THE HONG KONG POLYTECHNIC UNIVERSITY

Department of Computing

This is an open-book examination.

(COMP5311)

Internet Infrastructure and Protocols

18 December 2007 3.5 hours

[Answer at most 7 questions in section A and both questions in section B.]

Section A: Please answer AT MOST SEVEN questions in this section [8 marks each, making up a total of 56 marks out of 100.] You should always attach a succinct explanation to your answer.

1. (IP networks) Consider an IP network in Figure 1 which is configured with only `158.132.20.0/22`. All other subnets in `158.132.0.0/16` and external networks can be reached through router R. Subnet-directed broadcast packets will be dropped by the routers, and no proxy ARP is in place. Suppose that a sniffer program (S), such as WireShark, is set up to capture all packets sent on the network. Is it possible for S to capture the packets in (a)-(d)? Explain your answers. Assume that all packets are normal (i.e., no malicious packets sent by attackers).

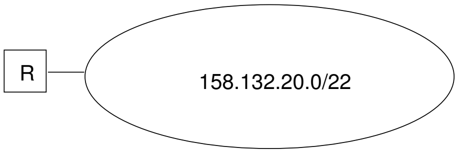

Figure 1: An IP subnet with two routers.

   (a) (2 marks) An ARP request frame with the source IP address as `158.132.21.21` and the target IP address as `158.132.28.28`.

   (b) (2 marks) An ARP request frame with the source IP address as `158.132.28.28` and the target IP address as `158.132.21.21`.

   (c) (2 marks) An ICMP packet encapsulated by an IP packet with the source IP address as `158.132.28.28` and the destination IP address as `158.132.21.21`.

   (d) (2 marks) An IP packet with the source IP address as `158.132.21.21` and the destination IP address as `158.132.23.255`.

2. (Proxy ARP and NAT) Consider the left figure in Figure 2. The IP address of a web server is hard-coded on a software on all PCs. Later, as depicted in the right figure in Figure 2, the web server is physically relocated to another subnet (the demilitarized zone, DMZ); its address is therefore reconfigured to `192.168.2.10`. However, the system administrator certainly does not wish to change the hard-coded address of the web server in all PCs. Describe how the firewall PFSense uses proxy ARP and network address translation (NAT) to solve this problem.

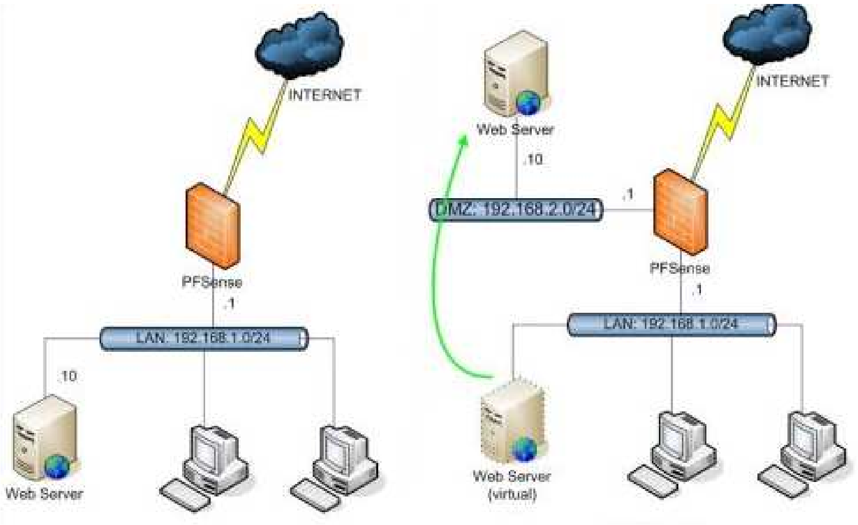

Figure 2: Create a virtual IP on `192.168.1.0/24` using proxy ARP and NAT.

3. (ICMP errors) The traceroute program uses three ICMP messages to discover the router-level forwarding path: ICMP Time Exceeded error message and ICMP Echo request and reply messages.

   The Time Exceeded Message (with type = 11 and code = 0) is sent by a router when a packet's TTL field becomes 0:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |     Code      |          Checksum             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             unused                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Internet Header + 64 bits of Original Data Datagram      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

   The Echo request (type = 8 and code = 0) is sent by a ping program, and the reply message (type = 0 and code = 0) will be sent back by the target destination:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |     Code      |          Checksum             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Identifier          |        Sequence Number        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Data ...
+-+-+-+-+-
```

To obtain the first-hop router, a ping program sends an ICMP Echo request and sets the TTL value in the IP packet to 1. Assuming that the destination is more than one router hop away, what is the packet returned by the first-hop router? Assume that the ping program is launched at `158.132.11.1` and the destination is `180.10.1.1`. Draw the payload of the received IP packet and fill in the fields with known values. Assume that there is no data in the Echo request message, no IP fragmentation, and no errors in the IP header checksum.

4. (Augment DV routing with path MTU finding) Recall that in a distance-vector routing protocol, a router R computes its shortest path to a destination D by taking the minimum of the cost to D through each neighbor. The cost through a neighbor S is just R's cost to D plus the cost from R to S. Now a COMP5311 student suggests to augment the distance vector (a vector of destinations and costs) with the minimum MTUs to the destinations. That is, the entry in the distance vector sent by S consists of:

   - Destination address D
   - Minimum cost for S to reach D, and
   - Minimum MTU for this shortest-path route.

   With this enhanced distance vector, design a simple algorithm for R to advertise the minimum MTU to its neighbors, such that all routers will also obtain the path MTUs for the best paths to all destinations after the routing protocol has converged. You may consider two cases: (1) advertising S's directly connected networks to other routers and (2) learning routes to other networks from R's neighbor routers and further advertising them.

5. (IP tunnel and fragmentation) Recall from assignment one the tunnel fragmentation problem depicted in Figure 3, where the path MTU between LAN A and LAN D is given by $\min\{MTU1, MTU2-20, MTU3-20, PMTU_{2,3}-40, MTU4\}$. The MTUs for all the network links are equal to 1500 bytes; therefore, $MTU1 = MTU2 = MTU3 = MTU4 = PMTU_{2,3} = 1500$ bytes. Consider part (b) again: A host on LAN A sends an IP datagram to a host on LAN D with a total length of 1500 bytes. As discussed, $R1$ will first fragment the tunneled packet once, and then $R2$ will fragment the first fragment after tunneling. Therefore, the original packet has been fragmented into three IP packets in the network between $R2$ and $R3$. We label the three IP fragments by 1, 2, and 3 according to their order of transmissions (fragment 1 being transmitted first). Answer the following questions.

   (a) (2 marks) How many IP header(s) does each fragment have?

   (b) (2 marks) What are the fragment offset values in the three fragments?

   (c) (2 marks) What are the M bits (on or off) in the three fragments?

   (d) (2 marks) If fragment 3 is dropped by a router somewhere between $R1$ and $R4$ due to "TTL exceeded", an ICMP error message will be sent by the router that drops the packet. Who will receive the ICMP message and why?

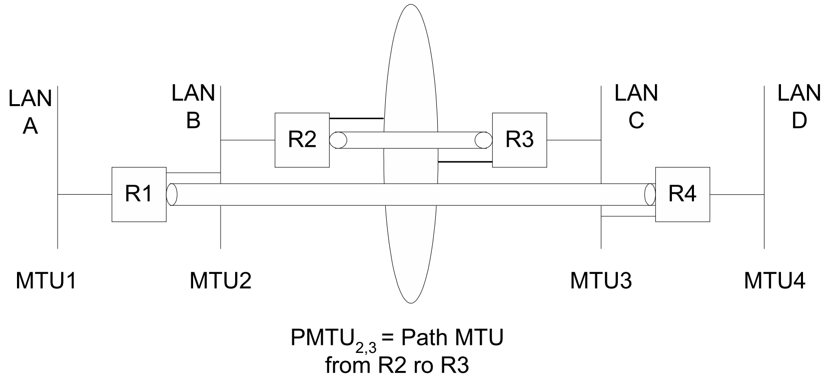

Figure 3: A nested IP tunnel scenario.

6. (TCP delayed ACK) Recall that a TCP receiver may delay sending an ACK when a data segment arrives. One of the delayed ACK strategy is to send an ACK for every second full-sized segment. However, due to implementation errors, a TCP receiver may not follow the ACK-every-other-segment strategy. Consider the following traces taken using tcpdump at receiver B. For example, in the first line, host A (using port 1000) sends a TCP packet to host B (using port 4000) at time 09:23:11.820187. The sequence number of this packet is 2049, and it carries 1448 bytes of TCP data. In the fourth line, B returns a pure ACK with acknowledgement number equal to 6393. The MSS used was 1460 bytes.

```
09:23:11.820187 A.1000 > B.4000: . 2049:3497(1448) ack 1
09:23:11.824147 A.1000 > B.4000: . 3497:4945(1448) ack 1
09:23:11.832034 A.1000 > B.4000: . 4945:6393(1448) ack 1
09:23:11.832222 B.4000 > A.1000: . ack 6393
09:23:11.934837 A.1000 > B.4000: . 6393:7841(1448) ack 1
09:23:11.942721 A.1000 > B.4000: . 7841:9289(1448) ack 1
09:23:11.950605 A.1000 > B.4000: . 9289:10737(1448) ack 1
09:23:11.950797 B.4000 > A.1000: . ack 10737
09:23:11.958488 A.1000 > B.4000: . 10737:12185(1448) ack 1
09:23:12.052330 A.1000 > B.4000: . 12185:13633(1448) ack 1
09:23:12.060216 A.1000 > B.4000: . 13633:15081(1448) ack 1
09:23:12.060405 B.4000 > A.1000: . ack 15081
```

(a) (2 marks) What is the ACK strategy used by B?

(b) (6 marks) It turned out that a bug in B's TCP implementation caused this problem. Specifically, the TCP implementation counts only the amount of TCP data received from A. When the total count reaches or exceeds 2MSS, B will send an ACK. However, in this case (not shown in the traces) A includes a TCP option which amounts to 12 bytes. Complete the remaining explanation for the problem.

7. (TCP congestion control) Similar to the TCP state transition diagram, the congestion control in TCP can be summarized in a state transition diagram shown in Figure 4. The three main states are slow start, fast recovery, and congestion avoidance. The values of `ssthresh` and `cwnd` are updated according to the state where the TCP sender is in. Initially, when the sender enters into the ESTABLISHED state, the values of `ssthresh` and `cwnd` are initialized to some system-dependent values (`ssthresh` = 65535 bytes and `cwnd` = MSS bytes in the figure). DUPACK stands for duplicate ACK. Fill in the missing items under (a)-(i) using `cwnd`, `ssthresh`, Flight-size, and MSS all of which count in bytes.

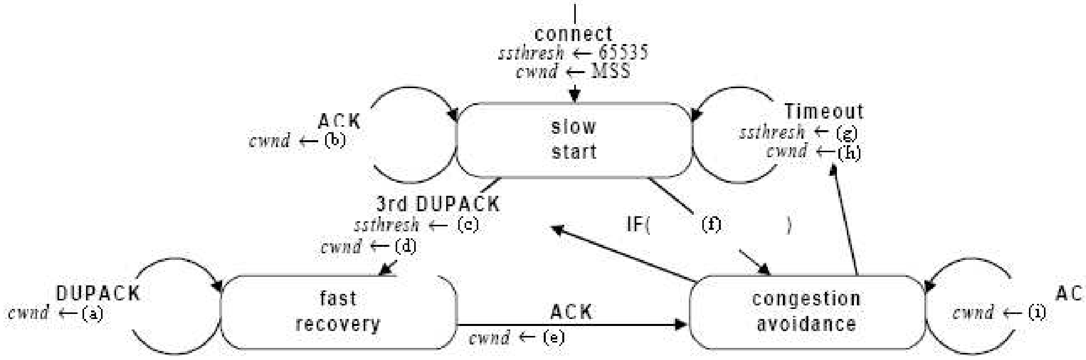

Figure 4: A state transition diagram for TCP congestion control.

8. (RIP with split horizon) Consider Figure 5 in which the three routers exchange their routes to an IP network denoted by *dest* using RIP-1. Split horizon with poisonous reverse is used and a distance of 16 is interpreted as destination unreachable. H's default router is set to R_1. How does H send a packet to *dest* for the following cases:

   (a) (2 marks) The distances announced by R_1, R_2, and R_3 are 5, 16, and 16, respectively.

   (b) (2 marks) The distances announced by R_1, R_2, and R_3 are 16, 16, and 5, respectively.

   (c) (2 marks) The distances announced by R_1, R_2, and R_3 are 16, 5, and 5, respectively.

   (d) (2 marks) The distances announced by R_1, R_2, and R_3 are 5, 5, and 16, respectively.

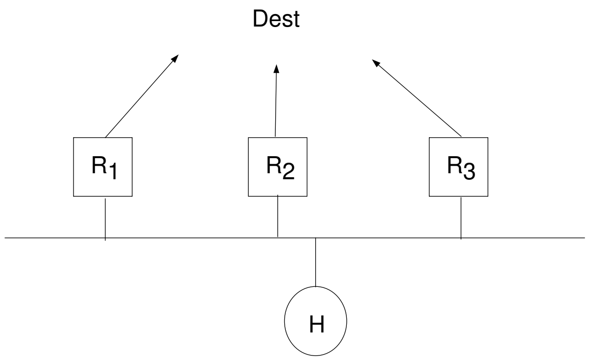

Figure 5: Three RIP-1 routers exchange routes for *dest* on the LAN.

9. (Reliable flooding) Consider the following implementation for the reliable flooding of link state packets (LSPs). A router R allocates $2n$ bits for each LSP, where n is the number of links connected to R. Half of the bits are *send bits*; if on, the ith send bit indicates that the LSP should be transmitted to the ith link. The other half are *ack bits*; if on, the ith bit indicates that an ack should be sent to the ith link. At most, one of the bits would be set for a given LSP and link.

   From time to time, R will scan the $2n$ bits for each LSP. If the send bit is found on, it will transmit the LSP on that link. If the ack bit is on, it will send an ack for that LSP on that link, and the ack bit is then set to off. Therefore, in this implementation, R does not transmit LSPs and acks immediately. Rather, it turns on and off the respective bits and then, with some time delay, processes them at the same time.

   Describe how R updates the send and ack bits in response to the following events. Note that there are other possible events that will update the bits.

   (a) (2 marks) Receiving a new LSP from ith link (the LSP is not in the link state database or the LSP carries a newer sequence number than the one in the database),

   (b) (3 marks) Receiving a duplicate LSP from ith link (the LSP has the same sequence number as the one in the database), and

   (c) (3 marks) Receiving an ack from ith link.

Section B: Please answer both questions in this section. Each question carries 22 marks.

10. (IP source route option) RFC 791 describes two IP source routing options: strict source route (SSR) and loose source route (LSR). The concept of the source route option is to let the originator (or source) of the datagram specify a list of points (routers) the datagram is to pass through on the way to its destination. The difference between SSR and LSR is that in SSR the route must specify router separated by only one network, but in LSR the path between stops on the specified route may be any length and determined by normal IP routing. The source route option includes a list of 32-bit IP addresses and a pointer that indicates which address in the list is to be used next.

    We use the following example of SSR to illustrate how the option works. Suppose that the source is host ISI-ARK with address `128.9.0.12` on ISI-NET. The destination is host DMC-CRC with address `128.43.0.1` on DRENET. The routers to be explicitly passed through are ISI-WB-GW, LL-GW, and CRC-GW. Each router has two addresses, one on each directly connected network.

```
+-----+
|     | 128.9.0.12
|IS-  |-------
|  ARK|     /ISI-NET
+-----+    /
          /
         /
        /     +-----+
       /      |ISI- | 28.45.0.0
       -------|  WB-|-------
   128.9.0.25 |   GW|     /WBNET
              +-----+    /
                        /
                       /    +-----+
                      /     |     | 10.5.0.10
                     -------|LL-GW|-------
                  28.19.0.0 |     |     /ARPANET
                            +-----+    /
                                      /
                                     /    +-----+
                                    /     |     | 128.43.1.1
                                   -------|CRC- |-------
                                10.1.0.15 |   GW|     /DRENET
                                          +-----+    /
                                                    /
                                                   /    +-----+
                                                  /     |     |
                                                 -------|DMC- |
                                             128.43.0.1 |  CRC|
                                                        +-----+
```

    When the datagram is sent from the source host (ISI-ARK) into the Internet, the source address (SA), destination address (DA), source route list (SRL), and source route pointer (SRP) fields are:

```
SA:  128.9.0.12
DA:  128.9.0.25
SRL: 28.19.0.0, 10.1.0.15, 128.43.0.1
SRP: 0  // no addresses have been processed so far.
```

After the datagram arrives at router ISI-WB-GW, it notices the source route option and moves the address from the next element of the SRL to the destination field, places its own reverse direction address in that element of the SRL, and increments the SRP. As the datagram leaves router ISI-WB-GW, the fields are:

```
SA:  128.9.0.12
DA:  28.19.0.0
SRL: 28.45.0.0, 10.1.0.15, 128.43.0.1
SRP: 1  // one address has been processed so far.
```

At router LL-GW, the processing is similar. The datagram leaves router LL-GW with its fields appearing:

```
SA:  128.9.0.12
DA:  10.1.0.15
SRL: 28.45.0.0, 10.5.0.10, 128.43.0.1
SRP: 2  // two addresses have been processed so far.
```

At router CRC-GW, the processing is similar. The datagram leaves router CRC-GW with its fields appearing:

```
SA:  128.9.0.12
DA:  128.43.0.1
SRL: 28.45.0.0, 10.5.0.10, 128.43.1.1
SRP: 3  // three addresses have been processed so far.
```

Finally, the datagram arrives at host DMC-CRC. Even though the source route option is still present, host DMC-CRC knows that it is the final destination, because the SRP now indicates that the SRL has been exhausted.

Using the LSR or SSR, an end-to-end IP forwarding therefore consists of a concatenation of IP forwarding, each of them has a different destination address.

(a) (2 marks) What is the first task in the processing of the source route option by a router, after checking the integrity of the option, if any?

(b) (2 marks) If a router's address is in the SRL, how can it determine its own reverse direction address to put into the SRL?

(c) (2 marks) If a router's address is in the SRL, does it need to decrement the TTL field and why?

(d) (2 marks) Does the IP header's checksum cover the source route option and why?

(e) (2 marks) If an IP packet with the source route option induces an ICMP error message from a router, will the source be able to receive this message and why?

(f) (2 marks) The implementation of the source route option include an one-byte option type code, an one-byte option length, an one-byte SRP, and the SRL. The SRL is composed of a series of IPv4 addresses. Any IP option will be padded to the 32-bit boundary.

How many bytes are needed to implement the IP source route option for the example above?

(g) (2 marks) Does the length of the IP packet change when only the source route option is used and why?

(h) (2 marks) If fragmentation occurs to an IP packet with the source route that will be processed next by R_n, how will the fragments be reassembled?

(i) (2 marks) If the IP payload is TCP packet, are the routers in the SRL required to update the TCP checksum?

(j) (2 marks) Note that the forwarding path to the destination is also recorded in the SRL. Therefore, the destination could also send an IP packet with source route option back to the sender using the recorded forwarding path. Give the SA, DA, SRL, and SRP for this packet sent back by DMC-CRC in the example above.

(k) (2 marks) If the LSR is used, are the router-level forwarding path (ISI-ARK to DMC-CRC) and the reverse path (DMC-CRC to ISI-ARK) identical?

11. (TCP partial ACKs) Consider a TCP connection between a sender and a receiver. Assume that all transmissions are perfect, i.e., lossless, in-order, etc. Furthermore, the sender is assumed to have the Nagle algorithm turned off and always has data to send. Therefore, the sender will immediately use the space freed up by an ACK to send a new data packet. The sender and receiver have also agreed on an MSS (in bytes) during the handshake. Moreover, the receiver always advertises a window size of one MSS, i.e., `RCV_WND` = 1 MSS. As a result, the sender's `SND_WND` is always 1 MSS, regardless of the `cwnd`, and he can send at most a TCP segment before receiving an ACK, which resembles a stop-and-wait protocol.

    Let the sequence number (SN) in the ith data packet S_i be s_i (i.e., s_i is the SN of the first byte of data) and the acknowledgment number (AN) in the ith ACK A_i be a_i. Note that the data size of S_i is given by $s_{i+1} - s_i$. The receiver in this case is not an ordinary TCP receiver which, for reasons unknown to you, may return a partial ACK. A *partial ACK* A_i is one for which $a_i < s_{i+1}$. A *full ACK* A_i, on the other hand, is one for which $a_i = s_{i+1}$. Moreover, all the ACKs are new: $a_0 < a_1 < a_2 < \ldots$. Figure 6 depicts the data and ACK transmission sequence in the SN space. The numbers inside `()` indicate the order of packet transmissions in time: S_0, A_0, S_1, A_1, $\ldots$. Note that all the ACKs in Figure 6 are full ACKs. For example, A_0 fully acknowledges S_0, A_1 fully acknowledges S_1, and so on.

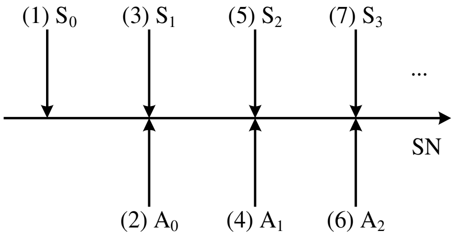

Figure 6: Stop-and-wait TCP data transmissions with full ACKs.

    (a) (4 marks) Consider another scenario in Figure 7 in which the receiver sends two partial ACKs to the sender: A_1 and A_2. What is the number of bytes that is left unacknowledged by A_1? Repeat the same question for A_2.

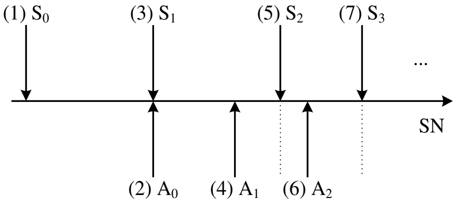

Figure 7: Stop-and-wait TCP data transmissions with two partial ACKs.

(b) (4 marks) Repeat the questions in part (a) for a new scenario in Figure 8.

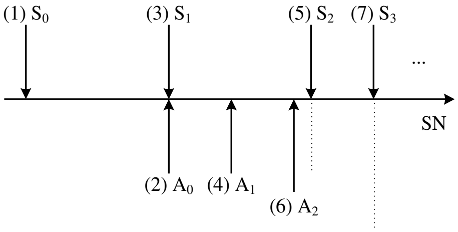

Figure 8: Another scenario for two partial ACKs.

(c) (3 marks) What is the relationship between `SND_NXT` and `SND_UNA` before receiving an ACK?

(d) (3 marks) When A_i (with AN = a_i) arrives, what is the relationship between `SND_NXT` and `SND_UNA` before sending a new segment?

(e) (4 marks) Note that after receiving data packet S_i, the receiver is able to determine the SN in S_{i+1} to be sent by the sender. What is the relationship between s_{i+1} and its state variable `RCV_NXT`?

(f) (4 marks) Let m_i be the number of bytes that is left unacknowledged by A_i. What is the acceptable range for m_i?

— End of the Examination Paper —