UNIVERSITY OF Southampton
School of Electronics and Computer Science

Network Layer 1

ELEC3227/ELEC6255

Alex Weddell
asw@ecs.soton.ac.uk

Source: ThinkStock

Overview

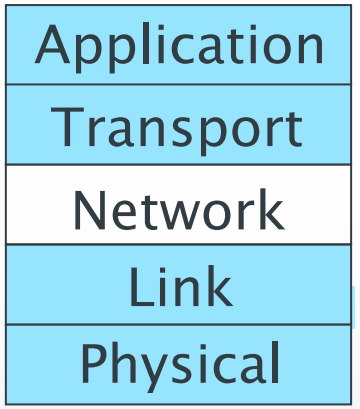
- The Network Layer.
- Connection-oriented vs. Connectionless philosophies.
- Store-and-forward packet switching.
- Virtual circuit and datagram-based routing.
- Comparison between virtual circuits/datagrams.
- Basic routing schemes:
  - Flooding
  - Distance Vector Routing (DVR)
  - Link State Routing (LSR)

UNIVERSITY OF Southampton
School of Electronics and Computer Science

The 5-layer Model

- Network is **below** the transport layer and **above** the link layer.
- **Relies on services** provided by the Link layer.
  - Interconnects between adjacent network nodes.
- **Provides services** to the Transport layer.
  - Packet delivery/switching.

- Responsible for delivering **packets** between endpoints over **multiple links.**

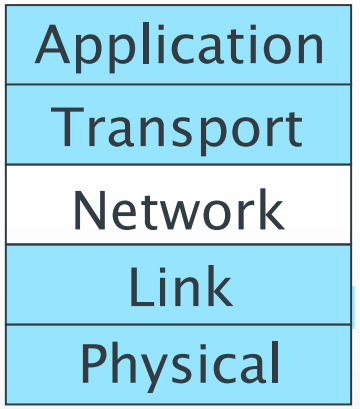


UNIVERSITY OF Southampton
School of Electronics and Computer Science

The Network Layer

- Deals with end-to-end transmission.
- Needs to know the network topology.
  - Chooses appropriate paths.
- Carefully chooses routes.
  - Ideally avoids congested links/routers.

- Services provided:
  - Independent on router technology.
  - Shielded from complexity of routes (number, type, topologies).
  - Use uniform addressing scheme, even across wider networks.



Connectionless vs. Connection-oriented

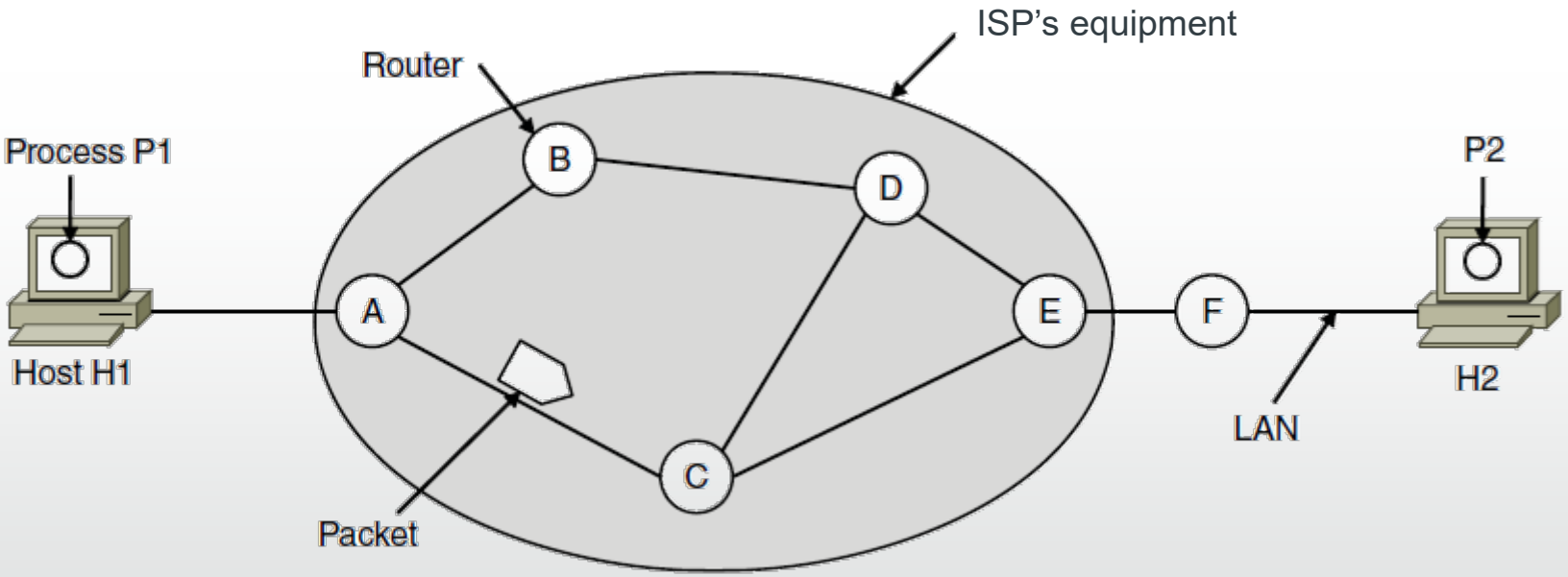
- Connectionless philosophy
  - Routers have one role: moving packets.
  - Networks are inherently unreliable.
  - Hence **hosts** should look after error and flow control themselves.
- Connection-oriented philosophy
  - Similar to traditional phone networks.
  - Networks should offer a reliable, connection-based service.
  - Quality of Service (QoS) essential for voice/video… *[or is it?]*
- Still no agreement – diversity of services has a range of requirements.
  - Internet communications use IP (Internet Protocol), connectionless.
  - Connections across internet are *virtual* – data delivered even though network layer provides a connectionless service (transport layer provides a virtual connection).

Connection-oriented vs. Connectionless

- Connectionless
  - Primitives needed: send packet, receive packet. No set-up needed.
  - No packet ordering, flow control.
  - Just send the message, hope it gets there, and don't worry if it doesn't (or arrives late)!
  - Packets known as "datagrams".
  - Each packet carries full destination address, as independent of predecessors.
- Connection-based
  - Set up a connection (route) for packets.
  - Packets carry a connection identifier – don't need complete destination address.
  - Overheads associated, but "guarantees" route.

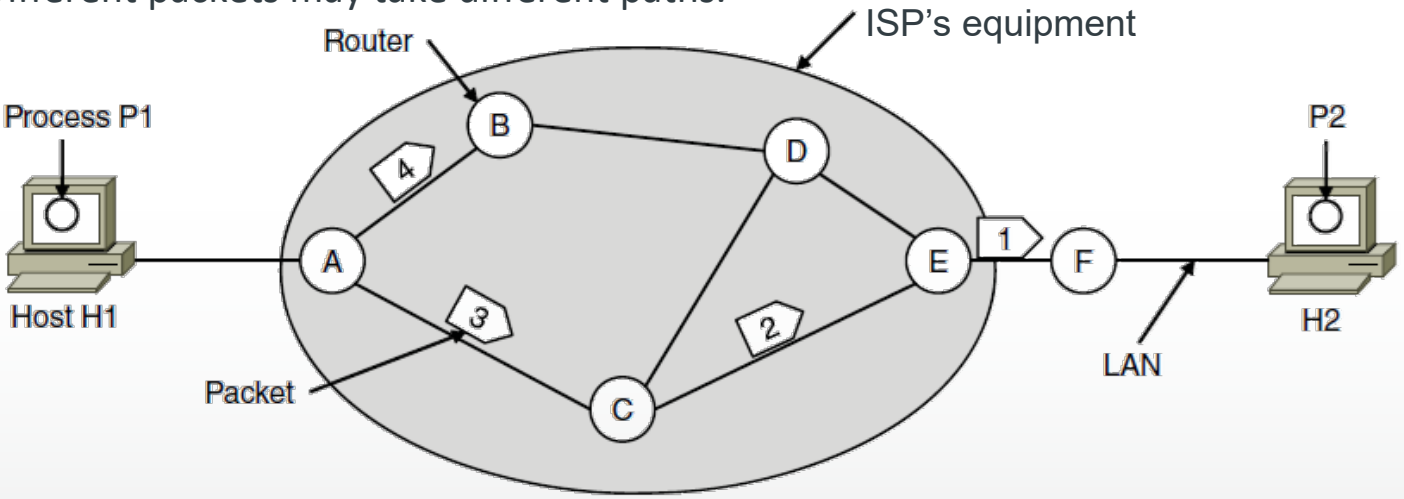
Store-and-Forward Packet Switching

- **Hosts** send **packets** into the network.
- Packets are **forwarded** by **routers.**



Connectionless Service - Datagrams

- Packet is forwarded using destination address inside it.
  - Different packets may take different paths.



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

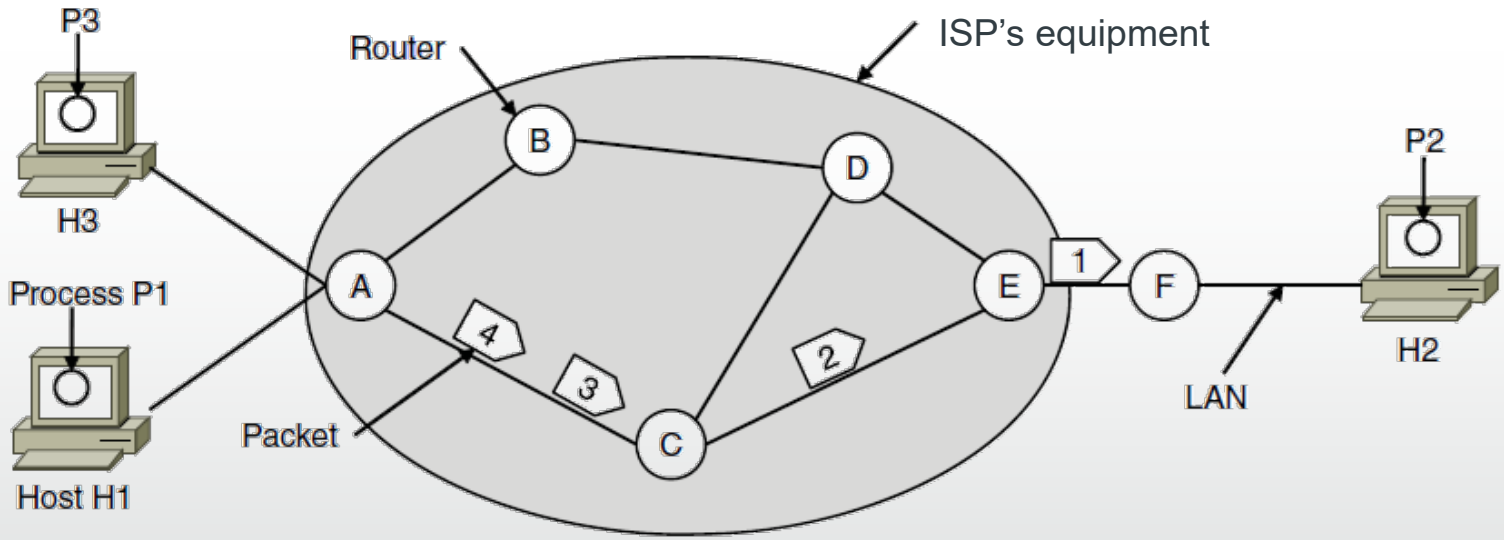
A	A
B	A
C	☒
D	E
E	E
F	E

E's Table

A	C
B	D
C	C
D	D
E	☒
F	F

Connection-Oriented – Virtual Circuits

- Packet is forwarded along a virtual circuit using tag inside it.
- Virtual circuit (VC) is set up ahead of time.
- Use **label switching** to avoid conflicts.



A's table

H1	1
H3	1

In

C	1
C	2

Out

C's Table

A	1
A	2

E	1
E	2

E's Table

C	1
C	2

F	1
F	2

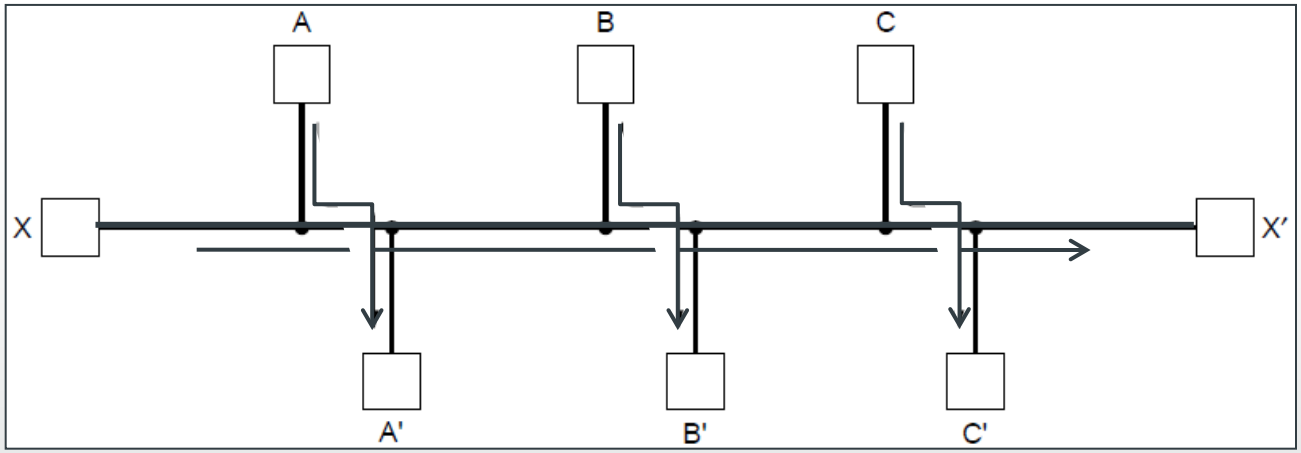
Comparison of Virtual Circuits/Datagrams

Issue	Datagram network	Virtual-circuit network
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

UNIVERSITY OF Southampton
School of Electronics and Computer Science

Routing vs. Forwarding

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



- **Forwarding** is the sending of packets along a path

Flooding

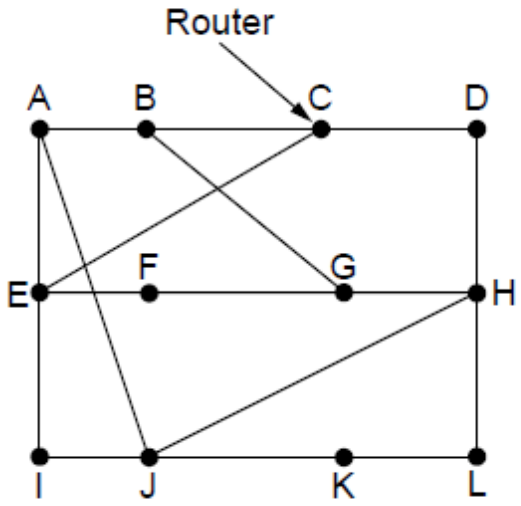
- A **simple** method to send a packet to **all** network nodes
- Each node floods a new packet received on an incoming link by sending it out to **all** of the other links
- Nodes need to keep track of flooded packets to stop the flood; even using a **hop limit** can blow up exponentially.

Distance Vector Routing

- **Distance vector** is a distributed routing algorithm
  - Shortest path computation is split across nodes

- Algorithm:
  - 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



Network

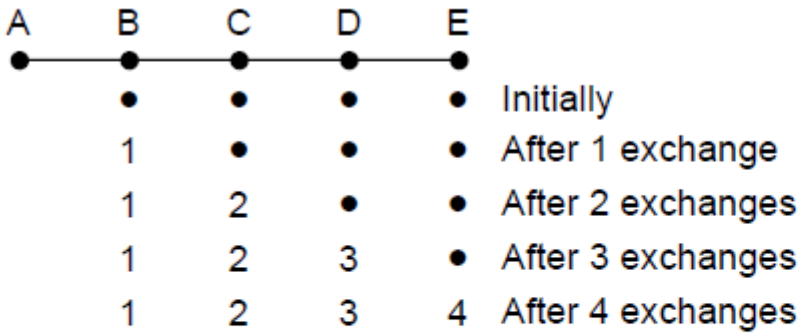
To	A	I	H	K	New estimated delay from J	Line
A	0	24	20	21	8	A
B	12	36	31	28	20	A
C	25	18	19	36	28	I
D	40	27	8	24	20	H
E	14	7	30	22	17	I
F	23	20	19	40	30	I
G	18	31	6	31	18	H
H	17	20	0	19	12	H
I	21	0	14	22	10	I
J	9	11	7	10	0	–
K	24	22	22	0	6	K
L	29	33	9	9	15	K
	JA delay is 8	JI delay is 10	JH delay is 12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

New vector for J

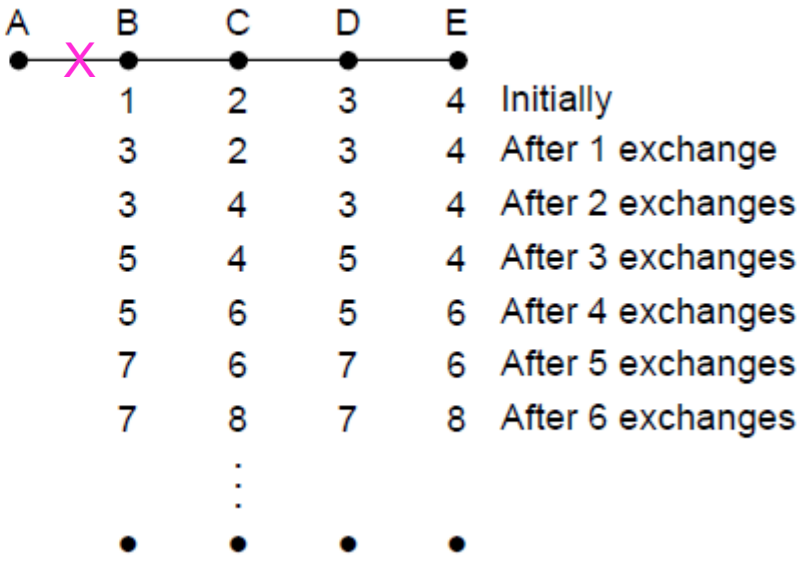
Vectors received at J from Neighbors A, I, H and K

The "Count-to-Infinity" Problem

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



Good news of a path to *A* spreads quickly



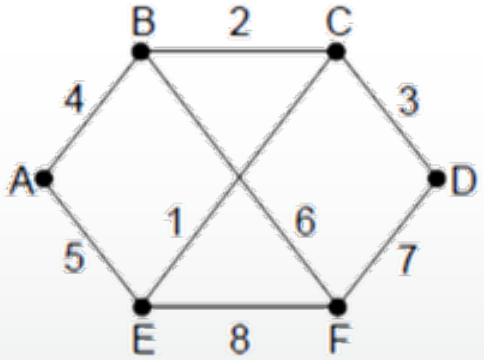
Bad news of no path to *A* is learned slowly

Link State Routing

- **Link state** is an alternative to distance vector
  - More computation but simpler dynamics
  - Widely used in the Internet (OSPF, IS-IS)
- Algorithm:
  - Each node **floods** information about its neighbors in LSPs (Link State Packets)
  - All nodes learn the **full network graph**
  - Each node **runs Dijkstra's algorithm** to compute the path to take for each destination
- The 5 steps:
  1. Discover neighbours and learn network addresses
  2. Set distance or cost metric to each of its neighbours
  3. Construct a packet telling all it has just learned
  4. Send this packet to and receive packets from all other routers
  5. Compute shortest path to every other router

Link State Routing

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



Network

A	
Seq.	
Age	
B	4
E	5

B	
Seq.	
Age	
A	4
C	2
F	6

C	
Seq.	
Age	
B	2
D	3
E	1

D	
Seq.	
Age	
C	3
F	7

E	
Seq.	
Age	
A	5
C	1
F	8

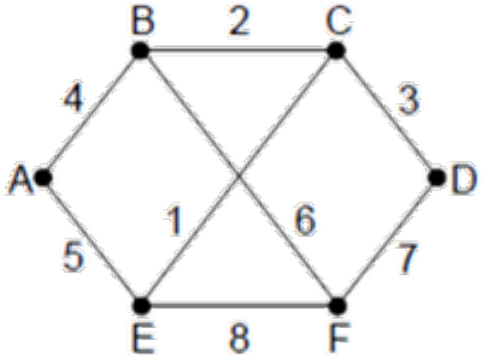
F	
Seq.	
Age	
B	6
D	7
E	8

LSP for each node

UNIVERSITY OF Southampton
School of Electronics and Computer Science

Link State Routing

- Seq. number and age are used for reliable flooding
  - New LSPs are acknowledged on the lines they are received and sent on all other lines
  - Example shows the LSP database at router B



			Send flags			ACK flags			
Source	Seq.	Age	A	C	F	A	C	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	

Summary

- The Network Layer.
- Connection-oriented vs. Connectionless philosophies.
- Store-and-forward packet switching.
- Virtual circuit and datagram-based routing.
- Comparison between virtual circuits/datagrams.
- Basic routing schemes:
  - Flooding
  - Distance Vector Routing (DVR)
  - Link State Routing (LSR)

Acknowledgment

- Lecture slides copied/revised from content provided by Andrew Tanenbaum and David Wetherall.