EKSAMEN / EXAM	TTM4100	04	06	2010

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

Skriv kandidatnummeret ditt her ⇨⇨ Write your candidate number here ⇨						

1.1

Riktig Galt True False	Riktig Galt True False	Riktig Galt True False	Riktig Galt True False	Riktig Galt True False
1.1.1. ☐ .... ☒	1.1.2 . ☐ .... ☒	1.1.3. ☐..... ☒	1.1.4.. ☒ .... ☐	1.1.5 . ☒ ..... ☐
1.1.6. ☐ .... ☒	1.1.7 . ☐ .... ☒	1.1.8. ☒ .... ☐	1.1.9.. ☐ .... ☒	1.1.10 ☒ ..... ☐

1.2

Riktig Galt True False	Riktig Galt True False	Riktig Galt True False	Riktig Galt True False	Riktig Galt True False
1.2.1. ☐ .... ☒	1.2.2 . ☒ .... ☐	1.2.3. ☐..... ☒	1.2.4.. ☒ .... ☐	1.2.5 . ☐ .... ☒
1.2.6. ☒ .... ☐	1.2.7 . ☐ .... ☒	1.2.8. ☒ .... ☐	1.2.9.. ☒ .... ☐	1.2.10 ☐ .... ☒

1.3

Riktig Galt True False	Riktig Galt True False	Riktig Galt True False	Riktig Galt True False	Riktig Galt True False
1.3.1. ☐ .... ☒	1.3.2 . ☒ .... ☐	1.3.3. ☒ .... ☐	1.3.4.. ☒ .... ☐	1.3.5 . ☐ .... ☒
1.3.6. ☐ .... ☒	1.3.7 . ☐ .... ☒	1.3.8. ☒ .... ☐	1.3.9.. ☒ .... ☐	1.3.10 ☐ .... ☒

1.4

Riktig Galt True False	Riktig Galt True False	Riktig Galt True False	Riktig Galt True False	Riktig Galt True False
1.4.1. ☒ .... ☐	1.4.2 . ☒ .... ☐	1.4.3. ☒ .... ☐	1.4.4.. ☐ .... ☒	1.4.5 . ☐ .... ☒
1.4.6. ☐ .... ☒	1.4.7 . ☒ .... ☐	1.4.8. ☒ .... ☐	1.4.9.. ☒ .... ☐	1.4.10 ☐ .... ☒

1.5

Riktig Galt True False	Riktig Galt True False	Riktig Galt True False	Riktig Galt True False	Riktig Galt True False
1.5.1. ☐ .... ☒	1.5.2 . ☐ .... ☒	1.5.3. ☒ .... ☐	1.5.4.. ☐ .... ☒	1.5.5 . ☒ ..... ☐
1.5.6. ☒ .... ☐	1.5.7 . ☒ .... ☐	1.5.8. ☐..... ☒	1.5.9.. ☒ .... ☐	1.5.10 ☐ .... ☒

Kontroller:	Eksamensvaktens signature / Invigilator's signature
• Kandidatnr. på alle sider • Samme kandidatnr. over alt	

EKSAMEN / EXAM | TTM4100 | 04 06 2010

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

Skriv kandidatnummeret ditt her ⇨⇨
Write your candidate number here ⇨

2.

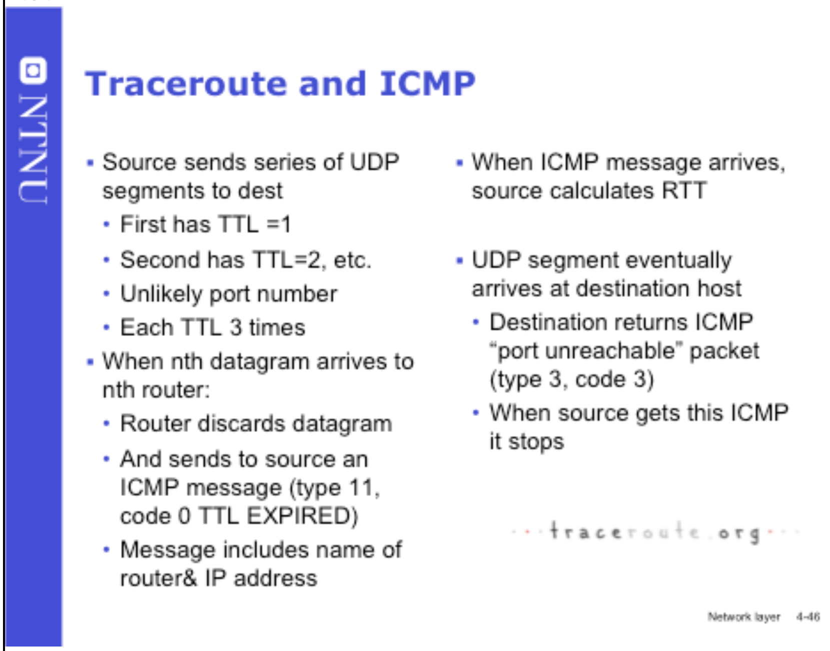
2.1

A protocol defines the format and order of messages sent and received among communicating entities, and actions taken on message transmission and receipt.

2.2

Packet is picked off network if destination **MAC address** of ethernet frame is equal to the MAC address of the end system. Ethernet frame **type field** indicates IP. Ethernet headers decapsulated and packet delivered IP protocol. **Protocol field** in IP indicates TCP. TCP protocol uses **the source and destination port field**s of the TCP header to decide which connection and application process the data belongs to.

2.3.1



2.3.2

Uses UDP to stop traceroute as ICMP port unreachable indicates packet has reached the destination. Thus, transport protocol port number is utilized, and IP does not have this. UDP is chosen as only a port number is needed, the services offered by TCP are not needed.

EKSAMEN / EXAM	TTM4100	04	06	2010

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

Skriv kandidatnummeret ditt her ⇨⇨ Write your candidate number here ⇨						

3.

3.1

Switches are plug and play, flat address structure, no configuration. They learn where all other nodes within the subnet are by looking av MAC addresses when they switch frames. (Flood when frame destination unknown.) Routers learn least cost paths through routing protocols which need to be configured

3.2

Switched ethernet has one sender and one receiver per segment. There are no collissions, and the switches can simultaneously transfer data between several segments.

3.3

Three MAC protocol classes + one protocol within each class should be shortly described.

1.Channel partitioning

- divide channel into smaller “pieces” (time slots, frequency, code)
- allocate piece to node for exclusive use

TDMA

- Access to channel in "rounds"
- Each station gets fixed length slot (length = packet transmission time) in each round
- Unused slots go idle

FDMA (Frequency Division Multiple Access)

- Channel spectrum divided into frequency bands
- Each station assigned fixed frequency band
- Unused transmission time in frequency bands go idle

CDMA (Code Division Multiple Access)

- Each group of users given a unique shared code
- All users share same frequency band
- Each user use its own “chipping” sequence (i.e. code) to encode data
- Many codes occupy the same channel, but only users associated with a particular code can understand each other
- (Allows multiple users to “coexist” and transmit simultaneously with minimal interference if codes are “**orthogonal**”)

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

Skriv kandidatnummeret ditt her ⇨⇨ Write your candidate number here ⇨						

2. Random access

- channel not divided, allow collisions
- when node has packet to send transmit at full channel data rate R
- must specifiy how to detect and recover from collisions

Slotted ALOHA

- All frames same size
- Time divided into equal size slots (time to transmit 1 frame)
- Nodes start to transmit only at slot beginning
- Nodes are synchronized
- When node obtains fresh frame, transmits in next slot
  - if no collision: node can send new frame in next slot
  - If 2 or more nodes transmit in slot, all nodes detect collision, node retransmits frame in each subsequent slot with probability p until success

AOHA (pure)

- When frame first arrives, transmit immediately
- Simpler than ALOHA, no synchronization
- Collision probability increases (frame sent at t_0 collides with other frames sent in $[t_0-1,t_0+1]$)
-

CSMA (Carrier Sense Multiple Access)

- Listen before transmit
  - If channel sensed idle, transmit entire frame
  - If channel sensed busy, defer transmission
- Collisions can still occur: two nodes may not hear each other’s transmission

CSMA/CD (CSMA/Collision Detect)

- carrier sensing, deferral as in CSMA
- Collision detection: colliding transmissions aborted, reducing channel wastage

CSMA/CA (CSMA/Collision Avoidance)

- Idea: allow sender to “reserve” channel rather than random access of data frames to avoid collisions of long data frames (request-to-send, clear-to-send)
- if sender senses channel idle for a period, transmit entire frame
- If sense channel busy then
  - start random backoff time
  - timer counts down while channel idle
  - transmit when timer expires
  - if no ACK from receiver, increase random backoff interval

EKSAMEN / EXAM	TTM4100	04	06	2010

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

Skriv kandidatnummeret ditt her ⇨⇨ Write your candidate number here ⇨						

3. Taking turns

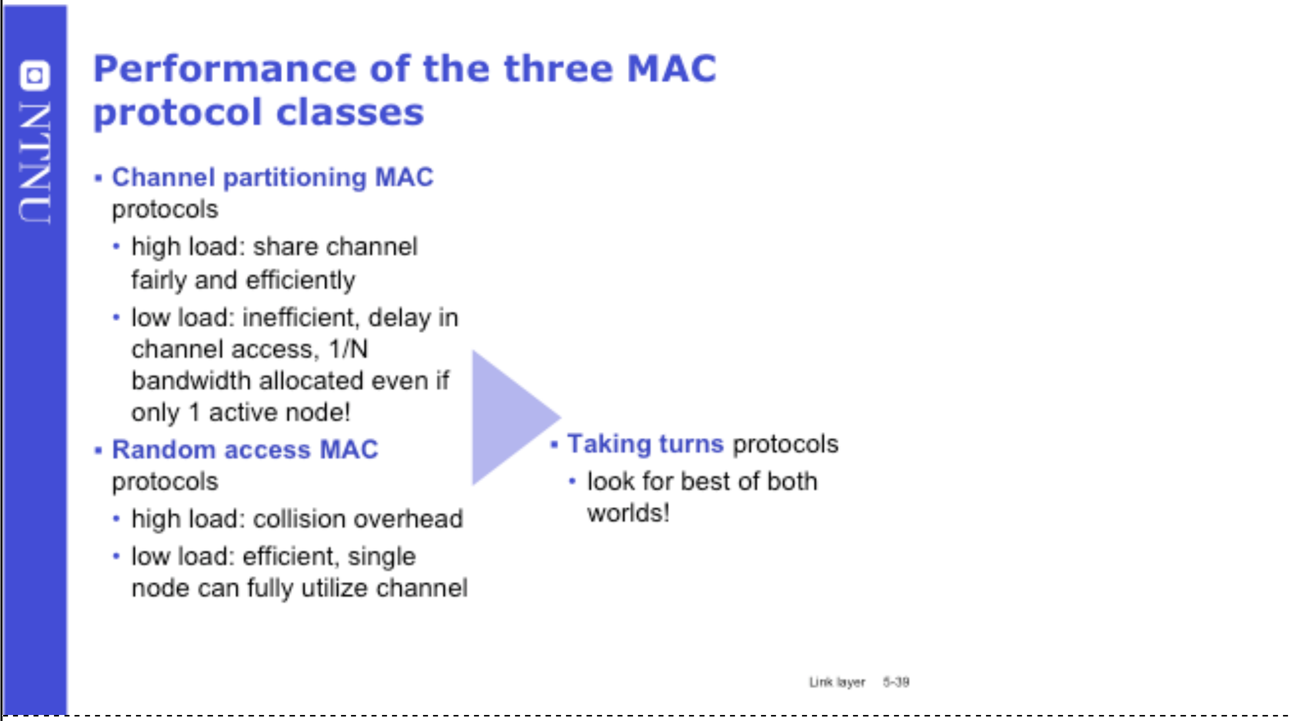
- nodes take turns, but nodes with more to send can take longer turns

Polling

- master node “invites” slave nodes to transmit in turn
- typically used with “dumb” slave devices

Token passing

- control token passed from one node to next sequentially
- token message
- no master node



Channel partitioning (and taking turns) best at high load, random best at low traffic load. (Taking turn do have some overhead to do polling/token passing, but are better than channel partitioning at low loads)

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

Skriv kandidatnummeret ditt her ⇨⇨
Write your candidate number here ⇨

4.

	Frame len	IP len	UDP or TCP data	Src MAC	Dst MAC	Src IP	Dst IP	Trans-port proto-col	TP proto-col flags	Highest protocol and msg type	Comments
1	342	328	308	00:1f:f3:5a:12:33	ff:ff:ff:ff:ff:ff	0.0.0.0	255.255.255.255	**UDP**	**----**	**DHCP req**	Only IP broadcast
2	342	328	308	00:0c:cf:32:48:00	00:1f:f3:5a:12:33	129.241.66.1	129.241.67.145	**UDP**	**----**	**DHCP ACK**	Only from 66.1
3	42	---	---	00:1f:f3:5a:12:33	ff:ff:ff:ff:ff:ff	---	---	**---**	**----**	**ARP req**	Ethernet broadcast without IP
4	60	---	---	00:0c:cf:32:48:00	00:1f:f3:5a:12:33	---	---	**---**	**----**	**ARP rsp**	
5	74	60	40	00:1f:f3:5a:12:33	00:0c:cf:32:48:00	129.241.67.145	129.241.0.200	**UDP**	**----**	**DNS query**	Only communication to
6	142	128	108	00:0c:cf:32:48:00	00:1f:f3:5a:12:33	129.241.0.200	129.241.67.145	**UDP**	**----**	**DNS resp**	this server
7	78	64	0	00:1f:f3:5a:12:33	00:0c:cf:32:48:00	129.241.67.145	74.125.79.147	**TCP**	**SYN**	**--**	TCP connection set-up
8	74	60	0	00:0c:cf:32:48:00	00:1f:f3:5a:12:33	74.125.79.147	129.241.67.145	**TCP**	**SYN ACK**	**--**	
9	66	52	0	00:1f:f3:5a:12:33	00:0c:cf:32:48:00	129.241.67.145	74.125.79.147	**TCP**	**ACK**	**--**	
10	946	932	880	00:1f:f3:5a:12:33	00:0c:cf:32:48:00	129.241.67.145	74.125.79.147	**TCP**	**ACK**	**HTTP REQ**	HTTP GET over TCP
11	1484	1470	1418	00:0c:cf:32:48:00	00:1f:f3:5a:12:33	74.125.79.147	129.241.67.145	**TCP**	**ACK**	**HTTP RSP**	HTTP response over TCP
12	79	52	13	00:0c:cf:32:48:00	00:1f:f3:5a:12:33	74.125.79.147	129.241.67.145	**TCP**	**ACK**	**HTTP RSP**	“
13	66	52	0	00:1f:f3:5a:12:33	00:0c:cf:32:48:00	129.241.67.145	74.125.79.147	**TCP**	**ACK**	**----**	
30 sec traffic pause											
14	66	52	0	00:1f:f3:5a:12:33	00:0c:cf:32:48:00	129.241.67.145	74.125.79.147	**TCP**	**FIN**	----	TCP connection close
15	66	52	0	00:0c:cf:32:48:00	00:1f:f3:5a:12:33	74.125.79.147	129.241.67.145	**TCP**	**ACK**	----	
16	66	52	0	00:0c:cf:32:48:00	00:1f:f3:5a:12:33	74.125.79.147	129.241.67.145	**TCP**	**FIN**	----	
17	66	52	0	00:1f:f3:5a:12:33	00:0c:cf:32:48:00	129.241.67.145	74.125.79.147	**TCP**	**ACK**	----	

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

Skriv kandidatnummeret ditt her ⇨⇨ Write your candidate number here ⇨						

5.

5.1
Flow control and congestion control are two *distinct* control mechanisms with *distinct objectives*.
Flow control makes sure that the sender of a connection does not overwhelm the buffers at the receiver by sending too many packets too fast.
Congestion control regulates the amount of data that an application can send into the network, helping to prevent congestion in the network (i.e. in the network router buffers).

5.2
TCP strategy for adjusting sending rate:

- Increases transmission rate by probing for usable bandwidth until loss occurs (slow-start, additive increase)
- Reduces rate after loss event (receive duplicate acks, timeout)

Implemented by changing the congestion window that limits number of bytes to be transmitted.

5.3
5.3.1 Slow start: 1-6, 23-26
5.3.2 Congestion avoidance 6-16, 17-22
5.3.3 16: duplicate acks, 22: time-out
5.3.4 Threshold initial value 32, 18: 21

6.

6.1
ServerSocket(), accept(), read(), write() and then close().

6.2
255.255.255.240 = /28

6.3
TCP is byte oriented. Need to seperate the http messages. HTTP has a `Content-Length` field indicating number of bytes after the space indicating end of header lines and start of message body. The server MUST return the responses in the same order as they were received.