Networks and Protocols
Jacobs University Bremen
Dr. Jürgen Schönwälder

Course: 320301
Date: 2008-12-17
Duration: 120 minutes

Final Examination

The Jacobs University's Code of Academic Integrity applies to this examination. Please fill in your name (please write readable) and sign below.

Name:

Signature:

This exam is **closed book**. In addition, you are not allowed to use any electronic equipment such as computers, smart phones, cell phones, or calculators.

Please answer the questions on the problem sheets. If you need more space, feel free to write on the back of the pages. Please keep the papers stapled.

Problem	**Max. Points**	**Points**	**Grader**
F.1	10		
F.2	20		
F.3	14		
F.4	20		
F.5	14		
F.6	12		
F.7	10		
Total	100		

Good luck and have a nice and relaxing winter break!

/js

Problem F.1: *ip fragmentation* (2+2+2+2+2 = 10 points)

Indicate which of the following statements are correct or incorrect by marking the appropriate boxes. For every correctly marked box, you will earn two points. For every incorrectly marked box, you will loose one point. Statements which are not marked or which are marked as true and false will be ignored. The minimum number of points you can achieve is zero.

true	false	
□	□	Fragmented IP packets may be reassembled by routers if the link to the next hop can carry the reassembled packet.
□	□	The IPv4 fragment offset and identification header fields in combination with the source and destination IP addresses uniquely identify an IP fragment.
□	□	Each IP fragment carries its own independent checksum.
□	□	The loss of an IP fragment may cause transport protocols to resent the whole IP packet and the likelihood to deliver the IP packet goes down quickly as network congestion increases.
□	□	IPv6 avoids fragmentation by requiring a larger MTU size and by notifying the sender of an IP packet in case a packet needs fragmentation.

Solution:

true	false	
□	☒	Fragmented IP packets may be reassembled by routers if the link to the next hop can carry the reassembled packet.
☒	□	The IPv4 fragment offset and identification header fields in combination with the source and destination IP addresses uniquely identify an IP fragment.
☒	□	Each IP fragment carries its own independent checksum.
☒	□	The loss of an IP fragment may cause transport protocols to resent the whole IP packet and the likelihood to deliver the IP packet goes down quickly as network congestion increases.
☒	□	IPv6 avoids fragmentation by requiring a larger MTU size and by notifying the sender of an IP packet in case a packet needs fragmentation.

Problem F.2: *tcp/ip packet decoding* (4+4+4+4+4 = 20 points)

Below are the first 60 bytes (printed in hexadecimal) of a TCP/IP packet captured on an IEEE 802.3 link (excluding the preamble). On the last pages of this exam are packet header templates. The Ethernet frame type for IPv4 is 0x0800 and 0x86dd for IPv6. The transport IPv4 protocol field / IPv6 next header field is 0x11 for UDP and 0x06 for TCP. The IP header does not use any options.

```
0015 1723 4a36 0017 f2d0 4c82 0800 4500
0098 10bb 4000 4006 9523 0a32 e785 40aa
6220 c00c 0050 9233 5122 4549 89d6 8018
ffff 950c 0000 0101 080a 3ff9 fe91 2d78
5701 4745 5420 2f20 4854 5450 2f31 2e30
0d0a 5573 6572 2d41 6765 6e74 3a20 5767
6574 2f31 2e31 312e 340d 0a41 6363 6570
743a 202a 2f2a 0d0a 486f 7374 3a20 7777
772e 6965 7466 2e6f 7267 0d0a 436f 6e6e
6563 7469 6f6e 3a20 4b65 6570 2d41 6c69
7665 0d0a 0d0a
```

Decode the packet for the following fields:

a) Ethernet MAC source and destination addresses (hexadecimal)

b) IP source and destination addresses (in standard notation)

c) TCP source and destination port numbers (decimal)

d) Flags of TCP segment (e.g., SYN, FIN, ACK, etc.)

e) Application layer protocol (so far this can be determined)

Solution:

a)	Source MAC address:	`00:15:17:23:4a:36`
	Destination MAC address:	`00:17:f2:d0:4c:82`
b)	IP source address:	`10.50.231.133` (0a 32 e7 85)
	IP destination address:	`64.170.98.32` (40 aa 62 20)
c)	TCP source port:	`49164` (0xc00c)
	TCP destination port:	`80` (0x0050)
d)	Flags of the TCP segment:	PSH, ACK (0x18)
e)	Application layer protocol:	HTTP/1.0 GET request to www.ietf.org

Problem F.3: *dns and smtp* (4+2+6+2 = 14 points)

Below is a short transcript of a Unix terminal session:

```
$ mail l.page@google.com
Subject: internship application

Dear Larry Page,

I have been using Google for years and I love to do an internship in
your great company.
.
$
```

The system is connected directly to the Internet and assume that there are no middleboxes. Explain what happens next on the system.

a) Assuming that the local DNS cache does not have the right information, which DNS records must be retrieved? How many DNS messages are needed for retrieving these records in the best case?

b) The Unix system tries to ship the message to the mail server handling emails for `google.com` but unfortunately the TCP connection underlying the SMTP dialog breaks before a proper TCP teardown procedure was executed. Does the server have to retry shipping the message in all cases? Justify your answer.

c) Briefly describe the difference between the envelope, the header and the body of an Internet mail message. Which elements of the shown transcript will become part of the envelope, the header, and the body?

d) SMTP uses 3-digit reply codes and this has been carried over to several other protocols. Explain why 3-digit reply codes are so popular and why protocol designers not simply enumerate the reply codes counting from zero or one.

Solution:

a) To deliver the email, the system needs the MX record for `google.com` to find the mail servers responsible for handling and at least the A or AAAA records for the mail server with the highest priority. In the best case, a single query and response message are sufficient if the DNS server provides all the needed information in the response.

b) Yes, unless the TCP connection breaks after the final positive reply code for the `DATA` command has been received by the client, which resets the SMTP state.

c) The envelope controls the delivery of a mail message. In the example, `l.page@google.com` becomes part of the envelope. The header contains machine readable structured information about the message and is followed by the body which contains the message contents. In the example, `Subject:  internship application` becomes part of the header while the body consists of the remaining text starting with `Dear Larry Page,`.

d) The 3 digits express a hierarchy. The first digit indicates whether a reply is positive or negative and whether it is preliminary, transient, or permanent. The second digit identifies a specific category of replies while the last digit identifies a specific reason of the category. The benefit of this scheme is that implementation can react reasonably even in case a new server returns an error code where for example the meaning of the last digit is unknown to the client.

Problem F.4: *tcp congestion control on high bandwidth delay paths* (4+6+6+4 = 20 points)

TCP congestion control was designed at a time where the data rate of networks was orders of magnitude lower than what we have available today. In this problem, you will evaluate how TCP behaves on networks with high data rates (bandwidth) and large delays.

a) Briefly describe the two TCP congestion control modes *slow start* and *congestion avoidance*. How does the congestion window changes in each mode during a round-trip interval? Which events cause a change of the congestion mode of a TCP connection?

b) Consider a 1 Gbps path carrying $b = 1250\,byte$ segments with $r = 100\,ms$ round-trip delay. How long does it take to saturate the 1 Gbps path during *slow start* if the initial window size is one segment and the threshold is arbitrarily large? Derive a general formula for the bandwidth used in the i-th round-trip interval during *slow start* and use this formula to answer the question for the given scenario.

c) Consider the same 1 Gbps path carrying $b = 1250\,byte$ segments with $r = 100\,ms$ round-trip delay. During congestion avoidance, how long does it take for a full cycle, i.e, from halving the congestion window until the capacity W is exceeded again? Derive a general formula and then use this formula to answer the question for the given scenario.

d) TCP is known to guarantee fairness. Explain what TCP fairness means. Given the 1 Gbps path scenario carrying $b = 1250\,byte$ segments with $r = 100\,ms$ round-trip delay, will it help to increase efficiency if an application protocol uses multiple TCP connections instead of one? A qualitative answer is sufficient; you do not have to provide formulas or calculate numbers.

Solution:

a) Slow start mode:

   - The slow start mode is used initially to quickly open the congestion window of a TCP sender.
   - Send two TCP segments are sent in response to each ACK that advances the sender's congestion window.
   - Exponential increase of the sending rate until a threshold is reached.

   Congestion avoidance mode:

   - The congestion avoidance mode is used to probe slowly but continously the available bandwidth.
   - Send an additional segment of data for each loss-free round-trip time interval.
   - Linear increase of the sending rate until a congestion event (duplicate ACKs, timeout, ECN) occurs

b) We consider segments of size b bits and a round-trip delay of r seconds. During the first round-trip interval, a single segment is exchanged and the data rate is b/r. In the second round, two segments are exchanged and the data rate is $2b/r$. In the third round, four segments are exchanged and the data rate is $4b/r$. In general, in round i with $i \in \{1, 2, \ldots\}$, the number of segments exchanged is $2^{(i-1)}$ and the data rate is $2^{(i-1)}b/r$.

   With $b = 10000$ bits and $r = 0.1$ seconds, we need to find the smallest i such that:

   $$2^{(i-1)}\frac{b}{r} = 2^{(i-1)}\frac{10000}{0.1} = 2^{(i-1)}100000 > 1000000000$$

   This is equivalent to finding the smallest i such that:

   $$2^{(i-1)} > 10000$$

   Since $2^9 = 512$ and $2^{10} = 1024$, after 11 round-trips or 1.1 seconds, the 1 Gbps path is saturated.

c) During congestion avoidance, the TCP performance cycles from $W/2$ to $W\ bps$ with a linear increase of b/r. After a congestion event and halving the congestion window to $W/2$, the after i round-trips is $W/2 + ib/r$. We are looking for the smallest i such that:

$$\frac{W}{2} + i\frac{b}{r} > W$$

Solving the equation for i gives us:

$$i > \frac{W}{2}\frac{r}{b} = \frac{1000000000}{2}\frac{0.1}{10000} = 5000$$

One cycle therefore takes 5000 round-trips or approximately 8 minutes and 33 seconds.

d) TCP fairness means that N connections sharing the same bottleneck link will get $1/N$ of the link capacity. Using multiple TCP connections causes higher utilization of the 1 Gbps path since the individual TCP connections will experience a link of 1/N Gpbs and thus the congestion avoidance probing cycles become shorter.

Problem F.5: *network address translation and file transfer protocol* (2+6+2+2+2 = 14 points)

Assume that an internal network of a university consists of the private subnet 192.168.0.0/24 and the university has a pool of five global IP addresses 212.201.49.1-5.

a) Explain the difference between a basic Network Address Translator (NAT) and a Network Address Port Translator (NAPT).

b) A port restricted cone network address translator preserving port numbers is used. The following events happen (in that order):

1. host 192.168.0.2:1234 retrieves a web page from the web server at 42.35.12.33
2. host 192.168.0.5:1234 retrieves a web page from the web server at 100.12.45.11
3. host 192.168.0.2:1234 retrieves a web page from the web server at 100.12.45.11

Write down the source and destination transport addresses used by the request and response packets on the internal and on the external network by filling out the following table.

event	internal		external	
packets	src ip:port	dst ip:port	src ip:port	dst ip:port
1.req				
1.res				
2.req				
2.res				
3.req				
3.res				

Assume that global IP addresses are assigned in increasing order by the NAT, starting with the lowest host number first.

c) The File Transfer Protocol (FTP) supports an active mode and a passive mode. Explain briefly how the two modes work.

d) Describe a scenario where the active mode of FTP fails due to network address translators.

e) Are there scenarios where passive mode might not work?

Solution:

a) A basic Network Address Translator (NAT) translates private IP addresses into public IP addresses. It does not touch transport layer port numbers. A Network Address Port Translator (NAPT) translates transport endpoint identifiers and it allows to share a single public address among many private addresses (masquerading).

b) A possible transport address binding looks as follows:

event	internal		external	
packets	src ip:port	dst ip:port	src ip:port	dst ip:port
1.req	192.168.0.2:1234	43.35.12.33:80	212.201.49.1:1234	43.35.12.33:80
1.res	43.35.12.33:80	192.168.0.2:1234	43.35.12.33:80	212.201.49.1:1234
2.req	192.168.0.5:1234	100.12.45.11:80	212.201.49.2:1234	100.12.45.11:80
2.res	100.12.45.11:80	192.168.0.5:1234	100.12.45.11:80	212.201.49.2:1234
3.req	192.168.0.2:1234	100.12.45.11:80	212.201.49.1:1234	100.12.45.11:80
3.res	100.12.45.11:80	192.168.0.2:1234	100.12.45.11:80	212.201.49.1:1234

c) In active mode, the client opens a listening TCP endpoint and the server actively initiates the data transfer connections. In passive mode, the server opens a listening TCP endpoint and the client actively initiates the data transfer connections.

d) Active mode fails if the client is behind a NAT because the NAT does not allow the server to establish a TCP connection to the client.

e) Passive mode can fail in scenarios where the server chooses a port number that is filtered by firewalls. Otherwise, passive FTP is usually fine. Passive FTP would fail if the server is behind a NAT but then it would be difficult for a client to establish a control connection in the first place.

Problem F.6: *secure electronic mail* (12 points)

Compare the technologies PGP, S/MIME and DKIM that were designed to secure Internet mail.

a) Who (sender MUA, sender MTA, receiver MTA, receiver MUA) is responsible to sign mail messages?

b) Who (sender MUA, sender MTA, receiver MTA, receiver MUA) is responsible to verify mail messages?

c) Which part of a mail message (header, body) carries the signature?

d) Which technologies provide privacy protection (encryption) of mail messages?

For each of the three technologies PGP, S/MIME, and DKIM, answer the questions by filling the table below with the correct answers.

Protocol	Signer (a)	Verifier (b)	Location (c)	Encryption (d)
PGP				
S/MIME				
DKIM				

Solution:

Protocol	Signer (a)	Verifier (b)	Location (c)	Encryption (d)
PGP	sender MUA	receiver MUA	body	yes
S/MIME	sender MUA	receiver MUA	body	yes
DKIM	sender MTA	receiver MTA	header	no

Problem F.7: *hypertext transfer protocol* (2+2+2+2+2 = 10 points)

Indicate which of the following statements are correct or incorrect by marking the appropriate boxes. For every correctly marked box, you will earn two points. For every incorrectly marked box, you will loose one point. Statements which are not marked or which are marked as true and false will be ignored. The minimum number of points you can achieve is zero.

true false

□ □ An HTTP client can retrieve the home pages of two different web sites hosted on the same server over a single persistent connection.

□ □ HTTP pipelining requires a persistent connection.

□ □ An HTTP client can request the transparent compression of the documents retrieved from a server.

□ □ HTTP supports chunked encoding to improve the efficiency of pipelining.

□ □ An HTTP client can send a range request to retrieve a part of a long document.

Solution:

true false

☒ □ An HTTP client can retrieve the home pages of two different web sites hosted on the same server over a single persistent connection.

☒ □ HTTP pipelining requires a persistent connection.

☒ □ An HTTP client can request the transparent compression of the documents retrieved from a server.

□ ☒ HTTP supports chunked encoding to improve the efficiency of pipelining.

☒ □ An HTTP client can send a range request to retrieve a part of a long document.

Ethernet Frame Format

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            Preamble                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   Destination MAC Address                     :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
:                               |                               :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
:                        Source MAC Address                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       Type / Length           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

IPv4 Header Format

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Version|  IHL  |Type of Service|          Total Length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Identification        |Flags|      Fragment Offset    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Time to Live |    Protocol   |         Header Checksum       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Source Address                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Destination Address                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Options                    |    Padding    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

IPv6 Header Format

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Version| Traffic Class |           Flow Label                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Payload Length        |  Next Header  |   Hop Limit   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               :
+                                                               +
:                                                               :
+                         Source Address                        +
:                                                               :
+                                                               +
:                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               :
+                                                               +
:                                                               :
+                      Destination Address                      +
:                                                               :
+                                                               +
:                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

UDP Header Format

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Source Port          |      Destination Port         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Length             |          Checksum             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

TCP Header Format

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Source Port          |       Destination Port        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Sequence Number                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Acknowledgment Number                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Offset| Reserved  |U|A|P|R|S|F|            Window             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Checksum            |         Urgent Pointer        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Options                    |    Padding    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Hexadecimal ASCII Character Set

00 nul	01 soh	02 stx	03 etx	04 eot	05 enq	06 ack	07 bel
08 bs	09 ht	0a nl	0b vt	0c np	0d cr	0e so	0f si
10 dle	11 dc1	12 dc2	13 dc3	14 dc4	15 nak	16 syn	17 etb
18 can	19 em	1a sub	1b esc	1c fs	1d gs	1e rs	1f us
20 sp	21 !	22 ‘	23 #	24 $	25 %	26 &	27 ’
28 (	29)	2a *	2b +	2c ,	2d -	2e .	2f /
30 0	31 1	32 2	33 3	34 4	35 5	36 6	37 7
38 8	39 9	3a :	3b ;	3c <	3d =	3e >	3f ?
40 @	41 A	42 B	43 C	44 D	45 E	46 F	47 G
48 H	49 I	4a J	4b K	4c L	4d M	4e N	4f O
50 P	51 Q	52 R	53 S	54 T	55 U	56 V	57 W
58 X	59 Y	5a Z	5b [	5c \	5d]	5e ˆ	5f _
60 ‘	61 a	62 b	63 c	64 d	65 e	66 f	67 g
68 h	69 i	6a j	6b k	6c l	6d m	6e n	6f o
70 p	71 q	72 r	73 s	74 t	75 u	76 v	77 w
78 x	79 y	7a z	7b {	7c |	7d }	7e ˜	7f del