EKSAMEN / EXAM	TTM4100	18	05	2004

LES REGLENE FØR DU STARTER!
READ THE RULES BEFORE YOU START!

Skriv studentnummeret ditt her / Write your student's number here: | | | | | |

Gjenta studentnummeret her / Repeat student's number here: | | | | | |

	Sant True	Feil False
1.1.1	☒	☐
1.1.2	☒	☐
1.1.3	☐	☒
1.1.4	☒	☐
1.1.5	☐	☒
1.1.6	☒	☐
1.1.7	☒	☐

	Sant True	Feil False
1.2.1	☒	☐
1.2.2	☐	☒
1.2.3	☐	☒
1.2.4	☐	☒
1.2.5	☐	☒
1.2.6	☒	☐
1.2.7	☐	☒

	Sant True	Feil False
1.3.1	☐	☒
1.3.2	☐	☒
1.3.3	☐	☒
1.3.4	☐	☒
1.3.5	☐	☒
1.3.6	☐	☒
1.3.7	☒	☐

1.4	N-con. req	N-con. cnf	N-data. req()	N-data. ind()	N-dis. req	N-dis. cnf	CR-NPDU	CC-NPDU	DT-NPDU()	DR-NPDU
A	☐	☐	☐	☐	☐	☐	☐	☐	☒	☐
B	☐	☒	☐	☐	☐	☐	☐	☐	☐	☐
C	☒	☐	☐	☐	☐	☐	☐	☐	☐	☐
D	☐	☐	☐	☐	☐	☒	☐	☐	☐	☐
P	☐	☐	☐	☒	☐	☐	☐	☐	☐	☐
Q	☐	☐	☐	☐	☐	☐	☐	☐	☐	☒
R	☐	☐	☐	☐	☐	☐	☒	☐	☐	☐
X	☐	☐	☐	☐	☐	☐	☐	☐	☒	☐
Y	☐	☐	☐	☐	☐	☐	☐	☒	☐	☐

	A	B	C	D	E
2.1	☐	☐	☐	☐	☒
2.2	☐	☐	☐	☒	☐
2.3	☐	☐	☐	☐	☒

	Sant True	Feil False
3.1.1	☒	☐
3.1.2	☐	☒
3.1.3	☒	☐
3.1.4	☐	☒
3.1.5	☒	☐
3.1.6	☒	☐

	A	B	C	D	E
3.2	☐	☐	☐	☒	☐
3.3	☐	☐	☐	☐	☒

	Sant True	Feil False
3.4.1	☒	☐
3.4.2	☐	☒
3.4.3	☒	☒
3.4.4	☐	☒

	Sant True	Feil False
3.4.5	☐	☒
3.4.6	☐	☒
3.4.7	☒	☐

Lykke til! Good luck!

EKSAMEN / EXAM	TTM4100	18	05	2004

Skriv studentnummeret ditt her ⇨⇨
Write your student's number here ⇨

Gjenta studentnummeret her ⇨⇨
Repeat student's number here ⇨

	Sant / True	Feil / False
4.1.1		X
4.1.2		X
4.1.3	X	
4.1.4	X	
4.1.5	X	
4.1.6	X	
4.1.7		X

	A	B	C	D	E
4.2				X	
4.3			X		
4.4			X		

	Sant / True	Feil / False
5.1.1		X
5.1.2	X	
5.1.3	X	
5.1.4	X	
5.1.5		X
5.1.6		X
5.1.7		X

	A	B	C	D	E
5.2			X		

6.1

	IDLE	WAITING	QUEUED	ESTABLISHED	SENDING	RECEIVING	DISCON-NECTING
A	X						
B		X					
C				X			
D					X		
E				X			
F							X
G	X						
H	X						
I			X				
J				X			
K						X	
L				X			
M				X			
N							X
O	X						

6.2

	LISTEN	CONNECT	DISCON-NECT	SEND	RECEIVE
a		X			
b				X	
c			X		
d	X				
e					X
f			X		

Lykke til! *Good luck!*

EKSAMEN / EXAM	TTM4100	18	05	2004

Skriv studentnummeret ditt her ⇨⇨
Write your student's number here ⇨

Gjenta studentnummeret her ⇨⇨
Repeat student's number here ⇨

	Sant / True	Feil / False
7.1.1	☐	☒
7.1.2	☐	☒
7.1.3	☐	☒
7.1.4	☒	☐
7.1.5	☐	☒
7.1.6	☒	☐

	Sant / True	Feil / False
7.2.1	☒	☐
7.2.2	☒	☐
7.2.3	☐	☒
7.2.4	☒	☐

	Sant / True	Feil / False
7.3.1	☐	☒
7.3.2	☐	☒
7.3.3	☒	☐
7.3.4	☒	☐
7.3.5	☐	☒
7.3.6	☐	☒

	Sant / True	Feil / False
7.4.1	☒	☐
7.4.2	☒	☐
7.4.3	☒	☐
7.4.4	☐	☒
7.4.5	☐	☒
7.4.6	☐	☒
7.4.7	☐	☒
7.4.8	☐	☒

	Sant / True	Feil / False
7.5.1	☒	☐
7.5.2	☒	☐
7.5.3	☐	☒
7.5.4	☐	☒
7.5.5	☒	☐
7.5.6	☐	☒
7.5.7	☒	☐
7.5.8	☐	☒

7.6

Answer proposal is included in the Comments part at the end of this document.

IKKE SKRIV HER – SKRIV IKKJE HER ⇨
DO NOT WRITE HERE ⇨

Lykke til! *Good luck!*

EKSAMEN / EXAM	TTM4100	18	05	2004

Skriv studentnummeret ditt her ⇨⇨
Write your student's number here ⇨

Gjenta studentnummeret her ⇨⇨
Repeat student's number here ⇨

7.7

Answer proposal is included in the Comments part at the end of this document.

Lykke til! *Good luck!*

IKKE SKRIV HER – SKRIV IKKJE HER ⇨
DO NOT WRITE HERE ⇨

Lykke til! *Good luck!*

Comments to the proposed solution

Ad 1.1.

The calculations applied are the same as in Compulsory Assignment 1

A component is a Host or a Node.
M = 32768 bytes
d = flying time on satellite link (= speed of light delay) = 0.3 sec
C = transmission capacity (=64000 bytes/sec)
Setuptime = time for connection establishment
X = Maximum length of packet excluding header (both DG and VC) = 4096 bytes
Hdg = The header length for Datagram = 80 bytes.
Hvc= The header length VC = 4 Bytes.
Xsg = Length of signaling packet = 80 bytes. (This assumes a data part of the signaling packet = 0 bytes as in Assignment 1. Using a data part of the signaling packet = 80 bytes will not change the results related to the questions 1.1)

Setuptime = 2 ((1+2)* Xsg /C +d) = 0.6075 Sec

<u>Packet switching based on Datagram</u>

The total message is split in 8 packets. The packets are pipelined, but the total transmission time must be added by two packet transmissions because there are two intermediate Nodes (See Tanenbaum Figure 2.39). Or the first packet must be read into a Component 3 times, but the following 7 packets are closely pipelined (3+7=10).

Tdg = (8+2) (X+Hdg)/C + d = 10*(4096+80)/64000+0.3 = 0.9525 sec

<u>Packet switching based on VC</u>

The total message is split in 8 packets as for DG and the pipelining is similar. The Header length Hvc is different and there is also a connection procedure. A signaling packet is sent as a datagram back and forth.

Tvc = (8+2) (X+Hvc)/C+ d + Setuptime = 10*(4096+4)/64000+ 0.3 + 0.6075 = 1.548 sec

<u>Circuit switching</u>

Setuptime is as for VC.
Tcs= M/C + d + Setuptime = 32768/64000 + 0.3 + 0.6075 = 1.4195 sec

Ad 1.4.

The proposed solution is a variant of Assignment 1. Assignment 4-5 is also related to the same system as in Figure 1-1 in the Exam.

Ad 2.1.

The only valid answer among the given five choices is E. After the Exam and the NEWS discussion this question has been independently discussed with 3 members of the Signal Processing Group at Department of Electronics and Telecommunications. Nyquist can be used if you know the modulation, but without knowing the modulation Nyquist cannot determine the max bit rate.

Ad 3.1.5
The context of the Exam is the Tanenbaum book and the context of 3.1.5 is Link Layer/ MAC Layer part. The only type of division used in the Tanenbaum book is based on modulo- 2 arithmetics. The slides for the Link Layer shows clearly the usage of modulo-2 arithmetics in the calculation of the CRC code. Assignment 8 provides a similar example of this kind of division.

Ad 3.4.3
It has been decided to accept both True and False answers. The question should have been related to a network based on CSMA/CD not Ethernet.

Ad. 3.4.6
All the topics of Wireless LAN, MACA, and error correction have been covered by the syllabus of the course (in the book p 267-279, p 281-292, and p 292-338).

Ad 4.1.5
After a discussion with different faculty members the conclusion is to only accept the True answer. The question was in the Network Layer part of the exam, and was about a machine that is connected to several different networks. In such a case there is a different IP address for each Network connection or Network interface used. In IP each interface has its own IP address. A terminal with more than one network connection will therefore have more than one IP address. This can also be seen from DNS where it is possible to associate more than one address to the same logical name. It is however common to use one address as the main address in the routing protocol.

Ad 6.1 and 6.2
A diagram for a simplified version of TCP is presented, and a given instance of the behavior shows a Message Sequence Chart. A basis for this problem was that the sequence in Figure 6-2 has taken place. Figure 6-2 shows a disconnection procedure based on 3-way handshake. Figure 6-1 allows for both 2-way and 3-way handshake. The interpretation of the Figure 6-1 given in Figure 6-21 in the Tanebaum book is a 2-way handshake disconnection procedure, while the real TCP behavior in Figure 6-33 in the book is only 3-way handshake.

Ad 7.1.6
After discussion with the sensor, it has been decided to only accept the True answer to this question. The following Explanation can be found in the "HTTP made really easy" page 15, which is part of the syllabus:
"Supporting HTTP 1.0 Requests
To be compatible with older browsers, HTTP 1.1 servers must support HTTP 1.0 requests. In particular, when a request uses HTTP 1.0 (as identified in the initial request line),
* don't require the Host: header, and
* don't send the "100 Continue" response."

Ad 7.6
This very particular question can be found in the Tanenbaum book, Exercise 40 on page 718. The question is part of the Application Layer and is not only related to the project assignment. The answer given below is the answer given by Tanebaum. Students that have submitted answers different from the Tanenbaum answer, but demonstrate understanding will get score. The Tanenbaum answer is as follows:

One kind of effectiveness is regarding the "hit rate". Most likely HTML pages change more often than JPEG files. Lots of sites fiddle with their HTML all the time, but do not change the images much.

But the effectiveness relates to not only the hit rate, but also to the payoff.

The size of the variant will influence on this. Image files are normally larger than plain text, so not having to send one is a big win.

There is not much difference between getting a 304 message and getting 500 lines of HTML. The delay is essentially the same in both cases because HTML files are so small.

Ad 7.7
Two out of five requirements that could not be met by HTTP had to be selected, and the needed added functionality to meet these requirements was the focus of 7.7.

None of the requirements can be solved directly by HTTP. Students that have submitted answers that vary from the proposed solution, but show understanding of the problem and knowledge of HTTP will get credit for this.

A proposed basis for the evaluation is given as follows:

Requirement (1): A Client may have several states. They are categorized in two states; online or offline.

Because HTTP is stateless, some other part of the application must include functionality for LoggedOn and LoggedOff states for each user.

A connection will be a representation of a Client and must be associated with one of the two states. The Server must be able to keep a binding between the Client's state and the connection. This implies that the Server will have to assign some identifier to each connection.

Functionality for this should be implemented as part of the Application Specific Module and left to it to handle.

Requirement (2): The Server must be able to send message to a given connected Client at any time

This requirement implies two things
Server push functionality [any given time].
The Server possesses the ability to differentiate between Clients.

HTTP does not have functionality for server-push. Push functionality will have to be added as part of the application specific module.

One solution is to introduce server functionality at the Client and client functionality at the Server. This way, the Server can initiate a connection with a normal HTTP request acting as a client. The Client will in this case act as a server and will reply to the Server by sending a HTTP response.

Furthermore, the Server must be able to differentiate between Clients. HTTP operates on connections given by IP-address [or hostname] and a port number. As such, the application specific module must provide the relation between a connection and a unique user of the system [a given client].

Requirement (3): The Server must be able to send messages to all connected Clients at any time

This implies two things:
Server push functionality [any given time].
The Server does not need the ability to differentiate between Clients, but should have functionality to handle the total set of users by keeping a list of connections.

HTTP does not have functionality for server-push. Push functionality will have to be added as part of the application specific module.

One solution is to introduce server functionality at the Client and client functionality at the Server. This way, the Server can initiate a connection with a normal HTTP request acting as a client. The Client will in this case act as a server and will reply to the Server by sending a HTTP response.

Furthermore, as HTTP operates on connections given by IP-address [or hostname] and a port number, the Server must maintain an inventory list of all open connections. It should also be quite easy to process a 1 – N message based on the inventory list.

Requirement (4): An online Client must be able to receive messages from the Server at any time (regardless of previous events).

This implies Server push functionality [any time].

HTTP does not have functionality for server-push. Push functionality will have to be added as part of the application specific module.

One solution is to introduce server functionality at the Client and client functionality at the Server. This way, the Server can initiate a connection with a normal HTTP request acting as a client. The Client will in this case act as a server and will reply to the Server by sending a HTTP response.

Requirement (5): The System design must not limit the number of separate Clients that can run on a single host

As the System needs push-functionality, a Client should have a HTTP server process that the Server knows how to reach (by addressing back to the hostname and a given port number).

In the case when the Client connects to the Server, the port number is universally known throughout the set of Client application entities. If this solution was to be used for the reverse channel (Server -> Client) as well, this would limit the number of Clients pr. host to one (1).

To meet the design requirement, the allocation of port numbers for Client HTTP server processes [on a single host] should not be static. Each Client must in turn inform the Server of this dedicated port through the application specific protocol.