

Computer Network HOMEWORK 2, Due Date Dec 22, 2023

Name: 林群賀
Dept.: 大氣四
ID: 109601003

Q1.

In our **rdt** protocols, why did we need to introduce sequence numbers?

Ans:

Sequence numbers are required for a receiver to find out whether an arriving packet contains new data or is a retransmission.

In our **rdt** protocols, why did we need to introduce timers?

Ans:

Timers were introduced to detect lost packets. If the ACK for a transmitted packet is not received within the duration of the timer for the packet, the packet (or its ACK or NACK) is assumed to have been lost. Hence, the packet is retransmitted.

Q2.

True or False

1. Host A is sending Host B a large file over a TCP connection. Assume Host B has no data to send Host A. Host B will not send acknowledgments to Host A because Host B cannot piggyback the acknowledgments on data.
2. The size of the TCP *rwnd* never changes throughout the duration of the connection.
3. Suppose Host A is sending Host B a large file over a TCP connection. The number of unacknowledged bytes that A sends cannot exceed the size of the receive buffer.
4. Suppose Host A is sending a large file to Host B over a TCP connection. If the sequence number for a segment of this connection is m , then the sequence number for the subsequent segment will necessarily be $m+1$.
5. The TCP segment has a field in its header for *rwnd*.

Ans:

1. False.
2. False.
3. True.
4. False.
5. True.

Q3.

UDP and TCP use 1s complement for their checksums. Suppose you have the following three 8-bit bytes: 01010011, 01100110, 01110100.

1. What is the 1s complement of the sum of these 8-bit bytes? (Note that although UDP and TCP use 16-bit words in computing the checksum, for this problem you are being asked to consider 8-bit sums.) Show all work.
2. Why is it that UDP takes the 1s complement of the sum; that is, why not just use the sum?
3. With the 1s complement scheme, how does the receiver detect errors?
4. Is it possible that a 1-bit error will go undetected?
5. How about a 2-bit error?

Ans:

1. Note, wrap around if overflow

```

0 1 0 1 0 0 1 1
+ 0 1 1 0 0 1 1 0
-----
1 0 1 1 1 0 0 1

```

and then

```

1 0 1 1 1 0 0 1
+ 0 1 1 1 0 1 0 0
-----
0 0 1 0 1 1 1 0

```

One's complement = 1 1 0 1 0 0 0 1.

2. To detect errors, the receiver adds the four words (the three original words and the checksum).
3. If the sum contains a zero, the receiver knows there has been an error.
4. All one-bit errors will be detected.
5. Two-bit errors can be undetected (e.g., if the last digit of the first word is converted to a 0 and the last digit of the second word is converted to a 1).

Q4.

In our discussion of TCP futures in Section 3.7, we noted that to achieve a throughput of 10 Gbps, TCP could only tolerate a segment loss probability of $2 \cdot 10^{-10}$ (or equivalently, one loss event for every 5,000,000,000 segments). Show the derivation for the values of $2 \cdot 10^{-10}$ (1 out of 5,000,000) for the RTT and MSS values given in Section 3.7. If TCP needed to support a 100 Gbps connection, what would the tolerable loss be?

Ans:

```

RTT = 100ms = 0.1s
MSS = 1,000,000 bytes
Throughput = 10 Gbps = 10,000,000,000 bits/s

```

$$\text{Throughput} = \text{MSS} / \text{RTT} * (1 - \text{Loss})$$

$$10,000,000,000 = 1,000,000 / 0.1 * (1 - \text{Loss})$$

$$\text{Loss} = 0.0000000002 = 2 * 10^{-10}$$

$$\text{RTT} = 100\text{ms} = 0.1\text{s}$$

$$\text{MSS} = 1,000,000 \text{ bytes}$$

$$\text{Throughput} = 100 \text{ Gbps} = 100,000,000,000 \text{ bits/s}$$

$$\text{Throughput} = \text{MSS} / \text{RTT} * (1 - \text{Loss})$$

$$100,000,000,000 = 1,000,000 / 0.1 * (1 - \text{Loss})$$

$$\text{Loss} = 0.00000000002 = 2 * 10^{-11}$$

Q5.

Consider a datagram network using 32-bit host addresses. Suppose a router has four links, numbered 0 through 3, and packets are to be forwarded to the link interfaces as follows:

Destination Address Range	Link Interface
11100000 00000000 00000000 00000000 through 11100000 00111111 11111111 11111111	0
11100000 01000000 00000000 00000000 through 11100000 01000000 11111111 11111111	1
11100000 01000001 00000000 00000000 through 11100001 01111111 11111111 11111111	2
otherwise	3

- Provide a forwarding table that has four entries, uses longest prefix matching, and forwards packets to the correct link interfaces.
- Describe how your forwarding table determines the appropriate link interface for datagrams with destination addresses:

```
11001000 10010001 01010001 01010101
11100001 01000000 11000011 00111100
11100001 10000000 00010001 01110111
```

Ans:

a.

Prefix Match	Link Interface
11100000 00	0
11100000 01000000	1
1110000	2
11100001 1	3
otherwise	3

or

Prefix Match	Link Interface
11100000 00	0
11100000 01000000	1
11100000	2
11100001 1	2
otherwise	3

b.

- Prefix match for first address is fifth entry: link interface 3
- Prefix match for second address is third entry: link interface 2
- Prefix match for third address is fourth entry: link interface 3

Q6.

Consider a subnet with prefix 128.119.40.128/26.

Give an example of one IP address (of form xxx.xxx.xxx.xxx) that can be assigned to this network.

Any IP address in range 128.119.40.128 to 128.119.40.191.

Suppose an ISP owns the block of addresses of the form 128.119.40.64/25. Suppose it wants to create four subnets from this block, with each block having the same number of IP addresses. What are the prefixes (of form a.b.c.d/x) for the four subnets?

Ans:

Any IP address in range 128.119.40.128 to 128.119.40.191.

Four equal size subnets:

WAN Side	LAN Side
24.34.112.235, 5001	192.168.1.1, 3346
24.34.112.235, 5002	192.168.1.2, 3445
24.34.112.235, 5003	192.168.1.2, 3446
24.34.112.235, 5004	192.168.1.3, 3545
24.34.112.235, 5005	192.168.1.3, 3546

Q8.

Consider the SDN OpenFlow network shown in Figure 4.30 . Suppose that the desired forwarding behavior for datagrams arriving at s2 is as follows:

- any datagrams arriving on input port 1 from hosts h5 or h6 that are destined to hosts 10.3.*.* or h2 should be forwarded over output port 2;
- any datagrams arriving on input port 2 from hosts 10.3.*.* or h2 that are destined to hosts h5 or h6 should be forwarded over output port 1;
- any arriving datagrams on input ports 1 or 2 and destined to hosts h3 or h4 should be delivered to the host specified;
- hosts h3 and h4 should be able to send datagrams to each other.

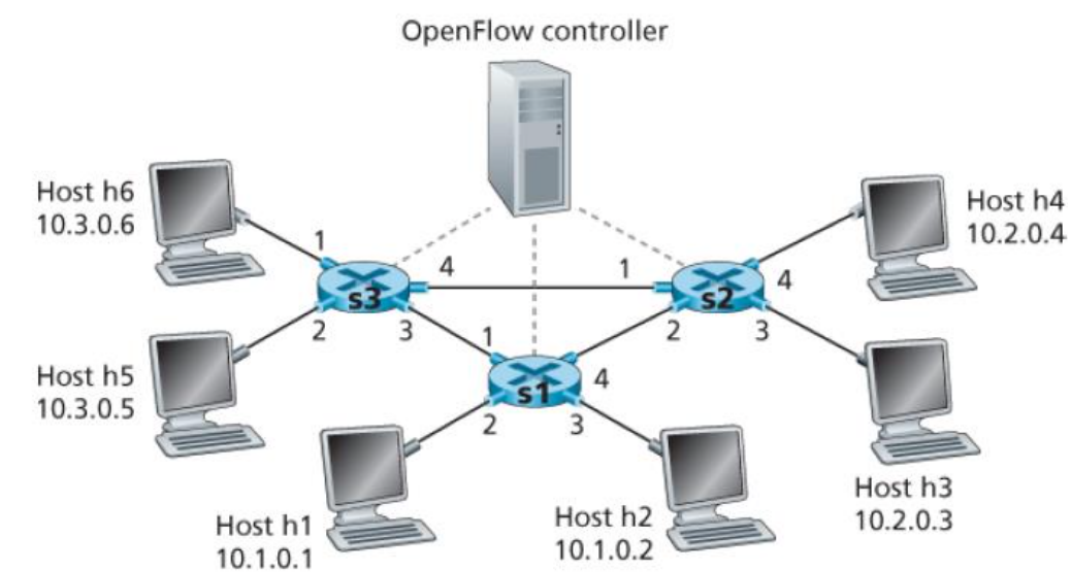


Figure 4.30 OpenFlow match-plus-action network with three packet switches, 6 hosts, and an OpenFlow controller

Ans:

S2 Flow Table

Match	Action
Ingress Port = 1; IP Src = 10.3.*.*; IP Dst = 10.1.*.*	Forward (2)
Ingress Port = 2; IP Src = 10.1.*.*; IP Dst = 10.3.*.*	Forward (1)
Ingress Port = 1; IP Dst = 10.2.0.3	Forward (3)
Ingress Port = 2; IP Dst = 10.2.0.3	Forward (3)
Ingress Port = 1; IP Dst = 10.2.0.4	Forward (4)
Ingress Port = 2; IP Dst = 10.2.0.4	Forward (4)
Ingress Port = 4	Forward (3)
Ingress Port = 3	Forward (4)

Q9.

Consider again the SDN OpenFlow network shown in Figure 4.30 . Suppose we want switch s2 to function as a firewall. Specify the flow table in s2 that implements the following firewall behaviors (specify a different flow table for each of the four firewalling behaviors below) for delivery of datagrams destined to h3 and h4. You do not need to specify the forwarding behavior in s2 that forwards traffic to other routers.

- Only traffic arriving from hosts h1 and h6 should be delivered to hosts h3 or h4 (i.e., that arriving traffic from hosts h2 and h5 is blocked).
- Only TCP traffic is allowed to be delivered to hosts h3 or h4 (i.e., that UDP traffic is blocked).
- Only traffic destined to h3 is to be delivered (i.e., all traffic to h4 is blocked).
- Only UDP traffic from h1 and destined to h3 is to be delivered. All other traffic is blocked.

Ans:

Step1

A flow table is a more comprehensive routing table. A flow table allows more variables to determine the outbound interface of a packet.

It also allows dropping of a packet and modifying packet header values.

The match of a flow table is the matching of the packet header values to the table entry values.

The action of a flow table is the dropping, modify or forwarding of the packet.

Step2:

The first requirement When a packet is destined for h3 destination IP address 10.2.0.3. When a packet is destined for h4 destination IP address 10.2.0.4. When a packet is sent from h1 it will have IP address 10.1.0.1 and should be forwarded. When a packet is sent from h6 it will have IP address 10.3.0.6 and should be forwarded.

If the packet doesn't find a match in the flow table, it will be dropped. In order for the packets from h5 and h1 to be dropped, they should simply not be included in the table.

The table is then:

Match	Action
-------	--------

Match	Action
Src IP: 10.1.0.1,Dst IP: 10.2.0.3	Forward(3)
Src IP: 10.1.0.1,Dst IP: 10.2.0.4	Forward(4)
Src IP: 10.3.0.6,Dst IP: 10.2.0.3	Forward(3)
Src IP: 10.3.0.6,Dst IP: 10.2.0.4	Forward(4)

Step3:

The second requirement To identify the traffic protocol, the source port field can be used. If the source port field is set to TCP, then the traffic is TCP traffic.

The table is then:

Match	Action
SRC port = TCP,Dst IP: 10.2.0.3	Forward(3)
SRC port = TCP,Dst IP: 10.2.0.4	Forward(4)

Step4:

The third requirement: If the table entries are only for h3, all traffic to h4 will be dropped.

The table is then:

Match	Action
Dst IP: 10.2.0.3	Forward(3)

Step5:

The fourth table:

Match	Action
SRC port = UDP,Src IP: 10.1.0.1,Dst IP: 10.2.0.3	Forward(3)

Q10.

Compare and contrast link-state and distance-vector routing algorithms.

Ans:

Link-state routing algorithms use global network information, meaning that the entire network topology and all link costs can be passed as input. Distance-vector algorithms only have access to local information communicated by neighbors, making them iterative, asynchronous, and distributed