NTNU
Kunnskap for en bedre verden

KANDIDAT

10125

PRØVE

TTM4100 1 Kommunikasjon - Tjenester og nett

Emnekode	TTM4100
Vurderingsform	Hjemmeeksamen
Starttid	14.05.2021 07:00
Sluttid	14.05.2021 11:00
Sensurfrist	07.06.2021 21:59
PDF opprettet	28.05.2021 10:11

Exam front page

Oppgave	Tittel	Status	Poeng	Oppgavetype
□	Cover page			Dokument

Application layer (and General)

Oppgave	Tittel	Status	Poeng	Oppgavetype
1.1	Protocol layers	Besvart	Rettes manuelt	Langsvar
1.2	World Wide Web	Besvart	Rettes manuelt	Langsvar
1.3	Delay in networks	Besvart	Rettes manuelt	Langsvar
1.4	E-mail security	Besvart	Rettes manuelt	Langsvar

Transport layer

Oppgave	Tittel	Status	Poeng	Oppgavetype
2.1	General functionality	Besvart	Rettes manuelt	Langsvar
2.2	Sequence numbers and ACK	Besvart	Rettes manuelt	Langsvar
2.3	TCP congestion control	Besvart	Rettes manuelt	Langsvar
2.4	SSL	Besvart	Rettes manuelt	Langsvar

Networking layer

Oppgave	Tittel	Status	Poeng	Oppgavetype
3.1	IPv4 fragmenting	Besvart	Rettes manuelt	Langsvar
3.2	IPv4 versus IPv6	Besvart	Rettes manuelt	Langsvar
3.3	IPv4 addressing	Besvart	Rettes manuelt	Langsvar

Link layer and Mobile/Wireless

Oppgave	Tittel	Status	Poeng	Oppgavetype
4.1	General functionality	Besvart	Rettes manuelt	Langsvar
4.2	2-dim parity versus CRC	Besvart	Rettes manuelt	Langsvar
4.3	ARP	Besvart	Rettes manuelt	Langsvar
4.4	W-LAN	Besvart	Rettes manuelt	Langsvar

1.1 Protocol layers

Gi en oversikt over protokoll-lags modellen som brukes for kommunikasjon over Internett og forklar spesielt hva som menes med begrepet "innkapsling" ("encapsulation") i denne sammenhengen.

Skriv ditt svar her

Protokoll-lags modellen gir en struktur for hvilke protokoller som tilhører hvilke lag, og består av fem lag; Applikasjons-, Transport-, Nettverks-, Link- og det fysiske-laget. Protokollene bruker dataen fra underliggende lag til å gjøre opp den fullstendige pakken som blir sendt.

Ved innkapsling menes det å "pakke" inn data fra en pakke inn i en annen. Dette blir gjort ved at en overordnet pakke putter pakken inn i sin egen "header". Et eksempel her er IP protokollen(Nettverks-protokoll)som putter TCP-pakken(transport laget) inn i payload i sin egen header. Dette gjøres ofte flere ganger.

Maks poeng: 6

1.2 World Wide Web

Forklar med egne ord (på høyt nivå) kommunikasjon over World Wide Web (WWW). (Stikkord: Protokoll brukt og egenskapene til denne; caching; cookies).

Skriv ditt svar her

I WWW blir som oftest HTTP og HTTPS protokollene brukt. HTTPS er HTTP med SSL(Nå TSL) kryptering.
HTTP pakken finner man i applikasjons-laget og blir innkapslet i TCP pakker.

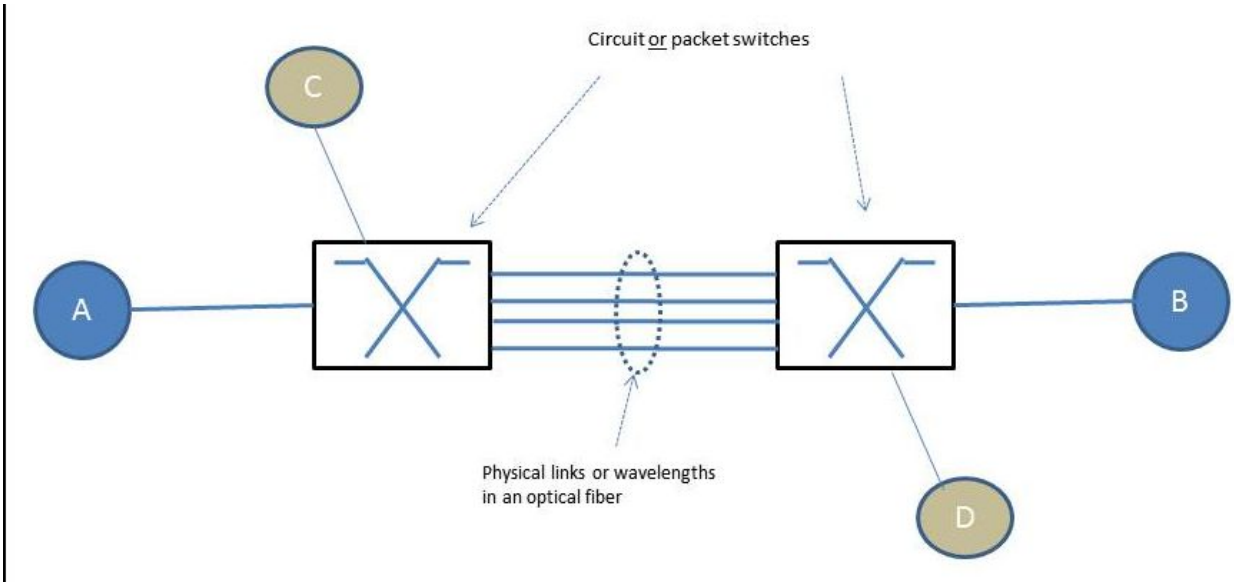
Det finnes ulike typer HTTP forespørsler, GET, POST, PUT og DELETE. De mest vanlige er GET og POST. GET brukes for å hente ut "bodyen" til en nettside, mens POST ofte blir brukt for å endre på informasjon på en nettside. Når man utfører en HTTP forespørsel vil serveren svare med en kode, 200 for ok, 404 for side ikke funnet, 401 for uatorisert, 500 for serverfeil osv. Hvis man har gjort en GET forespørsel, og fått 200 OK tilbake, vil da serveren sende deg innholdet av siden.

Siden HTTP er det man kaller tilstandsløs, men ofte vil man kune lagre informasjon fra HTTP forespørslene. Da kan man da bruker caching og cookies.
Cookies blir brukt av serveren for å spore brukerne sine. Serveren kan be om brukeren sin cookie fil, for å for eksempel lagre handlekurven på nettbutikken uten å være logget inn.
Caching er et slags mellomlag mellom serveren og klienten. Når en klient utfører en forespørsel til en server, vil svaret først gå gjennom cache, og kan lagre innholdet. Dette gjør at du slipper å laste inn nye requests hvis ingenting på nettsiden har endret seg.

Maks poeng: 7

1.3 Delay in networks

Se figuren nedenfor.



Anta at kommunikasjon mellom A og B i nettet kan være enten pakkesvitsjet (basert på "store-and-forward" prinsippet) eller linjesvitsjet. Anta at det allerede er satt opp en forbindelse mellom A og B. Det er også andre aktive forbindelser i nettet, f.eks. mellom C og D vist i figuren.

Gjør rede for de ulike bidragene til forsinkelse gjennom nettet når en bruker
a) linjesvitsjing for en informasjonsenhet som sendes fra A til B.
b) pakkesvitsjing for en informasjonsenhet som sendes fra A til B.

Skriv ditt svar her

a) I linjesvitsjing er hele linjen kun reservert for sender og mottaker, det vil si at siden man reserverer en hel linje, vil man bruke opp svært mye av båndbredden til nettverket. Når linjen er reservert har man ingen forsinkelser siden linjesvitsjing garanterer en fast rate.

b) Derimot pakkesvitsjing har flere årsaker til forsinkelser i nettet:
- Prosessering-forsinkelse: Det tar tid å undersøke headerne i hver pakke, same at man må håndtere ting som CRC og checksummer.

- Kø-forsinkelse: Hvis det er for mange som sender pakker på et nettverk enn mottakeren kan prosessere, vil det oppstå en kø. Lengden på kø-forsinkelsen avhenger av hvor mange pakker som ble sendt tidligere, men dette kan omgåes ved prioritets køer.

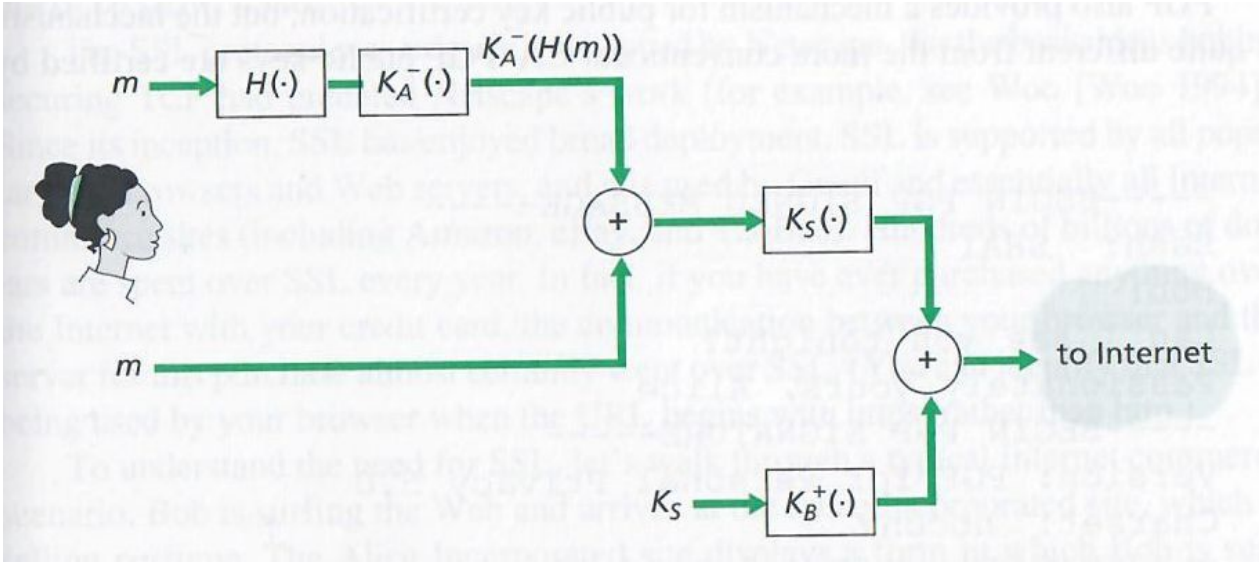
- Sendings-forsinkelse: Det tar tid for en node å sende en pakke, denne tiden er definert som L/R hvor L er lengden av pakken i bits og R er sendingraten i linken fra A til B.

-Link-forsinkelse (propagation delay): Dette er forsinkelsen som kommer av forsinkelser i medium pakken blir sendt gjennom. Fiber, altså optiske nettverk, vil hastigheten på pakken nærme seg lysets hastighet. Tid = hastighet*lengde.

Maks poeng: 7

1.4 E-mail security

Se figuren nedenfor som illustrerer (deler av) en prinsipiell implementasjon av sikker e-post utveksling.



Forklar med egne ord hvilke operasjoner som utføres i figuren og hva som oppnås med disse. Forklar også hvilke operasjoner som må legges til for å realisere sikker e-post utveksling ende-til-ende.

Skriv ditt svar her

I figuren:

Først generer senderen en tilfeldig symmetriske nøkkel(Ks), deretter blir Ks kryptert med mottakeren sin offentlige nøkkel(K^+_B). Dette gjør at kun mottakeren kan dekryptere den symmetriske nøkkelen med sin egen private nøkkel (K_B^-). Deretter sender senderen beskjeden inn i en hash-funskjon(H), som er en en til mange funskjon, det er altså umulig å finne ut hva den orginale beskjeden var. En hash-funskjon gjør også beskjeden om til en fast-lengde. Deretter krypterer senderen den hashete meldingen(H(m) med sin egen private nøkkel(K_a^-). Ved å gjøre dette oppnås det at de med senderen sin offentlige nøkkel kan dekryptere meldingen.

Den krypterte og hashete meldingen blir så kryptert med den symmetriske nøkkelen K_s, og sendt sammen med den krypterte symmetriske nøkkelen($K^+_B(K_s)$).

Når mottakeren da motar pakken, kan den finne ut hva K_s er ved å dekryptere $K^+_B(K_s)$ med sin egen private nøkkel K_B^-. Da kan mottakeren dekryptere resterende melding med K_s. Mottakeren vil da dekryptere den hashete meldingen med senderen sin offentlige nøkkel(K^+_A). Og til slutt hasher plaintext meldingen(m), og sammenligner den hashete meldingen senderen sendte med sin egen hashete meldingen. Hvis de er like kan man være sikker på at ingen har endret på meldingen. Denne serien med kryptering gir oss da konfidensialitet, autentifikasjon, og meldingsintegritet. Altså at ingen har kunnet lese hva orignal-beskjedenen var, motakeren kan verifisere at meldingen kom fra den senderen sa den var og ingen har tuklet med meldingen.

I tillegg for å realisere sikker ende til ende utveksling kan man verifisere de offentlige nøkklene ved hjelp av CA.

Maks poeng: 7

2.1 General functionality

Gi en oversikt over transportlaget for kommunikasjon over Internet. (Stikkord: hovedoppgaver/funksjoner, protokoll(er) brukt, hvor i nettet det er til stede).

Skriv ditt svar her

Hovedoppgavene til transport-laget er å innkapsle dataen som skal bli sendt slik at den vet hvor den skal når den kommer til mottakeren. Den sier altså ikke noe om hva ruteren skal gjøre, det er det nettverks-laget som gjør.

Det er hovedsaklig to protokoller i transport-laget; UDP og TCP.

UDP er en "bare minimum" pakke, altså den inneholder minst mulig informasjon for at pakken skal bli sendt. Derimot TCP inneholder mer informasjon om sender og mottaker, og medfører også funksjonalitet som, flyt-kontroll, overbelastnings-kontroll. TCP har også det man kaller pålitelig dataoverføring, ved at den sender og motar ACKS på sendte pakker.

Transportlaget sier altså noe om hvilket ende-system den skal til.
Transport-laget er implementert i ende-systemene, men ikke i rutere.

Maks poeng: 6

2.2 Sequence numbers and ACK

Anta en etablert TCP forbindelse mellom vertene A og B. På et gitt tidspunkt i kommunikasjonen har A mottatt fra B alle data opp til og med byte 433. Anta at Vert B deretter sender to segmenter med data til Vert A uten opphold mellom dem («back-to-back»). Første og andre segment inneholder henholdsvis 15 og 50 bytes med data. I det første segmentet er sekvensnummeret 434, kildeportnummeret er 495, og destinasjonsportnummeret er 344. Vert A sender alltid en kvittering når den mottar et segment fra Vert B.

a) Hva er sekvensnummeret, kildeportnummeret og destinasjonsportnummeret i det andre segmentet som sendes fra Vert B til Vert A?

b) For samme situasjon som beskrevet over: Hvis det andre segmentet sendt ankommer til vert A før det første segmentet, hva er kvitteringsnummeret sendt fra A til B for dette segmentet?

Skriv ditt svar her

a)
Kildeportnummer: 495
Destinasjonsportnummer: 344
Sekvensnummer: 449

b)
ACK#: 433, fordi den ikke har mottatt den første pakken enda

Maks poeng: 7

2.3 TCP congestion control

Gi en oversikt over hvordan overbelastningkontroll ("congestion control") er implementert i TCP protokollen. (Stikkord: tre hovedmekanismer; hovedformål og funksjonalitet for hver av disse).

Skriv ditt svar her

De tre hovedmekanismene man har for overbelastiningskontroll i TCP protokollen er: Slow start, Congestion avoidance og Fast recovery.

Slow start:
Når man øker sendingsraten på en TCP pakke vil man gjerne øke den med 1 MSS inntil man får pakketap, denne betyr ofte at i starten har man en veldig lav sendingsrate og det er lite effektivt. Med at man øker sendingsraten eksponensielt til man når SSThreshold, altså en linje som ofte er cwnd/2. Målet med slow start er å raskere utnytte den båndbredden man har.

Congestion avoidance:
Denne funksjonaliteten vet at når sendingsrate nærmer seg punktet det tidligere var pakketap, begynner man å roe ned økningen på sendingsraten. her begynner man kun å legge til 1 MSS for hvert ACK motatt. Målet med congestion avoidance er å prøve å holde seg så høy sendingsrate det går ann over lengre tid, slik at man utnytter kapasiteten maks.

Fast recovery:
Når sendereren merker pakketap vil den ofte halvere sendinggsraten sin og vil da gå inn i slow start igjen. Formålet med fast recovery er å komme seg raskt opp til høy sendingsrate så man utnytter båndbredden.

Maks poeng: 7

2.4 SSL

Gi en oversikt over hensikten med og implementasjonen av Secure Socket Layer (SSL).
(Stikkord: tre faser; hva oppnås i hver fase; "nonces"; MAC).

Skriv ditt svar her

De tre fasene: handshake, key derivation og data transfer.

Handshakefasen brukes som en sikkert måte at både sender og mottaker skal vite hva master nøkkelen er (M_S).

Key derivation fasen brukes for at begge partene skal vite hva alle de fire nøklene er
- K_c = krypterings nøkkel for data sendt fra sender til mottaker
- M_c = MAC nøkkel for data sendt fra sender til mottaker
- K_s = krypterings nøkkel for data sendt fra mottaker til sender
- M_s = MAC nøkkel for data sendt fra mottaker til sender

Data transfer fasen er fasen hvor den faktiske dataen blir sendt, ved hjelp at de fire nøklene partene har generert.

Nonces:
Nonces er en en randomisert generert serie med bits som kun skal bli brukt en gang. Dette er for å unngå såkalte replay-attacks, hvor en inntrenger sender samme pakke om igjen, kun litt modifisert. Mottakeren oppdager da at samme nonce er brukt flere ganger, og autentiserer ikke forespørselen.

MAC:
MACs brukes for å sikre meldingsintegritet. Man sender en MAC sammen med den originale pakken. Da kan mottakeren hashe dataen i pakken og sammeligne MACene for å sikre seg at ingen har tuklet med pakken. For å generere en MAC tar man dataen sammen med annen data som blir sendt i pakken og hasher det. Denne andre dataen kan være timestamps eller nonces. Man kan også hashe CA for å oppnå autentifikasjon.

Maks poeng: 7

3.1 IPv4 fragmenting

a) Hva menes med fragmentering (i Internet protokoll sammenheng) og hvorfor brukes det for IPv4 datagrammer?

b) Hvor blir fragmenter reassemblert (“reassembled”) når IPv4 brukes?

Skriv ditt svar her

a) Fragmentering brukes for å "dele" opp pakker i mindre biter, for å så bli satt sammen igjen når de når mottaker. Siden IP datagrammer kan bli sendt gjennom ulike medier (altså innkapslet i ulike link-lag protokoll) som har egne grenser for for store dataen kan være. Er det nødvendig å dele opp dataen i mindre biter slik at MTUen til link-lags protokollen ikke blir overskridt.

b) Fragmentene blir reassemblert i ende-systemene, altså ikke i ruterne. Dette er for å holde protokollen enkel, og unngå å overbelaste ruteren mer flere oppgaver.

Maks poeng: 7

3.2 IPv4 versus IPv6

Gjør rede for de viktigste forskjellene mellom IP versjon 4 og IP versjon 6.

Skriv ditt svar her

Adresser: IPv4 sine adresser består av kun 32-bits. Dette medfører en mangel av IPv4 adresser i verden, og man må bruke metoder som NAT for å ha nok IPv4 adresser. Derimot IPv6 sine adresser er 128-bits.

Ipv6 headern er 40byte fixed. Dette gjør det letterere å prosessere ipv6 enn ipv4.

Ipv6 har også to nye felt; traffic class, og flow label.

Ipv6 tillater ikke fragmentering, hvis en ruter får en ipv6 pakke som er for stor, dropper den pakken og sender en "for stor pakke" ICMP feilmelding tilbake.

Ipv6 fjerner header checksummen, dette er fordi det er en såpass dårlig checksum at det ikke er nødvendig og blir heller håndtert av lavere nivåer.

Ipv6 fjerner options. Dette er for å gjøre den lettere å prosessere.

Maks poeng: 6

3.3 IPv4 addressing

Anta at en ruter i nettet har følgende CIDR ("classless inter-domain routing") innslag i rutingstabellen:

Address/mask	Next hop
135.46.0.0/22	Interface 0
135.46.128.0/22	Interface 1
192.53.40.0/23	Interface 2
Default	Interface 3

I hvilken retning ut av svitsjen ("Next hop") sendes følgende ankommende IP pakker?
a) 135.46.129.10
b) 135.46.0.14
c) 135.46.48.2
d) 192.53.40.7
e) 192.53.56.7

Skriv ditt svar her

a) Interface 3

b) Interface 0

c) Interface 3

d) Interface 2

e) Interface 3

Maks poeng: 7

4.1 General functionality

Gi en oversikt over linklaget for kommunikasjon over Internet. (Stikkord: hovedoppgaver/funksjoner, protokoll(er) brukt, hvor i nettet det er til stede).

Skriv ditt svar her

Hovedoppgaven til link-laget er å frakte link-lang rammer (frames) mellom noder i nettverket. Dette kan være gjennom kabler som i ethernet eller i luften som ved WiFi.

Avhengig av hvilket medium rammene blir sendt gjennom må man bruke ulike protokoller, for kobber er det hovedsaklig ethernet som er dominerende, og gjennom luften bruker man ofte 802.11 protokollene.

Link-laget tilbyr flere tjenester:
- Flyt-kontroll
- Feil deteksjon og korreksjon gjennom CRC
- Pålitelig data overføring gjennom MAC-protokoller som CSMA/CD og CSMA/CA

Linklaget finner man i alle noder i nettverket og blir implementert i netttverkskort (NICs)

Maks poeng: 6

4.2 2-dim parity versus CRC

To av metodene brukt for feildeteksjon og (i noen grad) korreksjon er 2-dimensjonal paritetssjekk og Cyclic Redundancy Check (CRC). Gjør rede for hva som kan oppnås med disse to metodene. Hva er hovedforskjellen(e) mellom dem?

Skriv ditt svar her

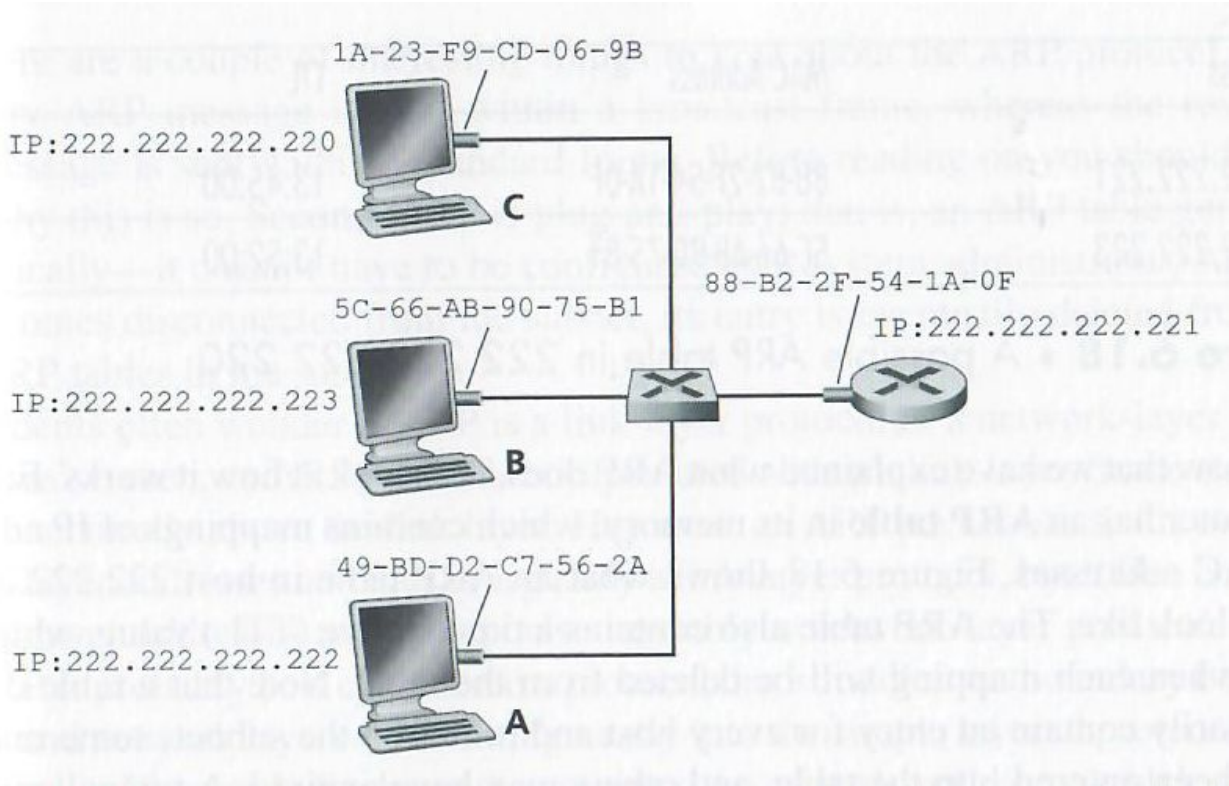
2-dimensjonal paritetsjekk brukes for å oppdage om en bit er flippet, men hvis det er flere bits vil ikke denne paietetssjekken ha vanskeligheter med å oppdage hvor feilen ligger eller i det hele tatt oppdage feilen.

CRC er en mer sofistikert og bedre måte for feildeteksjon. Ved CRC kan man oppdage feil på n odde feil, samt at man kan oppdage sammenhenge feil på lengde r (n redundancy bits) +1. Det er også en sannsynlighet for å oppdage feil som er lengre enn r+1.

Maks poeng: 7

4.3 ARP

Se figuren nedenfor.



a) Hvorfor trenger vi linklagsadresser (MAC adresser) i tillegg når vi allerede har IP-adresser (på nettverkslaget)?

b) Hvorfor har ikke linklagssvitsjen i figuren noen adresser?

c) Hva er ARP og hvorfor er den nødvendig?

Skriv ditt svar her

a) MAC-adresser blir brukt for portabilitet. Det vil si at man kan flytte på enheter som bruker nettverket. De blir brukt for å å sende link-lags rammer fra ett nettverkskort til en annet fysisk koblet nettverkskort på det loakel nettet.

b) Svitsjer har ikke adresser fordi de ikke trenger å kunne nåes. De er "usynlige". Hvis ruteren vil sende en pakke til mottaker A vil ikke ruteren trenge å vite at den skal gjennom en svitsj først.

c)
En ARP (adress resoultion protocol) blir brukt for å identifisere en MAC adresse sammen med en IP-adresse. Hver node i et nettverk har en ARP-tabell som består av <IP-adresse, MAC-adresse, Time to live>.

Maks poeng: 7

4.4 W-LAN

a) MAC protokollen CSMA/CA for trådløse nett basert på standarden 802.11 har definert ulike varianter av "Inter-Frame Spacing" (f.eks. "Short IFS - SIFS" og "Distributed IFS - DIFS") med ulik varighet. Hva oppnås i CSMA/CA ved å la SIFS være kortere enn DIFS?

b) "Request-to-Send (RTS)" og "Clear-to-Send (CTS)" er et opsjonelt tillegg til 802.11 MAC. Hva kan oppnås med denne mekanismen, og i hvilke tilfeller bør den benyttes?

Skriv ditt svar her

a) Ved å la SIFS være kortere enn DIFS vil man kunne unngå at viktige kontroll pakker eller ACKS i å kollidere.

b) Ved å bruke RTS og CTS kan man unngå kollisjoner ved "hidden terminal problem" altså at ikke alle noder på samme nettverk kan se hverandre. Når en host sender en RTS og mottakeren ikke har annen kommunikasjon sender broadcaster den en CTS til hele nettverket, da vet alle noder på hele nettverket at de ikke skal forstyrre. RTS og CTS burde brukes hvis man skal sende lange/store pakker, slik at den ikke blir avbrutt underveis.

Maks poeng: 6