

24 timer HJEMMEEKSAMEN

TK1104 DIGITAL TEKNOLOGI

Tillatte hjelpeemidler: Alle

Varighet: 24 timer

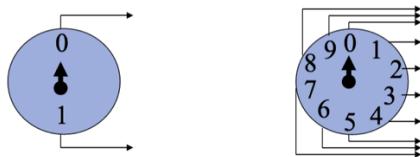
Karakterskala/vurderingsform: Bestått / Ikke bestått Dato: 13.-14. desember 2021

Oppgave 1. Generelt (15 %)

- a) Forklar hvorfor datamaskiner bruker binær logikk. Forklar kort hva binær logikk og maskinens fysiske oppbygning har å si for pålitelighet og produksjonskostnader for komponenter i en datamaskin. Tegn gjerne noen figurer som illustrerer dette.

En datamaskin bruker maskin språk som sin grunnleggende behandling og lagring av informasjon, instruksjoner og adresser og minner. Til dette har rådighet elektroniske impulser der strømmen enten på eller av. Ved «på» er notifisert med 1, ved «av» er det notifisert med 0. Maskinen har derved til rådighet å behandle alle instruksjoner med et språk bestående av 0 og 1. Dette kaller vi et binært språk. Informasjon lagres i RAM/CPU (Prosessor). Maskin bruker deretter binær logikk til utføre operasjoner. Dette er enkelt og plassbesparende.

Binærlogikk er enkelt å implementere. Maskinen har til rådighet strøm på eller strøm av (1 eller 0). Dette er et enkelt konsept, robust og mindre utsatt feil. Lagring foregår i bits/bytes bestående sekvenser av nuller og enere. I rimelig data modeller har man integrerte hovedkort som er rimelig. I det mer kostbare data maskiner har man utskiftbare komponenter på hovedkortet.



Kilde: TK1104_Leksjon_0x00_Intro

Maskinen behandler alle operasjoner, instruksjoner og adresser i CPU. Informasjon lagres binært i et mellomlagringshurtiglager for rask, sekvensiell eller parallel (Multitasking) behandling. Desto større kapasitet i CPU og tilsvarende i mellomlagret, jo raskere kan maskinen behandle instruksjonene. Samtidig øker kostanden med størrelsen på CPU og mellomlagring. Fordelen med stor kapasitet i CPU og mellomlagring er at man redusere «Overflow», «Crash», og andre feil som oppstår når maskinen pålegges flere operasjoner enn den kan behandle.

5	Applikasjonslaget
4	Transportlaget
3	Nettverkslaget/Internet
2	Datalinjelaget/Lenker
1	Fysisk

- b) i) Forklar forskjellen på TCP og UDP protokollene på Transportlaget.

TCP: «Transmission Control Protocol» er en standard internetmodus som håndter og organiserer overførsel av data, håndterer tap av data, håndterer «Float Control» samt kontrollerer hastighet hos sender når kø oppstår. TCP har ingen «Timing Control»

UDP: «User Datagram Protocol» er en lett vektløsing av TCP uten garantier for feil i data transmisjon. Den er enkelt, men segmenter kan bli tapt, mulig sekvensfeil og feil i handshake.

Hvorfor UDP?:

- Ikke noe oppsett er nødvendig for transmisjon,
- Den er «Stateless»
- Lite segmenthode
- Ingen kø håndtering
- Tillater Kringkasting (Broadcasting)

- ii) Gi noen eksempler på situasjoner det vil være fornuftige å velge den ene fremfor den andre.

Applications and tr-protocols:

Email	smtp / RFC 821	TCP
Terminal access	telnet / RFC 854	TCP
Web	http / RFC 2068	TCP
File Transfer	ftp / RFC 959	TCP
Streaming multimedia	proprietary	TCP / UDP
Remote file server	NES	TCP / UDP
IP telephone	proprietary	UDP

- c) Forklar hva en «floppy disk» i en PC er (var), hvilket bruksområde denne har?

Floppy Disk var vanlig en lagring metode i tidligere pc. Lagring var delt opp i spor og sektorer nummerert utenfra og innover. En sektor var typisk 512 byte. Sektorene ble lest parallelt. Sammenhengende sektorer innen samme spor ble kalt for blokk. Blokkstørrelsene var på 4kb. Aksessiden var langsomt i forhold til andre lagringsmedia i dag.

Oppgave 2. Tall og binære data (35 %)

a) Konverter de to følgende desimaltallene til binærtall (16 bits presisjon). (Vis utregning.)

$$527_{10} = 0000\ 00010\ 0000\ 1111_2$$

527 =	64 + 35														
15 =	8 + 7														
7 =	4 + 3														
3 =	2 + 1														
1 =	1														
2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
32768	16384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1
0	0	0	0	0	0	0	0	1	1	0	0	0	1	1	1

$$99_{10} = 0000\ 0000\ 0110\ 0011_2$$

99 =	64 + 35														
35 =	32 + 3														
3 =	2 + 1														
1 =	1														
2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
32768	16384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1
0	0	0	0	0	0	0	0	0	1	1	0	0	0	1	1

b) Utfør følgende binær addisjon med 8 bits presisjon. (Vis utregning.)

$$0111\ 1000_2 + 0110\ 1101_2 = 1110\ 0101_2$$

$$\begin{array}{r}
 & 1 & 1 & 1 \\
 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
 + & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \\
 \hline
 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1
 \end{array}$$

$$1100\ 1101_2 + 1100\ 1111_2 = 1001\ 1100_2$$

overflow

$$\begin{array}{r}
 & 1 & 1 & 1 & 1 & 1 \\
 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\
 + & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \\
 \hline
 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0
 \end{array}$$

- c) Utfør BINÆR subtraksjon av de to desimale tallene 376 minus 77. (Vis utregning ved bruk av toerkomplement, og utregningen MÅ gjøres med binære tall.)

$$376_{10} - 77_{10} = 0000\ 0001\ 0010\ 1011_2$$

376 =	256 + 120
120 =	64 + 56
56 =	32 + 24
24 =	16 + 8
8 =	8
2^15 2^14 2^13 2^12 2^11 2^10 2^9 2^8 2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0	32768 16384 8192 4096 2048 1024 512 256 128 64 32 16 8 4 2 1
375 ₁₀	0 0 0 0 0 0 0 1 0 1 1 1 1 0 0 0
77 =	64 + 13
13 =	8 + 5
5 =	4 + 1
1 =	1
2^15 2^14 2^13 2^12 2^11 2^10 2^9 2^8 2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0	32768 16384 8192 4096 2048 1024 512 256 128 64 32 16 8 4 2 1
77 ₁₀	0 0 0 0 0 0 0 0 0 1 0 0 1 1 0 1
Flippe 77 ₁₀	1 1 1 1 1 1 1 1 1 0 1 1 0 0 1 0
Pluss 1	1
Sum Flipt	1 1 1 1 1 1 1 1 1 0 1 1 0 0 1 1
1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1	
375 ₁₀	0 0 0 0 0 0 0 1 0 1 1 1 1 1 0 0
+	1 1 1 1 1 1 1 1 1 0 1 1 0 0 1 1
Sum 299 ₁₀ :	0 0 0 0 0 0 0 1 0 0 1 0 1 0 1 1

- d) Forklar forskjellen på UTF-8 og UTF-16, i hvilke tilfeller kan det lønne seg å velge det eneenkodingsformatet over den andre.

Unicode er et universalt kodesystem for å få med seg alle tegn, bokstaver og tall i verdens ulike språk. Det tillater 65536 kode punkter på 17 plan (mens ASCII kun tillater 128 tegn.)

Unicode enkodes ved hjelp transformasjon-formater. Det kan kodes både som UTF-8 og UTF-16. UTF-8 er den vanlige standard til internett.

UTF-8 og USC-2 er sub set av UTF-16

UTF-16 = Unicode points mens USC-2 er alltid 2 bytes

UTF-16 er vanligvis 2 bytes men kan være flere.

UTF-16 er Vanlig format for strenger i Java.

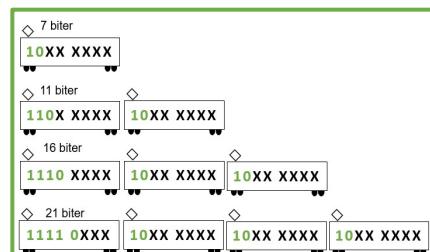
UTF-8: U+0000 til U+007F er det samme for ASCII

Plus to eller flere bytes for å kode tegn, minimum 1 byte

Over U+007F bruker UTF-8 to eller flere byte for å kode et tegn.

UFT-8 tillater koding på følgende måte:

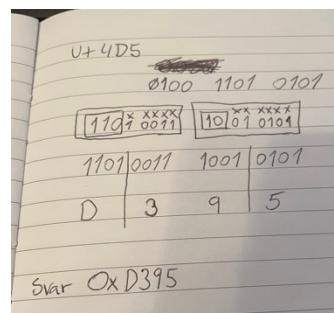
Kilde: TK1104_Leksjon_0x02_TXTmedia



Vanligvis bruker UTF-16 to byte for å kode et tegn, mens UTF-8 bruker tre eller flere tegn for å kode samme. UTF-16 sparar derfor plass på lagringsmediet.

Når meste parten av kodingen greier seg med ASCII tegn er det fordel å bruke UTF-8.

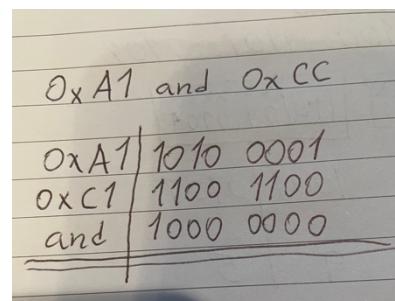
e) Utfør UTF-8 enkoding av Unicode-punktet U+4D5 (et tegn fra Kyrillisk alfabet – som ser likt ut som en bokstav i vårt latinske alfabet). Oppgi svaret i hex.



U+4D5
0100 1101 0101
1101 0011 1101 0101
1101 0011 1001 0101
D | 3 9 | 5
Svar 0xD395

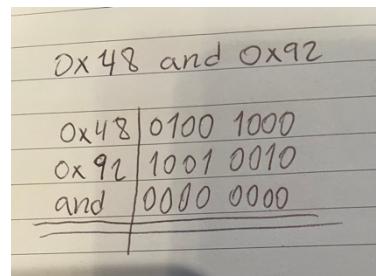
f) Utfør følgende boolske operasjoner. (Vis utregning.)

0xA1 AND 0xCC = 1000 0000



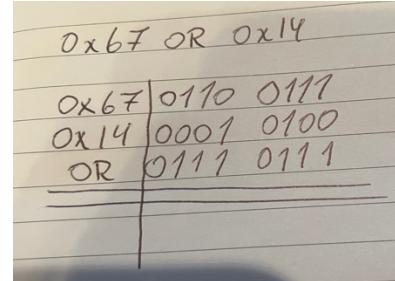
0xA1 and 0xCC
0xA1 | 1010 0001
0xCC | 1100 1100
and | 1000 0000

0x48 AND 0x92 = 0000 0000



0x48 and 0x92
0x48 | 0100 1000
0x92 | 1001 0010
and | 0000 0000

0x67 OR 0x14 = 0111 0111



0x67 OR 0x14
0x67 | 0110 0111
0x14 | 0001 0100
OR | 0111 0111

0xF0 OR 0x89 = 1111 1001

0xF0 OR 0x89	
0xF0	1111 0000
0x89	1000 1001
OR	1111 1001

0x51 XOR 0x95 = 1100 0100

0x51 XOR 0x95	
0x51	0101 0001
0x95	1001 0101
XOR	1100 0100

g) Hva er bitstyrken til et passord med 8 tegn hvor passordet består av små bokstaver (a-z) og tall? (Vis utregning.)

Bit styrke måler kvaliteten på et passord og uttrykker hvor mange forsøk en angriper trenger for å finne passordet (Brute Force)

Bit styrke = \log_2 (forskjellige tegn i passordet) * antall tegn i passordet.

Bit styrke = $\log_2 (26+10)*8 = \log_2(36)*8 = 5.169925 = \underline{41.3594}$

Oppgave 3. Praktiske oppgaver (30 %)

- a) Hva gjør kommandoen "nslookup -query=ns . > table" på kommandolinje/terminal? (Du kan svare for det operativsystemet du kjenner best; Windows, Linux eller OSX, men du må oppgi hvilket operativsystem du har brukt i besvarelsen.)

Jeg bruker OSX som er bygget på toppen av UNIX. Jeg bruker Terminal for spørring.

```
[Desktop: ~] ~$ nslookup -query=ns
> table
Server: (on Mac) 192.168.1.1
Address: 192.168.1.1#53
** server can't find table: NXDOMAIN
```

Av sikkerhetshensyn har jeg overstrøket med gult:

User navn

Min Server adresse

Min Router adresse

- b) Demonstre kommandoen «ftp» på kommandolinje/terminal, bruk domenet <ftp://speedtest.tele2.net> og last ned noen testfiler fra serveren. Hvilke nedlastingshastigheter oppnår du? Dokumenter kommandoer du har brukt og resultater enten med tekst eller screenshot.

```
$ telnet speedtest.tele2.net 80
Trying 90.130.70.73...
Connected to speedtest.tele2.net.
Escape character is '^A'.
POST /upload.php HTTP/1.1
Host: speedtest.tele2.net
Content-Length: 3

abc
HTTP/1.1 200 OK
Server: nginx
Date: Mon, 13 Dec 2021 19:12:41 GMT
Content-Type: text/html; charset=UTF-8
Transfer-Encoding: chunked
Connection: keep-alive

8
size=500
0
```

- c) I denne oppgaven skal du både demonstrere forståelse av bruk av verktøyet Wireshark og web browser, og generell forståelse av URL og web sider.

Først må bruke terminal for finne www.eastwillsecurity.com sever.

Kommandoen er nslookup www.eastwillsecurity.com

Deretter brukte jeg wireshark for å spore (Sniffe) web browser ved å bruke ip.addr == 77.111.240.75

Den blå linjen viser informasjonen.

```
$ nslookup www.eastwillsecurity.com
Server: [REDACTED]0.1
Address: [REDACTED]#53
Non-authoritative answer:
Name: www.eastwillsecurity.com
Address: 77.111.240.75
```

```

ip.addr == 77.111.240.75
No. | Time | Source | Destination | Protocol | Length | Info
---|---|---|---|---|---|---
 84 | 2021-12-13 20:26:34.477141 | 77.111.240.75 | 192.168.10.142 | TCP | 74 | 88 -> 55134 [SYN, ACK] Seq=0 Ack=1 Win=13177
 85 | 2021-12-13 20:26:34.477576 | 192.168.10.142 | 77.111.240.75 | TCP | 66 | 55133 -> 80 [ACK] Seq=1 Ack=1 Win=13177
 86 | 2021-12-13 20:26:34.477777 | 192.168.10.142 | 77.111.240.75 | TCP | 66 | 55134 -> 80 [ACK] Seq=1 Ack=1 Win=13177
+ 87 | 2021-12-13 20:26:34.478114 | 192.168.10.142 | 77.111.240.75 | HTTP | 643 | GET / HTTP/1.1
 88 | 2021-12-13 20:26:34.489990 | 77.111.240.75 | 192.168.10.142 | TCP | 66 | 80 -> 55133 [ACK] Seq=1 Ack=578 Win=643
 89 | 2021-12-13 20:26:34.489996 | 77.111.240.75 | 192.168.10.142 | HTTP | 397 | HTTP/1.1 304 Not Modified

> Frame 87: 643 bytes on wire (5144 bits), 643 bytes captured (5144 bits) on interface en0, id 0
> Ethernet II, Src: Apple_b1:9a:c9 (98:01:a7:b1:9a:c9), Dst: ZyxelCom_dd:B3:b3 (e4:18:6b:dd:B3:b3)
> Internet Protocol Version 4, Src: 192.168.10.142, Dst: 77.111.240.75
> Transmission Control Protocol, Src Port: 55133, Dst Port: 80, Seq: 1, Ack: 1, Len: 577
+ Hypertext Transfer Protocol

0000 | e4 18 b6 dd 83 b3 98 01 a7 b1 9a c9 08 00 45 00 .K... . . . . E.
0001 | 02 75 00 00 40 00 40 00 40 06 2f 92 c0 a8 0a 8e 4d 6f .K_@. / .Mo
0002 | 08 4b d7 5d 00 50 a2 42 d1 e9 90 45 58 b2 80 18 K ] P " .EX. .
0003 | 08 0a a3 49 00 00 01 01 08 0a 7a 08 06 18 9a a7 .I... z. . .
0004 | e7 3d 47 45 54 20 27 20 48 54 54 58 27 31 2e 31 .=GET / HTTP/1.1
0005 | 0d 0a 48 6f 73 74 3a 20 77 77 77 2e 65 61 73 74 .Host: www.east
0006 | 77 69 6c 73 65 63 75 72 69 74 79 2e 63 6f 70 willsecu rity.com
0007 | 65 6c 73 65 63 75 72 69 74 79 2e 63 6f 70 65 63 .Connect: keep-alive
0008 | 65 6c 73 65 63 75 72 69 74 79 2e 63 6f 70 65 63 .applied. . .
0009 | 43 6f 74 72 6f 6c 3a 20 6d 61 78 2d 61 67 65 Control: max-age
000a | 3d 30 0d 0a 55 70 67 67 62 64 65 2d 49 6e 73 65 =0.Upgrade-Insecure-Requests: 1
000b | 63 75 72 65 2d 52 65 71 75 65 73 74 73 3a 20 31 .User-Agent: Mo
000c | 0d 0a 55 73 65 72 2d 41 65 65 74 2a 3d 4d 6f .User-Agent: Mo

```

d) I denne oppgaven skal du demonstrere forståelse for filer og filformater, og praktisk bruk av en hexeditor. Først last ned denne filen:

http://www.eastwillsecurity.com/tk1104/oppgave3d/eksamen_oppg.dta

Magic Number i denne filen er CAFE BABE

Fil tabellen er type Unicode

0x4C = 0100 1100 som skrive ut alfabet L

	0	1	2	3	4	5	6	7	8	9	Å	ß	C	D	E	F	0123456789ÅBCDEF
0000h:	CA	FE	BA	BE	00	00	00	34	00	97	0A	00	22	00	4C	0A	E...4.-..L.
0010h:	00	4D	00	4E	07	00	4F	0A	00	03	00	4C	0A	00	50	00	.M.N..O....L..P.
0020h:	51	0A	00	26	00	52	09	00	26	00	53	07	00	54	0A	00	Q..&R..&S..T..
0030h:	08	00	4C	0A	00	26	00	55	0A	00	26	00	56	0B	00	57	..L..&U..&V..W
0040h:	00	58	07	00	59	0A	00	0D	00	4C	0B	00	57	00	5A	0B	X..Y....L..W.Z.
0050h:	00	5B	00	5C	0B	00	5D	00	5E	0B	00	5D	00	5F	07	00	.[.].^.A...-..
0060h:	60	0B	00	13	00	63	07	00	64	0B	00	65	0A	00	15	00c..d..e....
0070h:	66	0B	00	13	00	67	08	00	68	0B	00	69	00	6A	0B	00	f....g..h..i.j..
0080h:	69	00	6B	07	00	6C	0B	00	69	00	6D	07	00	6E	0A	00	i.k..l..i.m..n..
0090h:	26	00	6F	0A	00	26	00	70	0A	00	26	00	71	07	00	72	&o...&p...&q..r
00A0h:	08	00	73	0A	00	4D	00	74	07	00	75	07	00	76	07	00	s M.t..u..y..

e) I denne oppgaven skal du demonstrere grunnleggende ferdigheter i shell/terminal, oppgaven består av 6 små deler som skal utføres i rekkefølge.

Del 0: Åpne terminal (Linux/OSX) eller kommandolinje (Windows)

```
Last login: Mon Dec 13 21:13:12 on ttys000
-bash - 80x24
The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
[redacted]MBP:~ [redacted]$
```

Del 1: Opprett en ny mappe/katalog som heter 'OPPG_3E'

```
Last login: Mon Dec 13 21:13:12 on ttys000
The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
[redacted]P:~ [redacted]$ mkdir OPPG_3E
[redacted]P:~ [redacted]$
```

Del 2: Gå inn i mappen du opprettet

```
MBP:~ [REDACTED]b$ cd OPPG_3E/  
MBP:OPPG_3E [REDACTED]b$
```

Del 3: Opprett en fil som heter 'TEST.TXT' med innholdet 'TESTTESTTEST'

DEL 4: Skriv ut innholdet av filen på skjermen (ut i terminal/kommandolinje)

```
MBP:OPPG_3E [REDACTED]b$ echo TESTTESTTEST > TEST.txt  
MBP:OPPG_3E [REDACTED]b$
```

Del 5: Slett filen du opprettet

```
MBP:OPPG_3E [REDACTED]b$ rm TEST.txt  
MBP:OPPG_3E [REDACTED]b$
```

Del 6: Slett mappen/katalogen du opprettet

```
MBP:~ [REDACTED]b$ rmdir OPPG_3E/  
MBP:~ [REDACTED]b$
```

Oppgave 4. Forståelse av nettverk (20 %)

- a) Forklar forskjellen på en Hub, en Switch og en Router i et nettverk. Forklar hvordan de fungerer, hvilket TCP/IP lag de opererer på, og hvor i et nettverk de typisk brukes.

Hub:

- Hver mottatt byte bli omdirigert til andre lenker.
- Ingen buffering
- Ingen CSMA/CD – NIC oppdager kollisjoner.

Switch:

- Linker lang segmenter
- Utvider distansen mellom HUBene
- 10 BaseT og 100 BaseT er kompatible for linkup:
 - Slike koblinger krever en bro (brigde)
- Link Layer box: Mellomlagrer og transmisjoner av rammer (frames).
- Sjekker hoder (heads) når det er nødvendig
- Gjennomsiktig (Transparent)
- Plug and play
- Switch har en Switch tabell

Router:

- Router har lagring og videre sender
- Router er «Net layer units» (3. lag).
- Router bruker ruter tabeller / implementere ruter algoritmer

Routers vs. Switches:

- Router er net layer units (3. lag) / Switches er link layer units (2. lag).
- Router bruker routing tables / Switches bruker switch tables.
- Router implementerer router algoritmer / Switches filtrer + plug & play.

TCP/IP hadde fire nivåer: Application, Transport, Internet og Network Interface. Dagens TCP/IP har fem lag: Application, Transport, Network, Datalink og fysisk. TCP/IP headeren 20 bytes stor.

TCP brukes i transport lageret / IP i netverk routing.

- b) Forklar hvordan Linklagsprotokoller basert på Multiple Access (MA) fungerer, forklar også hvordan CSMA protokoller løser utfordringene med Multiple Access.

MA er en delt kanal (one shared channel) som kan motta «broadcasting». Den er «half-duplex» for å hindre interferens. Den benytter en distribuert algoritme som bestemmer hvordan nodene deler trafikken.

CSMA lytter (listens) før sending. Hvis kanalen er fri; send. Hvis den er opptatt; vent til den er ledig. Det er imidlertid et problem hvis nodene er for langt fra hverandre siden det øker fare for kollisjoner. Konsekvensen er at man deler mediet hvilket gjør det mulig å dele kapasiteten.

CSMA/CD (Collisions Control) Det er en videreutvikling av CSMA med kollisjons kontroll. Den lytter før sending; hvis ledig: send. Hvis opptatt; venter + lytter mens sending pågår; hvis kollisjon inntreffer: sendingen stoppes. Den oppdager kollisjoner på følgende måte: måler signalstyrken i enkle kablede nettverk. Dette er mere komplisert i wifi nettverk.

- c) Forklar hvordan DHCP fungerer, og hvilken oppgave denne protokollen har i et nettverk. Hva skjer hvis en maskin som skal bruke en DHCP server ikke får svar fra den – kan man sette opp et nettverk uten å bruke DHCP?

DHCP gjennomfører adressering med Plug & Play.

- Host: oppdager DHCP
- Server: DHCP tilbyr (offer),
- Host: DHCP forespørsel (query),
- Server: IP adr. + nettverk parametere + DHCP ack
- Host: Aksepterer verdiene som er sendt.
 - C:\User\blist og > ipconfig / frislipp – fornyelse

DHCP finner nettverk parameterne:

- Klienten spør nettverket om IP, nw.maske, std. Gateway, DNS ved å utstede FF-FF-FF-FF-FF-FF i LAN (for oppdagelse)
- Ledig DHCP server svarer med tilbud (oppbevares in nettverket)
- Når «leien» slutter, blir prosessen fornyet.

Hvis ikke noe svar fra Server foreligger, vil Host fortsette å tilby DHCP til svar oppnås. Hvis ikke noe svar foreligger etter x antall forsøk, brytes forbindelsen.

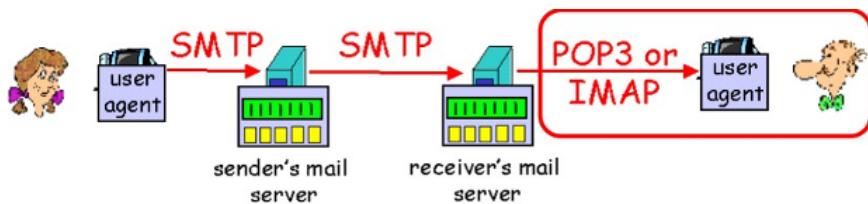
Det er vanskelig å sette opp et nettverk uten DHCP, men man kan tilordne statiske adresser.

- d) Forklar hvordan SMTP protokollen (for sending av epost) fungerer. Illustrer forklaringen med en skjematisk tegning som viser dataflyt mellom klient og server.

SMTP bruker TCP for å sende email fra klient til server; port 25 (587)
Protokollen etablerer direkte transmisjon av data server til server;

- Gjennomfører «handshake»
- Overføring av data finner sted
- Avslutter transmisjon.

Transmisjon av data foregår ved ASCII tekst. Meldingen sendes med 7-bits ASCII. Sendingen avsluttes med ‘.’



Kilde: TK1104_Leksjon_0x07_ApplikasjonsLaget