CS 3251 Homework 4

Wenqi He

November 30, 2018

4.1

(1)

The rows in each table are the distance vectors from each router's view:

w:

	w	x	y	z
w	0	5	1	1
y	1	4	0	∞
z	1	∞	∞	0

, **x:**

	w	x	y	z
x	5	0	4	6
y	1	4	0	2
z	1	6	2	0

, **y:**

	w	x	y	z
w	0	∞	1	1
x	∞	0	4	∞
y	1	4	0	2
z	1	6	2	0

, **z:**

	w	x	y	z
w	0	5	1	1
x	5	0	4	6
y	1	4	0	2
z	1	6	2	0

(2)

There will still be a "count-to-infinity" problem, because initially z routes to x through z-w-y-x, but y is unaware of it even though poison reverse is used. After y updates its route to y-z-x, it forms a closed loop of w, y and z. Wrong information will propagate along this loop repeatedly and increase the cost of each link slowly until they exceed the actual link costs, then the distance vectors will finally converge. As shown below, it takes 30 iterations in total.

1st iteration: (*x converges immediately*)

w:

	w	x	y	z
w	0	5	1	1
y	1	4	0	∞
z	1	∞	∞	0

, **x:**

	w	x	y	z
x	51	0	52	50
y	1	4	0	2
z	1	6	2	0

, **y:**

	w	x	y	z
w	0	∞	1	1
x	∞	0	4	∞
y	1	9	0	2
z	1	6	2	0

, **z:**

	w	x	y	z
w	0	5	1	1
x	5	0	4	6
y	1	4	0	2
z	1	6	2	0

2nd iteration:

w:

	w	x	y	z
w	0	10	1	1
y	1	9	0	∞
z	1	∞	∞	0

, **y:**

	w	x	y	z
w	0	∞	1	1
x	51	0	52	50
y	1	9	0	2
z	1	6	2	0

, **z:**

	w	x	y	z
w	0	5	1	1
x	∞	0	∞	50
y	1	∞	0	2
z	1	6	2	0

3rd iteration:

w:

	w	x	y	z
w	0	10	1	1
y	1	9	0	∞
z	1	∞	∞	0

, **y:**

	w	x	y	z
w	0	∞	1	1
x	51	0	52	50
y	1	9	0	2
z	1	6	2	0

, **z:**

	w	x	y	z
w	0	10	1	1
x	∞	0	∞	50
y	1	∞	0	2
z	1	11	2	0

4th iteration:

w:

	w	x	y	z
w	0	10	1	1
y	1	9	0	∞
z	1	∞	∞	0

, **y:**

	w	x	y	z
w	0	∞	1	1
x	51	0	52	50
y	1	14	0	2
z	1	11	2	0

, **z:**

	w	x	y	z
w	0	10	1	1
x	∞	0	∞	50
y	1	∞	0	2
z	1	11	2	0

5th iteration:

w:

	w	x	y	z
w	0	15	1	1
y	1	14	0	∞
z	1	∞	∞	0

, **y:**

	w	x	y	z
w	0	∞	1	1
x	51	0	52	50
y	1	14	0	2
z	1	11	2	0

, **z:**

	w	x	y	z
w	0	10	1	1
x	∞	0	∞	50
y	1	∞	0	2
z	1	11	2	0

8th iteration:

w:

	w	x	y	z
w	0	20	1	1
y	1	19	0	∞
z	1	∞	∞	0

, **y:**

	w	x	y	z
w	0	∞	1	1
x	51	0	52	50
y	1	19	0	2
z	1	16	2	0

, **z:**

	w	x	y	z
w	0	15	1	1
x	∞	0	∞	50
y	1	∞	0	2
z	1	16	2	0

14th iteration:

w:

	w	x	y	z
w	0	30	1	1
y	1	29	0	∞
z	1	∞	∞	0

, **y:**

	w	x	y	z
w	0	∞	1	1
x	51	0	52	50
y	1	29	0	2
z	1	26	2	0

, **z:**

	w	x	y	z
w	0	25	1	1
x	∞	0	∞	50
y	1	∞	0	2
z	1	26	2	0

20th iteration:

w:

	w	x	y	z
w	0	40	1	1
y	1	39	0	∞
z	1	∞	∞	0

, **y:**

	w	x	y	z
w	0	∞	1	1
x	51	0	52	50
y	1	39	0	2
z	1	36	2	0

, **z:**

	w	x	y	z
w	0	35	1	1
x	∞	0	∞	50
y	1	∞	0	2
z	1	36	2	0

26th iteration:

w:

	w	x	y	z
w	0	50	1	1
y	1	49	0	∞
z	1	∞	∞	0

, **y:**

	w	x	y	z
w	0	∞	1	1
x	51	0	52	50
y	1	49	0	2
z	1	46	2	0

, **z:**

	w	x	y	z
w	0	45	1	1
x	∞	0	∞	50
y	1	∞	0	2
z	1	46	2	0

27th iteration: (*z finally converges*)

w:

	w	x	y	z
w	0	50	1	1
y	1	49	0	∞
z	1	∞	∞	0

, **y:**

	w	x	y	z
w	0	∞	1	1
x	51	0	52	50
y	1	49	0	2
z	1	46	2	0

, **z:**

	w	x	y	z
w	0	50	1	1
x	∞	0	∞	50
y	1	∞	0	2
z	1	50	2	0

28th iteration:

w:

	w	x	y	z
w	0	50	1	1
y	1	49	0	∞
z	1	50	∞	0

, **y:**

	w	x	y	z
w	0	∞	1	1
x	51	0	52	50
y	1	53	0	2
z	1	50	2	0

29th iteration: (*x converges*)

w:

	w	x	y	z
w	0	51	1	1
y	1	53	0	∞
z	1	50	∞	0

, **y:**

	w	x	y	z
w	0	∞	1	1
x	51	0	52	50
y	1	53	0	2
z	1	50	2	0

30th iteration: (y *converges*)

y:

	w	x	y	z
w	0	51	1	1
x	51	0	52	50
y	1	52	0	2
z	1	50	2	0

4.2

(1)

W would use WABX or WACX to reach X and WACY to reach Y. X would use XBAW or XCAW to reach W and XCY to reach Y.

(2)

A should advertise path AW to B only, and AV to both B and C. C receives AS routes: CY, CX, CB, CBX, CBA, CBAW, CBAV, CA, CAV

4.17

a

First create a variable a_0 and assign to it the identification number of the first datagram. Then, whenever a new packet is captured, if the identification number equals $a_i + 1$ for any a_i, increment that variable: $a_i = a_i + 1$, otherwise assign the number to a new variable. Because the identification numbers of the packets from each host is contiguous, each variable keeps track of the datagrams coming from a single host. Therefore, the number of variables is exactly the number of unique hosts.

b

No, it would not work. Because the above technique relies entirely on the identification numbers being sequential. If the number is random, then there would be no way to tell if a new datagram comes from the same host as any previously received datagrams.

6.11

a

For a node to succeed in any given time slot, only that node can transmit in that time slot, that is

$$P(\text{A succeeds}) = P(\text{A transmitting}) \cdot P(\text{B, C, D not transmitting}) = p(1-p)^3$$

If slot 5 is the first time it succeeds, it must have failed in slots 1 - 4, therefore the probability is

$$\boxed{P(\text{success}) \cdot P(\text{fail})^4 = p(1-p)^3 \left[1 - p(1-p)^3\right]^4}$$

b

Since the events that any of the nodes succeeds are disjoint, the probablity is simply

$$P = P(A) + P(B) + P(C) + P(D) = 4p(1-p)^3$$

c

$$P = 4p(1-p)^3\Big[1 - 4p(1-p)^3\Big]^2$$

d

The probablity that the first success occurs in slot n is a geometric distribution

$$P(n) = 4p(1-p)^3\Big[1 - 4p(1-p)^3\Big]^{n-1}$$

The expected number of time slots needed to transmit 1 packet is the expected number of time slots needed for the first success:

$$E[n] = \frac{1}{4p(1-p)^3}$$

If no collision occured, each time slot could allow one packet to transmit. So the efficiency is

$$\frac{\text{\# packets actually sent}}{\text{\# packets could have been sent}} = \frac{1}{E(n)} = 4p(1-p)^3$$

6.15

a

No. F is in the same subnet as E, so E can get F's MAC address through ARP and send the datagram to F directly. The source IP and MAC addresses are E's IP and MAC addresses, and the destination IP and MAC addresses are F's.

b

No, because B is not in the same subnet as E. The source and destination IP are E and B's IP. The source MAC address is E's MAC address, the destination MAC address is the router R1's MAC address.

c

Since S1's forwarding table contains an entry for B, it will just forward the message through the corresponding port instead of broadcasting it, therefore R1 will not receive the message.

No, because B has already learned A's MAC address from the ARP request message it just received from A. S1 would just forward the message to A, because it has also learned A's MAC address from the ARP request it received earlier.

6.21

A: IP: 111.111.111.111, MAC: 1A-1A-1A-1A-1A-1A
R1 (Subnet 1): IP: 111.111.111.110, MAC: 2A-2A-2A-2A-2A-2A
R1 (Subnet 2): IP: 222.222.222.222, MAC: 1B-1B-1B-1B-1B-1B
R2 (Subnet 2): IP: 222.222.222.220, MAC: 2B-2B-2B-2B-2B-2B
R2 (Subnet 3): IP: 333.333.333.330, MAC: 2C-2C-2C-2C-2C-2C
F: IP: 333.333.333.333, MAC: 1C-1C-1C-1C-1C-1C

i

source MAC: 1A-1A-1A-1A-1A-1A, destination MAC: 2A-2A-2A-2A-2A-2A
source IP: 111.111.111.111, destination IP: 333.333.333.333

ii

source MAC: 1B-1B-1B-1B-1B-1B, destination MAC: 2B-2B-2B-2B-2B-2B
source IP: 111.111.111.111, destination IP: 333.333.333.333

iii

source MAC: 2C-2C-2C-2C-2C-2C, destination MAC: 1C-1C-1C-1C-1C-1C
source IP: 111.111.111.111, destination IP: 333.333.333.333

6.28

Suppose the EE hosts are in subnet 111.111.111.111/24, and the CS hosts are in subnet 222.222.222.222/24. The router's interface in EE subnet has IP addresss 111.111.111.000 and MAC address AA-AA-AA-AA-AA-AA, and its interface in CS subnet has IP 222.222.222.000 and MAC address BB-BB-BB-BB-BB-BB.

Suppose EE host A with IP 111.111.111.111 and MAC address 1A-1A-1A-1A-1A-1A wants to send an IP datagram to CS hsot B with IP 222.222.222.222 and MAC address 2B-2B-2B-2B-2B-2B. The steps taken are:

1. Host A's network layer creates an IP datagram with source IP 111.111.111.111 and destination IP 222.222.222.222.
2. A's link layer broadcasts an ARP request for the MAC address of the router, and the router responds with its MAC address AA-AA-AA-AA-AA.
3. A's link layer encapsulates the IP datagram in an Ethernet frame with source MAC address 1A-1A-1A-1A-1A-1A and destination MAC address AA-AA-AA-AA-AA-AA, and sends out the frame.
4. The router receives the frame, extracts the IP datagram and passes it up to the network layer. From the forwarding table, the network layer determines that it should forward the datagram to the port corresponding to the CS subnet.
5. The router's link layer broadcasts an ARP request in the CS subnet for B's MAC address, and B responds with 2B-2B-2B-2B-2B-2B.
6. The router's link layer then encapsulates the IP datagram in a new Ethernet frame with source MAC address BB-BB-BB-BB-BB-BB and destination MAC address 2B-2B-2B-2B-2B-2B, and sends out the frame.
7. Finally B receives the frame, extracts the IP datagram and passes it up to its network layer.