Høgskolen i Østfold

Løsningsforslag til

EKSAMEN

Emnekode: **ITF20205**	Emne: **Datakommunikasjon**
Dato: **09.Des 2013**	Eksamenstid: **kl 9:00** til kl **13:00**
Hjelpemidler: • 4 sider (A4) (2 ark) med egne notater. • Kalkulator. • Gruppebesvarelse, som blir delt ut på eksamensdagen til de som har fått den godkjent	Faglærer: Erling Strand
Eksamensoppgaven: Oppgavesettet består av totalt 6 sider, hvorav 4 sider med oppgaver, og 2 sider med vedlegg. Kontroller at oppgaven er komplett før du begynner å besvare spørsmålene. *Oppgavesettet består av 3 oppgaver. Alle spørsmålene teller likt.*	
Sensurdato: 3. Januar 2014 Karakterene er tilgjengelige for studenter på studentweb senest dagen etter oppgitt sensurfrist. Følg instruksjoner gitt på: http://www.hiof.no/index.php?ID=7027	

Oppgave 1

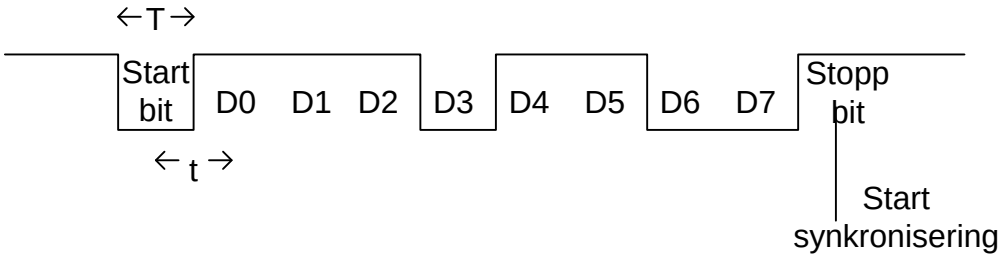
a) *Hva er forskjellene på, og hvordan virker synkron og asynkron dataoverføring?*

Synkronisering betyr at sender og mottager synkroniserer seg slik at databitene tas imot riktig, dvs at for eksempel databit 0 blir tatt imot som databit 0, og databit 1 som databit 1 osv..

Asynkron

Her foregår synkroniseringen for hver byte. Det sendes en startbit før databitene sendes. En forbindelse hvor det ikke går data, er på logisk 1. Startbiten er logisk 0, og det er den som mottageren synkroniserer seg på. Det gjøres ved at mottageren punktprøver det mottatte signal med en høyere (16x eller 64x) frekvens enn dataene. Med en gang det oppdages en logisk 0, starter mottageren. En ny punktprøve blir tatt en halv bitlengde senere, slik at midten i

startbiten blir prøvet. Hvis det (fortsatt) er logisk 0, er mottageren sikker på at det er startbiten. De kommende 7 eller 8 bit blir punktprøvet mitt i biten, og plassert i riktig bitposisjon i mottageren (USART). Når siste bit er lest, forventer mottageren en (eller 2) stoppbit, som er logisk 1. Når stoppbiten er mottatt, begynner mottageren å lete etter neste startbit, slik at den kan synkronisere seg på den neste byten.



Synkron

Dette er nøye gjennomgått:

Ved synkron dataoverføring foregår synkroniseringen i starten av hver blokk. Når synkronisering først er oppnådd, er det ingen synkronisering igjen før neste blokk. Det er da viktig at senderklokka og mottagerklokka er eksakt like, derfor overføres senderklokka i sammen med dataene. Det gjøres ved å kode inn klokkesignalet i datastrømmen. På mottagersiden blir dette klokkesignalet dekodet ut av datastrømmen, og brukes til å klokke inn dataene. På den måten blir mottagerklokka og senderklokka eksakt like.

Dette er gjennomgått, men det ble brukt lite tid på det:

Synkroniseringen kan foregå på to måter. Det ene er bytesynkronisering. Da er det en bestemt byte, synkroniseringsbyte, som er lagret i mottageren. I starten av blokka er det en slik synkroniseringsbyte. Mottageren sjekker det mottatte bitmønsteret med denne byten for hver bit mottatt. Da bitmønsteret er likt, er synkronisering oppnådd. Hvis denne synkbyten er en del av dataene, setter senderen inn en ekstra byte, som betyr at etterfølgende byte ikke er synkroniseringsbyte, men en databyte. Denne ekstra byten blir tatt ut på mottagersiden.

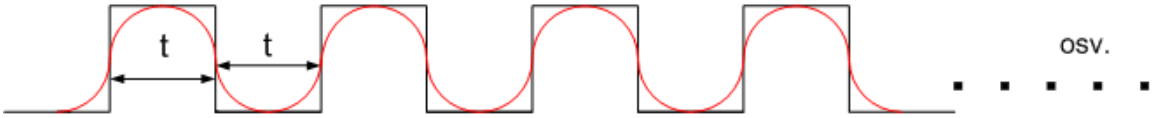
Dette er ikke gjennomgått, og de trenger da ikke ta med denne metoden:

– Den andre måte en bitsynkronisering. Der ser mottageren etter et bestemt bitmønster. Når det har kommet 6 enere, etter en nuller, og det deretter kommer en nuller, er synkronisering oppnådd. Hvis det samme synkroniseringsmønsteret kommer som en del av dataene, vil senderen sette inn en ekstra 0 i datastrømmen, slik at det ikke kommer 6 1'ere etter hverandre i dataene. Denne ekstra 0'er blir tatt ut igjen ved mottageren

b) *Anta at du skal ha et bithastighet på 100 Mbit/s. (Altså $100 \cdot 10^6$ bit/s). Hvilket krav må du sette til båndbredden på overføringsmediet, hvis det brukes:*

a. *Ukodet datasignal.*

Ved bruk av ukodet datasignal, er kravet til båndbredden halvparten av bithastigheten, Kravet til båndbredde blir altså 100/2 MHz = **50 MHz**. Dette kan illustreres med en figur der det vises sending av skiftevis 0, 1, 0,1 (ukodet) osv. Hvis bithastigheten er $100 \cdot 10^6$ bit/s, blir tiden for en bit:
$t = 1/(100 \cdot 10^6) = 10 \cdot 10^{-9}$ s = 10 ns.
Hvis dette signalet blir sendt gjennom et overføringsmedium hvor grensefrekvensen er 50 MHZ, vil signalet se ut som den røde kurven i figuren. Dette er også grunnfrekvensen til datasignalet, som skifter mellom 0, 1, 0 ,1 osv. Perioden i en frekvens er definert som tiden mellom et punkt på sinuskurven, til tilsvarende punkt på neste syklus. I figuren blir det 2·t.
Frekvensen til dette signalet er $f = 1/(2 \cdot t) = 1/2 \cdot 10 \cdot 10^{-9}$ [1/s]= 50 MHz



b. *Manchesterkodet datasignal.*

Et Manchersterkodet datasignal har alltid en transisjon mitt i biten. Da vil den minste tiden mellom to transisjoner være halvparten av tiden for en bit. Hvis vi ser på figuren over, vil tiden for en bit være 2·t. Da vil kravet til båndbredde bli lik bithastigheten, som da blir **100 MHz**.

c) Anta at du får følgende info etter en ping kommando:

Pinging www.stanford.edu [2607:f6d0:0:925a::ab43:d7c8] with 32 bytes of data:
Reply from 2607:f6d0:0:925a::ab43:d7c8: time=184ms
Reply from 2607:f6d0:0:925a::ab43:d7c8: time=183ms
Reply from 2607:f6d0:0:925a::ab43:d7c8: time=183ms
Reply from 2607:f6d0:0:925a::ab43:d7c8: time=183ms

Ping statistics for 2607:f6d0:0:925a::ab43:d7c8:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 183ms, Maximum = 184ms, Average = 183ms

Anta at datahastigheten (den «fysiske») du har til www.stanford.edu er på 100 Mbit/s. (Altså $100 \cdot 10^6$ bit/s). Du skal bruke idle RQ overføring, med en pakkestørrelse på 1024 Byte. Hva blir den effektive datahastigheten på overføringen, - altså den hastigheten du som bruker opplever?

Må først regne ut effektiviteten U

I dataene ser vi at RTT = 183 ms.

Antall bit i pakken er L= 1024 [byte] · 8 [bit/byte] = 8192 [bit]
Bithastigheten R = 100 [Mbit/s] = 100 · 10^6 [bit/s]

$$U = \frac{t_{trans}}{RTT + t_{trans}} = \frac{L/R}{RTT + L/R} = \frac{8192/100 \cdot 10^6}{0{,}183 + 8192/100 \cdot 10^6} = \frac{8192}{18{,}3 \cdot 10^6 + 8192} = \frac{8192}{18308192} = 4{,}5 \cdot 10^{-4}$$

Den effektive datahastigheten blir nå: 100 Mbit/s · $4{,}5 \cdot 10^{-4}$ = 45000 bit/ = **45,0 Kbit/s**

d) Hvordan kan den hastigheten økes?

Man kan øke pakkestørrelsen. I uttrykket over blir da L større.

e) Hva er (den hele) IPv6 adressen til www.stanford.edu? Du skal altså skrive den hele IPv6 adressen, ikke den forkortede versjonen som er angitt over (i oppgave c)?

IPv6 adressen til www.stanford.edu er 2607:f6d0:0:925a::ab43:d7c8, skrevet på forkortet form. Vi ser at det er en plass hvor det er to kolon etter hverandre, slik :: . Her er det 0'ere som skal inn. Så mange som er nødvendig for å få en hel adresse. Det er også en plass hvor det står en 0'er. Her skal det inn med fire 0'ere. Da blir hele den 128 bit (16 byte) lange adressen:

2607:f6d0:0000:925a:0000:0000:ab43:d7c8

f) His du gir en tracert-kommando kan du f.eks. få denne info:

Tracing route to www.princeton.edu [128.112.132.86]
over a maximum of 30 hops:

1	<1 ms	<1 ms	<1 ms	c6500-h-1.hiof.no [158.39.165.3]
2	<1 ms	<1 ms	<1 ms	halden-gw3.uninett.no [128.39.46.129]
3	2 ms	2 ms	2 ms	ifi2-gw.uninett.no [128.39.254.241]
4	2 ms	2 ms	2 ms	oslo-gw1.uninett.no [128.39.230.117]
5	6 ms	2 ms	2 ms	oslo-gw.uninett.no [128.39.255.225]
6	9 ms	9 ms	9 ms	se-tug.nordu.net [109.105.102.21]
7	111 ms	111 ms	111 ms	us-man.nordu.net [109.105.97.45]
8	112 ms	122 ms	107 ms	xe-2-3-0.118.rtr.newy32aoa.net.internet2.edu [109.105.98.10]
9	117 ms	113 ms	113 ms	local.internet2.magpi.net [216.27.100.53]
10	115 ms	115 ms	115 ms	remote1.princeton.magpi.net [216.27.98.114]
11	133 ms	116 ms	116 ms	core-87-router.princeton.edu [128.112.12.130]
12	109 ms	109 ms	109 ms	foundry-lb-vip-30.princeton.edu [128.112.132.86]

Trace complete.

Her ser vi at IP-adressen til hvert punkt i forbindelsen blir listet opp. Forklar hvordan tracert får tak i IP-adressen til hvert punkt i forbindelsen.

Tracert sender ut en serie med ICMP pakker, hvor TTL feltet i IP hodet blir satt til en bestemt verdi. I hvert hopp, altså i hver ruter på veien, blir TTL minsket med 1. Den ruteren som gjør at TTL=0, vil ikke sende ICMP pakken videre, men i stedet sende en ICMP pakke tilbake til avsender, med info om hvor TTL=0, dvs hvor pakken stoppet. Denne info er altså IP adressen til ruteren. Tracert setter først TTL = 1. Da vil den første ruteren gjøre at TTL=0, og den vil sende ICMP-pakke med IP nummeret til der hvor pakken stoppet, som nå er i den første ruteren. Neste gang sendes en ICMP pakke med TTL=2, Da vil den andre ruteren på veien stoppe pakken, og sende info tilbake til senderen. Slik fortsetter det helt til siste hopp er til mottageren. – Så tracert får tak i denne info ved å sende pakker hvor TTL starter på 1 og øker med 1 for hver gang.

g) Vi er nå i en tid hvor det brukes både IPv4 og IPv6. Forklar først hovedforskjellene mellom IPv4 og IPv6, og beskriv deretter hvordan det er mulig å bruke begge disse to samtidig, i Internet.

Det er flere hovedforskjeller:

1) Adressefeltet er mye større i IPv6. Det er det 128 bit, mens i IPv4 er det 32 bit. Det er også flere adressetyper i IPv6. I tillegg til unicast adresse (en enkelt enhet), slik som også IPv4 har, har IPv6 anycast og multicast adresse. Anycast tas imot av den første, den nærmeste, av et sett host's som utgjør anycast adressen. En multicast adresse blir sendt til alle som er en del av multicast adressen. Det kan sammenlignes med broadcast adressen i IPv4.

2) IPv6 hodet har sløyfet en del felt, som gjør at prosesseringen på info i IPv6 hodet går raskere. Header Checksum er fjernet, som gjør at lag3-utstyr slipper å foreta beregningen på header checksum. IPv6 har sløyfet info om lengden på hodet, og muligheten av "option" i hodet. I stedet legger IPv6 opp til å bruke flere etterfølgende hoder. Denne metode gjør prosesseringen på et standardhode raskere.

Det er mulig å bruke begge typene i internet. Den ene måten er å bruke"tunneling" for å sende en IPv6 pakke igjennom et IPv4 nettverk. Hele IPv6 pakka pakkes da inn i datadelen av IPv4 pakka.
Det er også mulig å bruke «dual stack». Da vil sender, eller mottager, kunne bruke både IPv4 og IPv6. Senderen eller mottageren velger da den type som er i bruk.

Oppgave 2

a) Hva er forskjellene på en switch og en router?

En switch virker på lag 2. Dvs at den leser MAC adressen på en pakke som kommer inn, og sender den ut på kun den porten hvor den MAC adressen befinner seg.

En router virker på lag 3 (i tillegg). Dvs at den også leser IP adressen, og ruter pakken ut på riktig port. Når pakken sendes ut på porten, setter den også på riktig MAC adresse til nærmeste mottager.

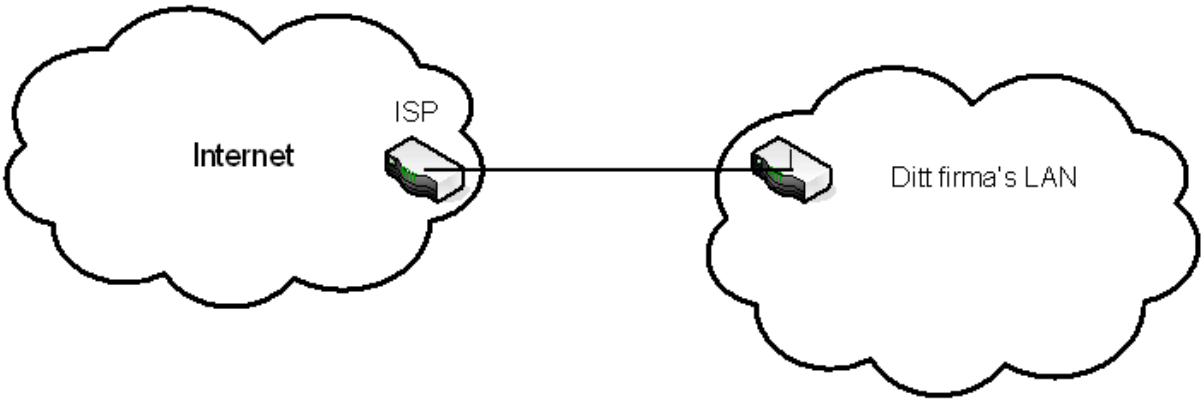
b) *Hva er hovedforskjellene mellom UDP og TCP? Nevn også litt om til hvilke typer applikasjoner de forskjellige passer best, og hvorfor.*

UDP står for User Datagram Protocol og er en ”connectionless” protokoll. Det vil si at forbindelsen ikke blir satt opp først. Det er data i første pakke. UDP er en usikker protokoll da den ikke gir noen respons tilbake om pakka har kommet fram, og veien som den tar kan være forskjellig for hver pakke som går. UDP inneholder lite ”overhead”, og det går da raskere å sende data med UDP enn TCP

TCP står for Transmission Control Protocol og er en ”connected oriented” protokoll. Det vil si at forbindelsen blir satt opp før data sendes. Alle pakkene som sendes i en melding følger samme veien. Pakkene kommer også fram i riktig rekkefølge og det er mulighet for retransmisjon ved feil. TCP-hodet inneholder nok info til å styre alt dette. Det blir da forholdsvis mye ”overhead”, som gjør at det går tregere med TCP enn UDP.

TCP er egnet der hvor det er viktig at alle pakker kommer riktig fram, selv om det ikke går så hurtig. F.eks. hvis en webside blir overført, eller i en forbindelse med banken. UDP er egnet der hvor det ikke er så kritisk hvis en pakke skulle bli borte, men det er viktig at forbindelsen er rask. F.eks. der hvor lyd og/eller film blir overført.

Anta at du har startet et firma, og ønsker å ha et eget datanett til det firmaet. I dette datanettet skal alle host være direkte tilknyttet Internet, via en ruter (uten NAT). Av en internet-leverandør (ISP) får du nettnummeret, med maske: 132.65.40.00/21.



c) *Hvor mange host kan du ha på dette nett?*

Antall host bestemmes av antall 0’ere i masken, dvs antall bit i host-delen av adressen. Her er det 32-21 = 11 bit i hostdelen av adressen. Det gir:

2^{11}-2 = 2048 - 2 = **2046**

Vi tar bort 2, fordi et er til nettnummeret og et er til broadcastnummeret. Ingen host kan få disse nummer.

d) *Hva blir broadcast-adressen på dette nett?*

Broadcastadressen er der hvor hostdelen av adressen er bare 1'ere:

Vi har nettnummer 132.65.40.00/21. Vi ser på de to siste bytene i den adressen, og setter bare 1'ere i hostdelen: (**Fet skrift** indikerer hostbit)
00101**111**.**11111111** = 47.255. Det gir broadcastadressen = **<u>132.65.47.255</u>**

Nå skal ditt firma utvide med en fem nye avdeling. I begynnelsen ligger alle de seks avdelingene i samme hus. Du synes det er best å la disse avdelinger få hvert sitt datanett. Du må da dele ditt datanett i like store subnett.

e) Hva blir nettadressene til disse seks subnett, og hva blir nettmasken?

For å få seks subnett, må det brukes tre bit av nettnummeret. Fordi 2^3-2=6 nett.
Det blir altså tre nye 1'ere i nettmasken, i forhold til det vi hadde fra før. Nettmasken til disse subnett blir da: 255.255.255.00, eller /24, fordi det var /21. Tre bit til blir /24

Vi bruker av disse tre bit for å lage subnettnummer. Vi ser på de to siste byte:
*00101*000**.**00000000 -> 132.65.40.00 - Hovednettet det subnettes fra.
*00101***001.**00000000 -> 132.65.41. 00/24 – subnett 1
*00101***010.**00000000 -> 132.65.42. 00/24 – subnett 2
*00101***011.**00000000 -> 132.65.43. 00/24 – subnett 3
*00101***100.**00000000 -> 132.65.44.00/24 – subnett 4
*00101***101.**00000000 -> 132.65.45. 00/24 – subnett 5
*00101***110.**00000000 -> 132.65.46. 00/24 – subnett 6

f) Hvor mange host kan det være på hvert av disse subnett?

/24 gir 24 stk 1'ere i masken, som gir 32-24=8 bit til host. Antall host blir da

2^8-2 = **<u>254 host</u>**

g) Hva blir laveste og høyeste IP adresse på en host på et av disse subnett? (Du velger selv hvilket subnett du ønsker å angi det på)

Den laveste IP adressen er en over nettnummeret. Den høyeste adressen er en under broadcastadressen.

Hvis vi ser på subnett 1, så har den nettadressen:
*00101***001.**00000000 -> 132.65.41.00/24
En over nettadressen er:
*00101***001.**00000001 -> 132.65.41.01
En under den broacastadressen er:
*00101***001.**11111110 -> 132.65.41.254

h) *Nå skal fire av disse subnett flyttes til fire andre byer. Du må da sette opp en punkt-til-punkt forbindelse til hver av disse fire subnett. Hvilket subnettnummer og maske, vil du gi disse fire punkt-til-punkt forbindelsene?*

En punkt til punkt forbindelse trenger bare to bit til host, slik at masken på disse punkt til punkt forbindelsene blir 255.255.255.252, eller /30

Vi bør gi disse to nett IP nummer som ligger helt i ytterkant av nummerområdet. Det gir

*00101*000**.00000100** -> 132.65.40.04 *0010111***1.111110**00 -> 132.65.47.248/30
*00101*000**.00001000** -> 132.65.40.08 *0010111***1.111010**00 -> 132.65.47.244/30

Oppgave 3

a) *Hva brukes en DHCP server til? Forklar litt om virkemåten*

DHCP står for Dynamic Host Configuration Protocol, og den brukes for å dele ut IP-adresse til en maskin som forespør om en IP-adresse.

Det finnes en DHCP server som dekker et eller flere subnett. Denne har et sett med IP-adresser som den kan dele ut fra, til maskiner som forespør om en IP-adresse.

Et eksempel på DHCP:
En maskin som blir slått på, og som er satt til å forespørre en DHCP-server om sin IP-adresse, sender en DHCP DISCOVER pakke til DHCP-serveren. Når denne kommer fram til DHCP-serveren, returnerer den IP-adresse i en DHCP-OFFER pakke. Hvis maskinen som mottar denne, godtar dette nummer, sender den en DHCPREQUEST pakke tilbake til DHCP serveren. DHCP-serveren sender tilbake en DHCPACK pakke. Samtidig settes en timer i gang. Enhver maskin må "oppdatere" sin IP-adresse jevnlig, ved å sende en melding til DHCP-serveren. Hvis DHCP-serveren ikke mottar en slik oppdatering innen timeren er gått ut, vil den frigjøre IP-adressen, som da kan deles ut til en annen maskin som forespør om en IP-adresse.

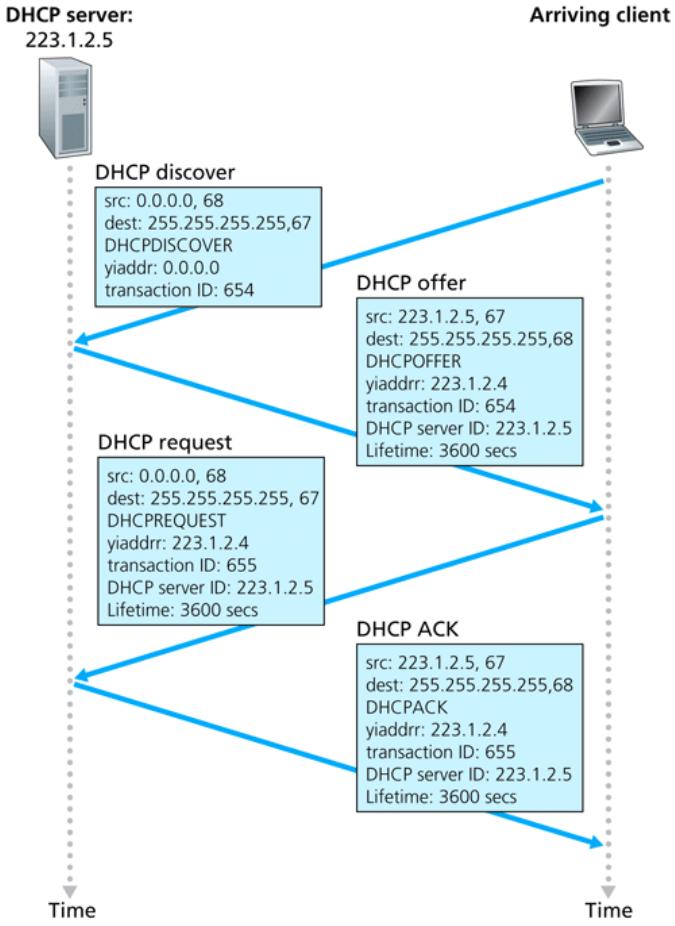


Figure 4.21 ♦ DHCP client-server interaction

.

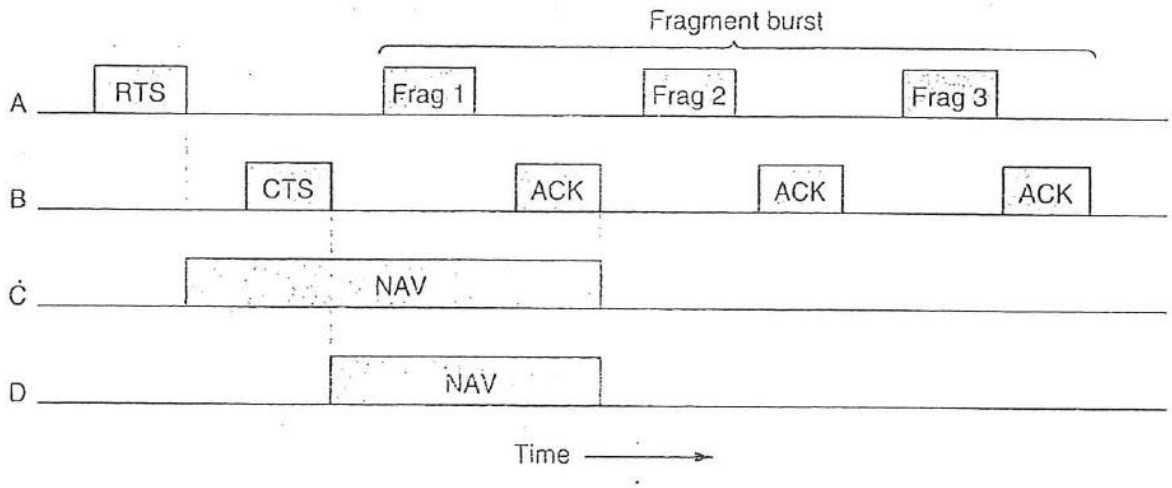
b) Forklar hvordan aksessmetoden CSMA/CA virker.

Carrier **S**ense **M**ultiple **A**ccess med **C**ollision **A**voidance virker slik: En stasjon som ønsker å sende, må først lytte i en gitt tid, for å høre om det er noen aktivitet på kanalen. Hvis det ikke er noen aktivitet på kanalen, sender stasjonen først en kort pakke, for å fortelle alle andre om at den skal sende data. Deretter sender den datapakka. CSMA/CA kan også bruke RTS og CTS. Stasjonen som skal sende, sender da først en RTS pakke, hvor det er info om hvor lang tid sendingen vil pågå. Det er for at andre stasjoner ikke skal sende i denne perioden. Den mottagende stasjon svarer på RTS med en CTS pakke, med samme info om tid på sendingen. De stasjonene som kun hører mottager, vil dermed også ikke sende i den perioden. Når sende stasjonen har mottat CTS, sende den datapakka.

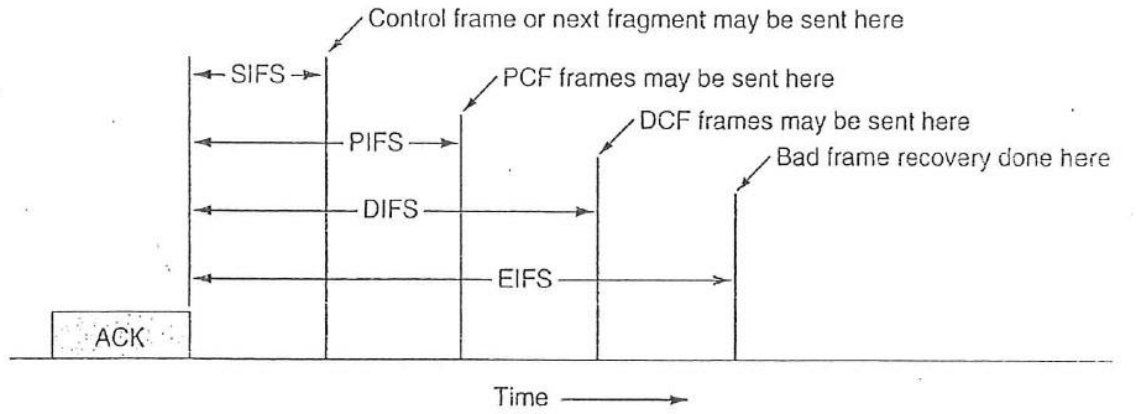
c) I WiFi kan både DCF mode og PCF mode brukes. Forklar forskjellene på disse to, og hvordan begge kan eksistere i samme nett. Hvordan det løses.

PCF står for Point Coordination Function. Her er det en hovedenhet som styrer overføringen. Denne hovedenheten spør hver enkel host om den har noe å sende. Ingen host kan sende uten at den er forespurt.

DCF står for Distributed Coordination Function
En host som skal sende data lytter først for å høre om det går datatrafikk, om mediet er ledig. Hvis mediet er ledig, kan den sende. Den sender da ofte en RTS pakke først. Her ligger det bla. info om hvor lang tid den skal bruke på sendingen. Alle de som hører dette vil da avstå fra å sende i denne tiden. Det gjøres ved å sette en NAV (Network Allocation Vector). – Mottager svarer med å sende en CTS, men samme info om tid. De som da ikke hørte RTS sendingen, vil da høre CTS sendingen. Så starter sendingen, og hver pakke blir akseptert ved en ACL pakke tilbake. Når sendingen er ferdig, kan andre starte sending.



For at ikke andre skal starte sending innimellom pakkene som går, er det innført forskjellige ventetider etter en pakke er sendt.



En pågående sending har første prioritert, dvs at den sender neste pakke innen tiden SIFS. Hvis det også er et PCF system som sender, har det systemet mulighet til å ta en sending, etter at en pågående DCF sending er ferdig. PCF starter da sending innen tiden PIFS. Hvis det ikke er noen PCF sending, kan neste DCF sending starte etter tiden DIFS. Ved bruk at disse forskjellige ventetidene, kan både DCF og PCF eksistere i samme nett.

d) *Du skal dimensjonere et fiberoptisk anlegg, med bruk av SM fiber. Senderen har en innkoblet effekt i fiberen på + 3,0dBm. Fiberkabelen har en dempning på 0,3 dB/km, og en dispersjon på 3,5 ps/(nm·km).. Lyskilden (laseren) har en spektral båndbredde på 2,0 nm. Det er ingen skjøter, og ingen kontakter. Du kan regne med innkoblingstap ved mottageren på 1,0 dB. Hva blir maksimal fiberstrekning når mottageren har en følsomhet på – 40,0 dBm, og det skal sendes data med en (ukodet) bithastighet på 10 Gbit/s?*

Regner først på effekten:
Setter systemmarginen til 3,0 dB (mellom 3,0 og 7,0)
$P_{inn} - P_{fiber} - P_{innk} - P_{syst} = P_m$
+3,0 - x· 0,3 – 1,0 – 3,0 = - 40,0
x· 0,3 = 40,0 -1,0 – 3,0 + 3,0 = 39,0
x = 130,0 km

Ved bruk av systemmargin på 7,0 dB blir det
+3,0 - x· 0,3 – 1,0 – 7,0 = - 40,0
x· 0,3 = 40,0 -1,0 – 7,0 + 3,0 = 35,0
x = 116,7 km

Regner så på båndbredden:
Med en ukodet bithastighet på 10,0 Gbit/s, kreves det en båndbredde på 5,0 GHz.
Det gir en maks dispersjon på: $\tau = 0{,}44/(5{,}0 \cdot 10^9) = 88{,}0 \cdot 10^{-12}$ s
Den dispersjonen fås ved y km:
$88 \cdot 10^{-12} = 3{,}5 \cdot y \cdot 2{,}0 \cdot 10^{-12}$
$y = 88 \cdot 10^{-12} / 7{,}0 \cdot 10^{-12} = 12{,}6$ km

Maksimal fiberstrekning blir **12,6 km**

e) Anta at du har et ZigBee wireless nett. Hva er typiske bruksområder for et slikt nett? Beskriv også hvordan det er bygd opp, og virkemåten.

ZigBee er for der det skal måles eller styres, hvor det er små datamengder som skal sendes. Det kan f.eks brukes til å automatisere et hus, eller hvor man skal ta målinger på mange forskjellige steder.

ZigBee bruker tre forskjellige noder: ZC(coordinator), ZR(router) og ZED(end device). I et nett finnes det kun en ZC. En ZR kan virke som en ruter som sender trafikken videre i nettverket, i tillegg til at den kan være en aktiv node. ZR kan også bli en ZC. ZED er den enkleste enhet, som kun kan sende eller motta data. Den kan ikke rute trafikk videre. ZC kalles også PAN coordinator. ZR kalles FFD(Full Function Device) og ZED kalles RFD(Reduced Function Device).

Det kan være forskjellige typer nettverkstypologier, - mesh, star, cluster.

Enhver node har 64 bit adresse, som er unik for den noden. Ingen andre noder har den adressen. Det finnes også en kort adresse, som er på 16 bit, og som brukes i et nettverk. Så et ZigBee nettverk kan da ha litt over 65000 noder.

VEDLEGG

$$B = \frac{0{,}44}{\tau}$$

$$U = \frac{L/R}{RTT + L/R}$$

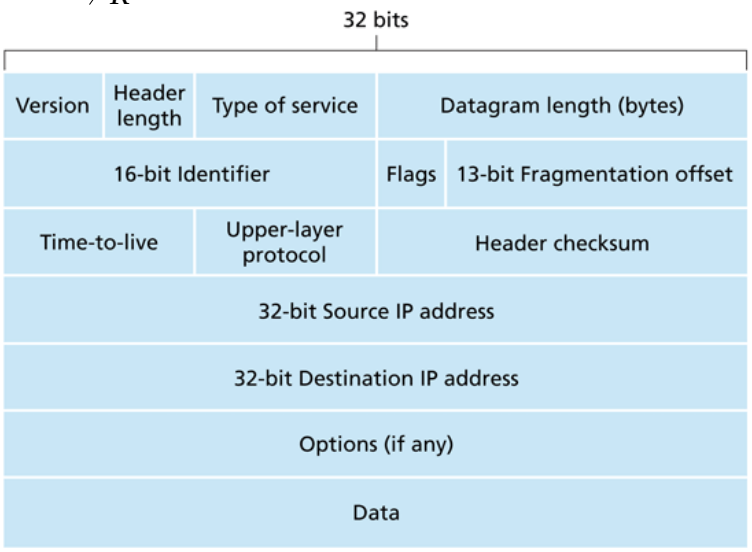


Figure 4.13 ♦ IPv4 datagram format

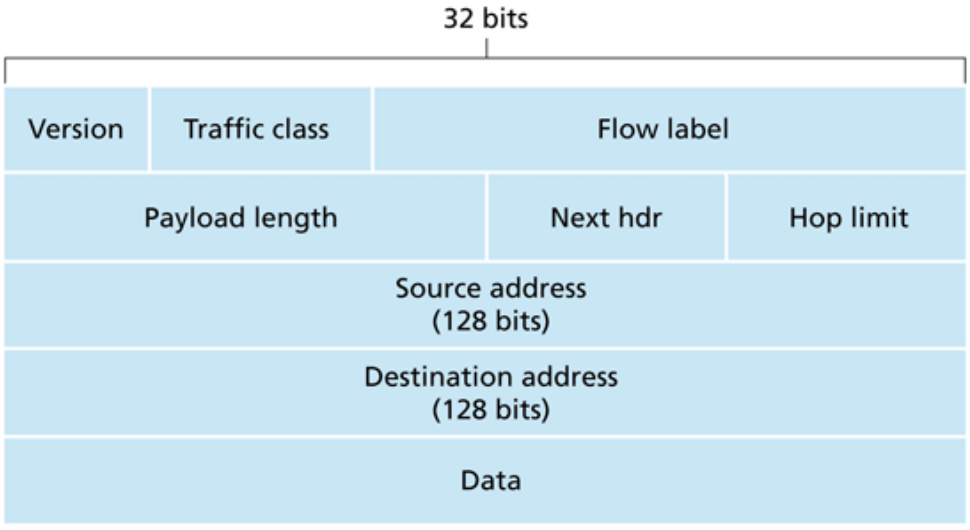


Figure 4.24 ♦ IPv6 datagram format

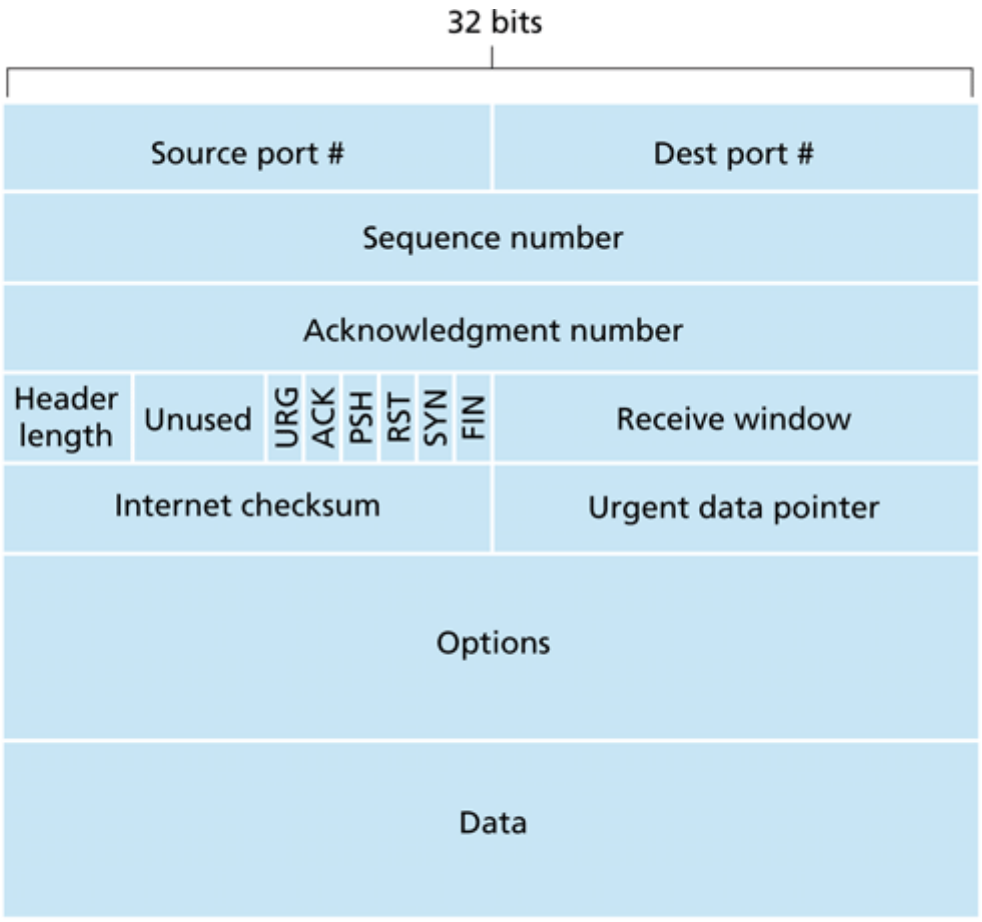


Figure 3.29 ♦ TCP segment structure

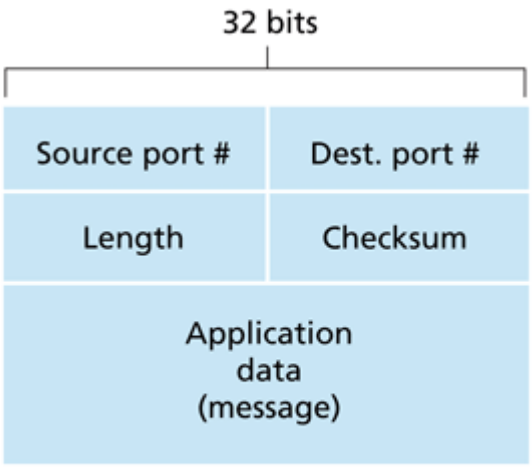


Figure 3.7 ♦ UDP segment structure