EEC-484/584 Computer Networks

Lecture 11

Wenbing Zhao

wenbing@ieee.org

(Lecture nodes are based on materials supplied by Dr. Louise Moser at UCSB and Prentice-Hall)

Outline

- Midterm #1 results
- The Network Layer
  - Network Layer Design Issues
  - Routing Algorithms

EEC484 Midterm#1 Results

P1	P2	P3	P4	P5	P6	P7	P8	Total		Normalized Total
18	18	10	10	10	8	10	10	94		104.434
20	18	10	10	10	4	10	10	92		102.212
20	20	10	10	10	10	10	0	90	A	100
18	12	10	10	10	10	10	10	90		100
16	14	10	10	10	5	8	8	81		90
18	12	10	7	10	5	0	3	65	B-	72.222
20	12	10	10	10	5	0	0	67		74.444
14	17	10	9.5	5	0	0	2	57.5	C	63.8825
12	10	5	10	5	10	0	0	52	D &C-	57.772
12	14	0	6	10	0	0	0	42		46.662

EEC584 Midterm#1 Results

P1	P2	P3	P4	P5	P6	P7	P8		Total
18	18	10	10	10	10	9	10		**95**
18	18	10	10	10	10	8	10		**94**
20	16	10	9	10	10	10	6	A	**91**
14	16	10	10	10	10	10	10		**90**
20	18	8	8.5	10	7	8	10		**89.5**
18	18	10	10	10	10	6	5		**87**
16	16	10	10	10	3	10	10		**85**
18	20	10	4	9.5	5	10	8		**84.5**
16	18	10	7	10	10	2	8		**81**
16	14	10	10	10	0	10	9	A-	**79**
20	12	4	10	10	10	10	3		**79**
16	16	10	7	8	10	6	5		**78**
14	14	10	10	10	5	10	5		**78**
16	14	10	8	10	0	10	8		**76**

EEC584 Midterm#1 Results

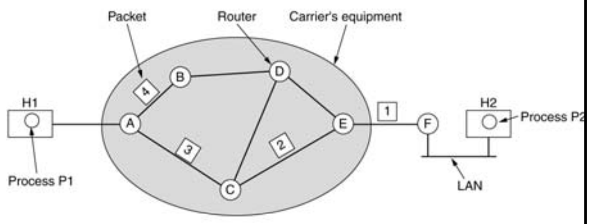
P1	P2	P3	P4	P5	P6	P7	P8		Total
16	18	10	6	10	5	8	2		**75**
18	16	10	10	10	5	0	6		**75**
14	14	10	10	10	0	8	6	B+	**72**
14	14	10	10	10	0	2	10		**70**
12	12	8	0	10	10	7	8		**67**
14	12	10	6.5	5	0	9	10		**66.5**
18	12	10	10	10	0	3	2		**65**
14	18	10	0	10	0	8	4	B	**64**
18	12	10	9	5	10	0	0		**64**
18	14	8	6	10	0	2	4		**62**
14	12	10	8	10	0	0	0		**54**
12	18	8	1	5	0	2	8	C	**54**
16	10	5	7	10	0	0	7		**55**
10	14	10	0	5	0	4	6		**49**

Network Layer

- Main concern: end-to-end transmission
  - Perhaps over many hops at intermediate nodes
- Network Layer Design Issues
  - Services Provided to the Transport Layer
    - Routing
    - Congestion control
    - Internetworking – connection of multiple networks
  - Goals – services should
    - Be independent of subnet technologies
    - Shield transport layer from number, type, topology of subnets
    - Uniform network addresses across LAN/WAN

Connectionless Service

- **Routing within a datagram subnet**
  - Router has table telling which outgoing line to use for each possible destination router
  - Each datagram has full destination address
  - When packet arrives, router looks up outgoing line to use and transmits packet



Dest.	A's table initially (Line)	A's table later (Line)	C's table (Line)	E's table (Line)
A	–	–	A	C
B	B	B	A	D
C	C	C	–	C
D	B	B	D	D
E	C	B	E	–
F	C	B	E	F

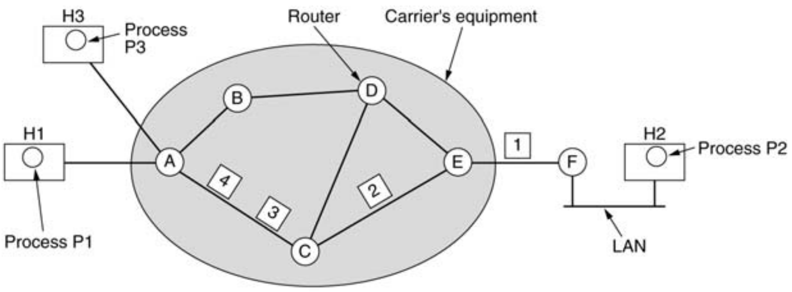
Connection-Oriented Service

- **Routing within a virtual-circuit subnet** - Each router must remember where to forward packets for each currently open Virtual Circuit (VC) passing through it
  - Each router must maintain a table with one entry per open VC
  - Each packet must contain VC number field in its header
  - Router must forward packet to correct router based on
    - Line on which it arrived and VC number

Connection-Oriented Service

- **Problem**: each machine chooses VC number independently when connection set up
  - Same VC number is likely on two different paths through same intermediate router
- **Solution**: to create new outbound virtual circuit, choose lowest circuit number not in use and overwrite circuit number in set up packet with that number

Connection-Oriented Service



A's table

In		Out	
H1	1	C	1
H3	1	C	2

C's table

A	1	E	1
A	2	E	2

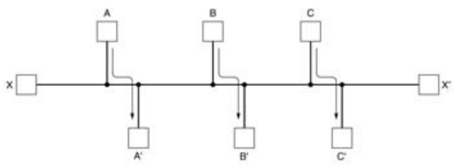
E's table

C	1	F	1
C	2	F	2

Comparison of Virtual-Circuit 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

Routing Algorithms



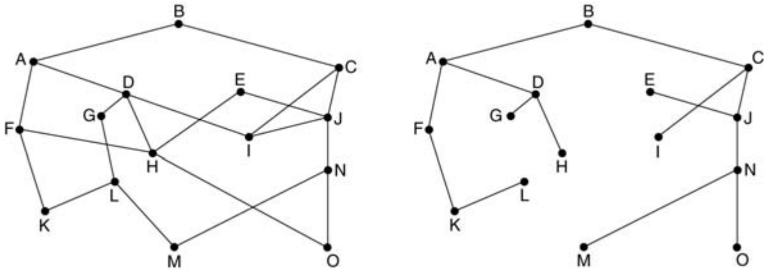
- Desirable properties
  - Correctness, simplicity
  - Robustness to faults
  - Stability – converge to equilibrium
  - Fairness and optimality – hard to achieve both
- Two classes
  - Non-adaptive (static) - Route computed in advance, off-line, downloaded to routers
  - Adaptive (dynamic) - Route based on measurements or estimates of current traffic and topology

Routing Algorithms

- The Optimality Principle
- Shortest Path Routing
- Flooding
- Distance Vector Routing
- Link State Routing
- Hierarchical Routing
- Broadcast Routing
- Multicast Routing
- Routing for Mobile Hosts
- Routing in Ad Hoc Networks
- Node Lookup in Peer-to-Peer Networks (skip)

The Optimality Principle

- **Optimality principle** – if router J is on the optimal path from router I to router K, then the optimal path from J to K also falls along the same route
- **Sink tree** – the set of optimal routes from all sources to a given destination form a tree rooted at the destination



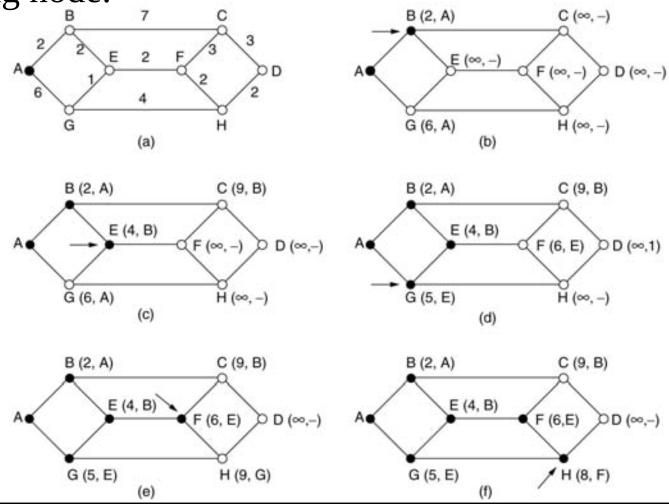
A subnet A sink tree for router B

Shortest Path Routing

- Shortest in what sense?
  - Number of hops, geographical distance, least queueing and transmission delay
- Dijkstra's Algorithm
  - Each node labeled with distance from source node along best known path
  - Initially, no paths known all nodes labeled with infinity
  - As algorithm proceeds, labels may change reflecting shortest path
  - Label may be tentative or permanent, initially, all tentative
  - When label represents shortest path from source to node, label becomes permanent

Shortest Path Routing

- Steps computing the shortest path from A to D, using Dijkstra's algorithm. The arrows indicate the working node.

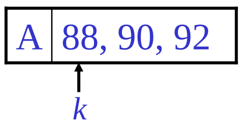


Flooding

- Idea: every incoming packet is sent out on every outgoing line except the one it arrived on
- **Adv**: flooding always chooses shortest path, no other algorithm has shorter delay
- **Disadv**: generates lots of duplicates

Flooding

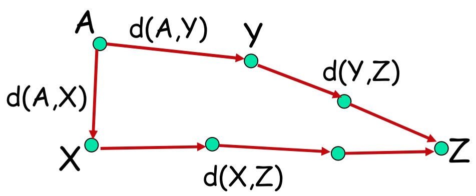
- Methods to damper this effect
  - Header of each packet has hop counter that is decremented at each hop; packet discarded when counter reaches 0
  - Source router puts sequence number in each packet it receives from its hosts
    - Each router has list for each source router telling which sequence numbers originating at that source have already been seen
    - Each list augmented with counter *k*, meaning all sequence numbers up through *k* have been seen
    - If packet is duplicate, router discards it



Distance Vector Routing

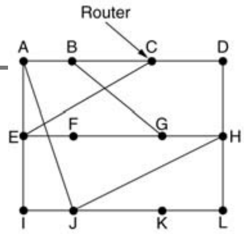
- Also called Bellman-Ford or Ford-Fulkerson. Used in ARPANET
- Idea: each router maintains a table (a vector), giving best known distance to each destination and which line to use to get there
  - Table is updated by exchanging info with neighbors
  - Table contains one entry for each router in network with
    - Preferred outgoing line to that destination
    - Estimate of time or distance to that destination
  - Once every *T* msec, router sends to each neighbor a list of estimated delays to each destination and receives same from those neighbors

Distance Vector Routing



At router A, for Z
Compute d(A,X) + d(X,Z) and
d(A,Y) + d(Y,Z), take minimum

Distance Vector Routing



(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

(b)

- Delay A to B 12ms, to C 25ms, to D 40ms, to G 18ms
- Delay J to A 8ms, to I 10ms, to H 12ms, to K 6ms
- Delay J to A to G 8+18 = 26ms
  to I to G 10+31 = 41ms
  to H to G 12+6=18ms
  to K to G 6+31=37ms

Problems in Distance Vector Routing

- Delay metric in ARPANET was queue length, did not consider bandwidth of lines
- Takes too long to converge upon router failure (count to infinity problem)

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

(a)

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
		⋮			
	∞	∞	∞	∞	

(b)