PRINCE OF SONGKLA UNIVERSITY

FACULTY OF ENGINEERING

**Final Examination:** Semester 1

**Academic Year:** 2009-2010

**Date:** October 2, 2009

**Time:** 09:00 – 12:00

**Subject Number:** 241-643

**Room:** R300

**Subject Title:** The Internet and its Protocols

---

Name: ______________________________

Student Number: ____________ Signature: ________________

---

**Exam Duration:** 3 hours

**This paper has 13 pages** (including this page).

**Authorised Materials:**

- Anything the student can carry, except for mobile phones and other communication devices.

**Instructions to Students:**

- *Answer questions in English.* Good English is **not** required.
- Attempt all 7 questions.
- Write the answers in the spaces provided in the examination paper.
- Anything illegible is incorrect.
- Answer briefly where possible, essays are **not** required.
- The marks allocated for each question are shown next to that question. There are 150 marks total for this examination. This will contribute 50% of the course total.

---

*For marker's use only.*

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | Total |
|---|---|---|---|---|---|---|---|
| | | | | | | | |

**Question 1.** *(12 marks)*

An Internet Protocol (IP) version 4 packet header contains a length field, defined as the total length of the IP packet (or fragment) of which the header is part.

An IP version 6 packet header contains a length field, defined as the length of the IP packet (or fragment) not including the size of the header.

Explain why this change was made when IPv6 was created.

**Question 2.** *(21 marks)*

List as many advantages as you can think of for using a grammar to define a protocol. Also list any disadvantages.

Does it make a difference what kind of protocol is being defined to the decision whether to use a grammar to help with the specification?

What alternative methods of protocol specification exist, and when might they be better choices than grammars?

**Question 3.** *(40 marks)*

A set of IPv4 packets are being sent to host S, with IP address 192.168.22.13. That is, the **Destination Address** field of all of the packet headers contains 192.168.22.13.

All of the packets are UDP packets (the **protocol** field in all IP packet headers contains the value **17** indicating UDP.)

All of the packets transmitted have a destination UDP port number of 1111, and host S has a server waiting for packets, from any sender, that are sent to UDP port 1111.

The **TOS**, **checksum**, and **TTL** fields of the IP header are all irrelevant to this question, though you may assume that the **TTL** in all arriving packets is greater than zero, and that all checksums are valid.

The **version** field of all packet headers is 4, and the **Header Length** field of all packet headers is 5 (indicating a header length of 20 octets, and no IP options present.) The **DF** and **RF** flag bits in the headers of all packets are both zero (always).

The content of other IP header fields of the set of packets that arrive at host S during a short interval are shown in the table on the following page. All values are shown in decimal, otherwise as they appear in the IP header.

Immediately before the first of those packets arrives at S, there are no packets waiting at S to be delivered to the application, and no fragments of earlier packets waiting to be reassembled before being delivered.

As soon as a packet is available at host S for delivery to the UDP transport layer, and then to the waiting application, that packet is delivered.

| | Source Address | Total Length | Packet Identifier | MF | Fragment Offset |
|---|---|---|---|---|---|
| 1 | 10.1.2.3 | 820 | 10234 | 0 | 0 |
| 2 | 201.2.3.4 | 1220 | 5555 | 1 | 0 |
| 3 | 10.1.2.3 | 640 | 20432 | 0 | 0 |
| 4 | 62.4.5.6 | 1500 | 5555 | 0 | 555 |
| 5 | 201.2.3.4 | 1220 | 5555 | 1 | 150 |
| 6 | 201.2.3.4 | 1220 | 5678 | 1 | 0 |
| 7 | 8.9.10.11 | 532 | 88 | 1 | 64 |
| 8 | 201.2.3.4 | 729 | 5555 | 0 | 300 |
| 9 | 201.2.3.4 | 420 | 5678 | 0 | 300 |
| 10 | 62.4.5.6 | 1500 | 5555 | 1 | 185 |
| 11 | 62.4.5.6 | 1500 | 5555 | 1 | 0 |
| 12 | 8.9.10.11 | 420 | 88 | 0 | 128 |
| 13 | 201.2.3.4 | 1220 | 5678 | 1 | 150 |
| 14 | 62.4.5.6 | 1500 | 5555 | 1 | 370 |
| 15 | 8.9.10.11 | 532 | 88 | 1 | 0 |
| 16 | 201.2.3.4 | 1220 | 5555 | 1 | 150 |
| 17 | 33.4.55.2 | 709 | 1235 | 0 | 0 |
| 18 | 8.9.10.11 | 120 | 315 | 0 | 300 |
| 19 | 201.2.3.4 | 1220 | 5555 | 1 | 300 |
| 20 | 8.9.10.11 | 1220 | 134 | 1 | 150 |
| 21 | 33.4.55.2 | 620 | 1236 | 1 | 0 |
| 22 | 62.4.5.6 | 1500 | 5556 | 1 | 0 |
| 23 | 62.4.5.6 | 1500 | 5556 | 1 | 185 |
| 24 | 33.4.55.2 | 109 | 1236 | 0 | 75 |
| 25 | 8.9.10.11 | 137 | 134 | 0 | 300 |
| 26 | 10.1.2.3 | 820 | 7654 | 0 | 0 |
| 27 | 8.9.10.11 | 1220 | 236 | 1 | 0 |
| 28 | 8.9.10.11 | 1220 | 315 | 1 | 0 |
| 29 | 8.9.10.11 | 980 | 236 | 0 | 150 |
| 30 | 62.4.5.6 | 1500 | 5556 | 1 | 370 |

A) In the table below, fill in the fields to show the details of the packets that are delivered from the IP layer to UDP, in the order that they are made available to UDP.

Note that there are more rows provided in the table than packets delivered to UDP, you can expect to have some number of empty rows.

You need only allow for packets delivered to UDP up to the time immediately after the last of the packets from the table on the previous page (incoming packets) has arrived.

You should assume that all of those packets arrive within a period much shorter than one second, that is, no adjustments to TTLs of waiting fragments or packets is required (and hence, for this question, the TTL header field is irrelevant.)

[25 marks]

| | Source Address | Total Length | Packet Identifier | MF | Fragment Offset |
|---|---|---|---|---|---|
| 1 | | | | | |
| 2 | | | | | |
| 3 | | | | | |
| 4 | | | | | |
| 5 | | | | | |
| 6 | | | | | |
| 7 | | | | | |
| 8 | | | | | |
| 9 | | | | | |
| 10 | | | | | |
| 11 | | | | | |
| 12 | | | | | |
| 13 | | | | | |
| 14 | | | | | |
| 15 | | | | | |
| 16 | | | | | |

B) After the packets indicated in your answer to the previous part of this question have been delivered, there will be a number of partially reassembled packets waiting on queues in the IP layer of host S.

Show those queues, including information about the packets (fragments) waiting upon them.

For each queue, show the IP source address and Packet Identifier, (the protocol for all packets here is UDP, so you may omit that, and the Destination Address is 192.168.22.13 so that may also be omitted.) Then for each packet on that (each) queue, show the Total Length, MF, and Fragment Offset fields.

[15 marks]

**Question 4.** *(30 marks)*

In the previous question, all packets were UDP packets. The question was written that way to be more realistic, as TCP packets rarely need to be fragmented, whereas for UDP packets fragmentation is sometimes unavoidable.

A) Explain the difference between TCP and UDP that leads to that result.

[6 marks]

B) Why do TCP implementations prefer to avoid fragmentation?

[6 marks]

C) What mechanism, or mechanisms, exist that allow TCP (and some UDP applications) to avoid fragmentation by only sending packets small enough that fragmentation is not needed.

[6 marks]

D) Explain how that mechanism, or those mechanisms or protocols, work to allow a packet sender to determine a suitable packet size.

[12 marks]

**Question 5.** *(12 marks)*

What are some of the advantages of application protocols where all of the data is encoded as normal (readable) text?

What are some disadvantages?

**Question 6.** *(20 marks)*

A) For TCP that is not using window scaling (that is: the window scale option is not present) complete the missing entries in the table to show the (approximate) maximum possible TCP throughput (in bits/second) for each of the combinations of Round Trip Time (RTT) measured in milliseconds, and network bandwidth, measured in bits/second:

[12 marks]

| | | Round Trip Time *(ms)* | | | |
|---|---|---|---|---|---|
| | | 0.2 | 2.0 | 10.0 | 500.0 |
| Bandwidth *bits/sec* | 256000 | | | | |
| | 1000000 | | | | |
| | 10000000 | | | | |
| | 100000000 | | | | |
| | 1000000000 | | | | |
| | 10000000000 | | | | |

B) Indicate **for which of the above combinations** window scaling would assist in improving TCP's throughput, and in each case where window scaling would assist, **which value of the window scale option** would achieve best results. You can assume that there will not be a problem with memory for TCP buffers that might otherwise limit the maximum window size.

[8 marks]

**Question 7.** *(15 marks)*

Which do you believe is more important when designing a protocol, efficiency, or extensibility?

Why?

Give reasons for your opinion, including examples from protocols that support your argument (which can be cases where a positive result was achieved from following the advice you would give, or cases where a poor result was achieved after adopting the other approach).