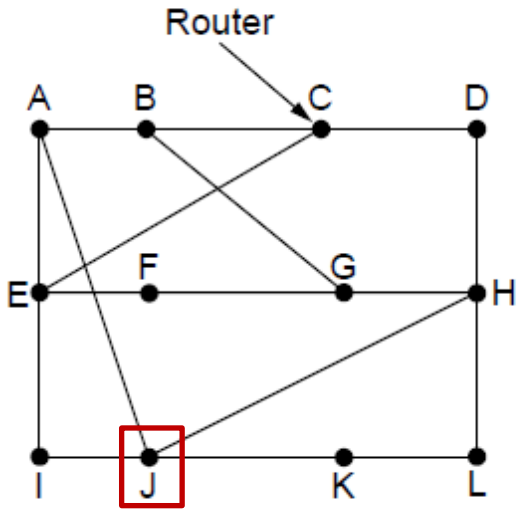
Distance Vector Routing

- A dynamic algorithm
- Each router maintains a table which includes the best known distance to each destination and which line to use to get there
- Tables are exchanged with neighbouring routers
- Idea: "Global information shared locally"

Distance Vector Routing Contd

- Algorithm Overview:
  - Each node knows distance of links to its neighbors
  - Each node advertises vector of lowest known distances to all neighbors
  - Each node uses received vectors to update its own
  - Repeat periodically

Distance Vector Routing Contd



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		

New vector for J

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