

UNIVERSITETET I OSLO

Det matematisk-naturvitenskapelige fakultet

Eksamen i	INF 240 - Datakommunikasjon
Eksamensdag:	Torsdag 6. juni 2002
Tid for eksamen	9.00 - 15.00
Oppgavesettet er på	4 sider
Vedlegg:	Ingen
Tillatte hjelpemidler	Alle trykte og skrevne hjelpemidler, og kalkulator

Kontroller at oppgavesettet er komplett før du begynner å besvare spørsmålene. Husk å skriv tydelig og lesbart. Gi kortest mulige svar, ikke lange utlegninger. Dersom du på noe punkt finner oppgaveteksten uklar kan du gjøre dine egne presiseringer. Gjør i så fall tydelig rede for disse i besvarelsen din.

Opgave 1. Link nivå problemer

Figuren viser en sender A og en mottaker B koplet sammen ved hjelp av en fysisk link. Vi har følgende vitale data:

- vi benytter en pålitelig "stop-and wait" protokoll
- den nominelle dataraten for linken er 1 Mb/s
- bitfeilraten er 10^{-5}
- størrelsen på en datapakke er 10.000 bit og inkluderer nyttelast og et pakkehode på 100 bit. En kvitteringspakke er på 100 bit
- propagasjonstiden for linken er 0,01 Sek
- prosesseringstiden pr pakke – data eller kvittering – er 10^{-6} Sek



I første omgang ser vi bort fra feil.

1. Hva forstår vi med transmisjonstiden for en pakke? Hvor stor er denne for henholdsvis datapakker og kvitteringer?
2. Hvor lang er tiden fra A starter å sende en datapakke til denne er mottatt av B?
3. Hvor lang tid går det fra A starter å sende en pakke til A kan sende neste pakke?
4. Hvor stor er utnyttelsesgraden av linken?

I en reell situasjon vil ting kunne feile. Og på grunn av varierende belastninger i A og B, vil prosesseringstidene kunne variere mellom 10^{-6} og 10^{-5} Sek.

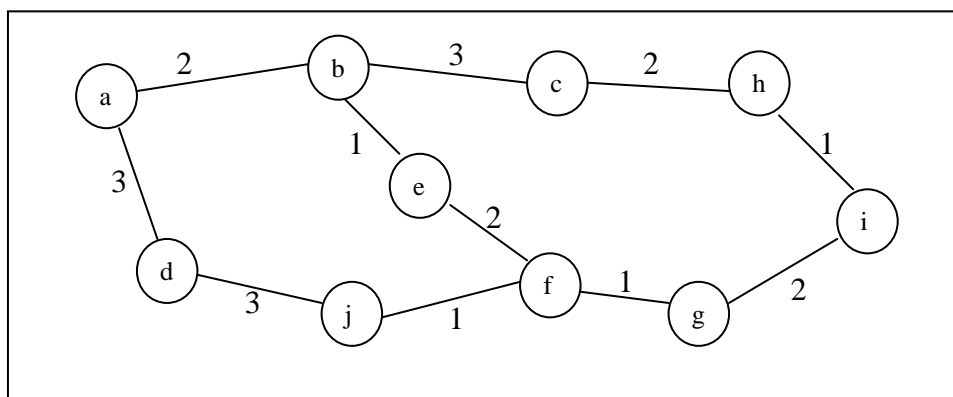
5. Beskriv kort hvilke feilsituasjoner vi kan komme opp i, og forklar hvordan vi håndterer dette, slik at vi får en feilfri overføring.
6. Du skal nå spesifisere en link protokoll basert på de betraktninger vi nettopp har gjort oss. Protokollen skal legges til rette for at flere applikasjonsprosesser kan benytte linken

samtidig. Hvilke felter trenger vi i pakkehodet og hvor store bør de være. Tegn og fortell og begrunn dine valg!

7. Dersom du finner ut at vi trenger en "timeout"-mekanisme, hva vil en fornuftig verdi for denne være og på hvilket tidspunkt startes "timer"-en?
8. Hvor stor er sannsynligheten for at en datapakke blir overført feilfritt mellom A og B, og hva er sannsynligheten for feilfri overføring av en kvitteringspakke?
9. Vurder om vi kan forbedre protokollen ved å innføre glidende vindu. Dersom du finner ut at det er en fordel å benytte glidende vindu, hvor stort bør da vinduet være og hvor mange bit vil du reservere for sekvensnummerering av pakkene?

Oppgave 2. Pakkesvitsjing

Figuren viser et enkelt pakke-svitsjet nett med 10 noder. Hver link er markert med en kostfaktor som er lik for datatransporten i begge retninger, for å gjøre figuren enklest mulig.



1. Vi tar for oss node a. Hva slags informasjon vil en linktilstandspakke som a sender inneholde? Hvem sender a slike pakker til?
2. Beregn minimum-kost treet for node a og illustrer dette i en figur.
3. Hvordan vil fremsendingstabellen for node a se ut?
4. Du vil muligens oppdage at flere noder kan nås på forskjellig måte. Kan vi utnytte dette på en eller annen måte?
5. Vi antar nå at nettet er stabilt med konstante kostfaktorer, og antar derfor at minimum kost basert på distansevektor metoden gir samme resultat som linktilstandsmetoden. Hvordan vil distansevektoren som a sender til b se ut? Hvem sender node a distansevektorer til?
6. Dersom du skulle velge rute-konsept for nettet i figuren over, ville du velge et distansevektor-opplegg eller et linktilstands-opplegg. Begrunn valget ditt.

Oppgave 3: TCP mekanismer

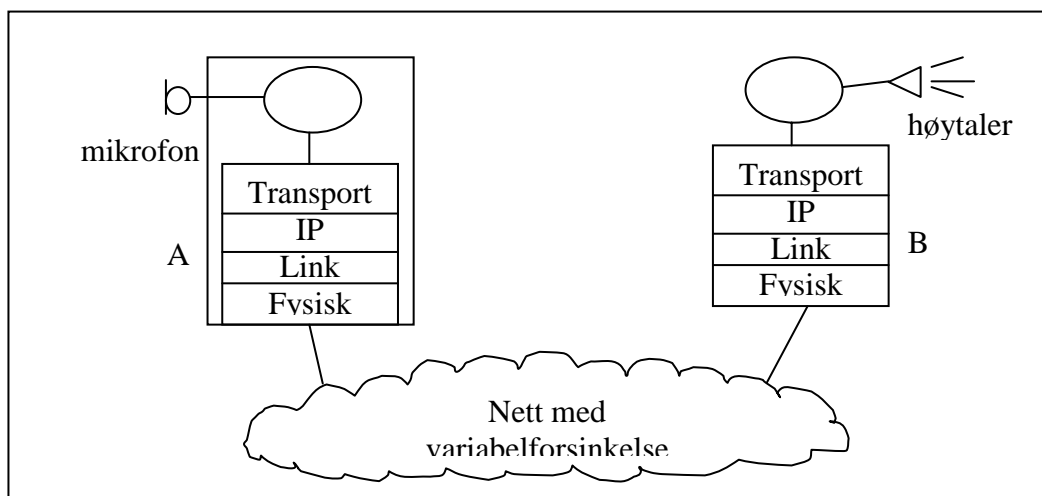
- a) Sendersiden på en TCP forbindelse mottar en data- eller kvitteringspakke med annonsert vindu (advertised window) med verdi lik null. Det vil si at sendersiden er blokkert. Hvilken mekanisme blir benyttet på sendersiden for å bringe den ut av den blokkerte tilstanden? Dersom denne mekanismen ikke ble benyttet, kunne vi da for eksempel utstyre mottakersiden med en ekstra "timer"? Hva skulle denne i så fall brukes til?
- b) Sekvensnummer-feltet i TCP hodet er 32 bit langt, hvilket er stort nok til å adressere over 4 milliarder bytes med data. Selv om så mange bytes aldri blir overført over en enkelt forbindelse, er det fortsatt mulig for sekvensnummeret å "gå rundt" fra $2^{32} - 1$ til 0. Forklar hvorfor.
- c) Du er hyret inn for å designe en pålitelig byte-strøm protokoll som benytter glidende vindu (på tilsvarende måte som TCP). Protokollen vil bli benyttet på et 100Mbit/s nett. RTT for

dette nettet er 100ms, og maksimal levetid for et segment er 60 sekunder. Hvor mange bit ville du benytte i ”annonsert vindu” og ”sekvensnummer” feltene i ditt protokoll-hode? Beskriv hvordan du kommer fram til disse tallene, og forklar hvilke verdier som kan være usikre.

- d) Anta at vertsmaskin A mottar to SYN pakker fra den samme porten på en fjern vertsmaskin B. Vi kan tenke oss to årsaker til dette: 1) det kan tenkes at den ene SYN pakken er en original SYN pakke mens den andre er en retransmisjon av originalen; 2) det kan og tenkes at B først sender en SYN pakke, deretter kræsjer, rebooter meget raskt og sender et helt nytt oppkoplingskall (SYN pakke). Beskriv forskjellen mellom disse to tilfellene slik som vertsmaskin A vil oppleve det. Gi en algoritmisk beskrivelse (pseudokode) av hva TCP-laget må gjøre når det mottar en SYN-pakke. Her skal du både ta med i betraktningen de to tilfellene ovenfor (duplikat/ny oppkopling), og den muligheten at ingen prosess lytter på destinasjons-porten.
- e) Hva er begrunnelsen for den eksponensielle økningen i ”timeout”-verdien som er foreslått av Karn og Patridge? Begrunn spesielt hvorfor en lineær (eller langsommere) økning vil kunne være uønsket.
- f) Vi forutsetter at TCP-mottakersiden sender en eksplisitt ACK for hvert mottatte segment som faller innenfor mottakerens vindu, enten denne er i riktig sekvens eller ikke. Dersom en ACK blir borte, oppfatter sendersiden dette som at nettet er i ferd med å gå, eller er gått, i metning. Hvordan utnyttes dette til å kontrollere sendersidens evne til sende data inn i nettet? Gi her en kort begrunnet forklaring på hvordan metningskontrollen fungerer.
- g) I oppgave f) er det bare endesystemene som er involvert. Vi ønsker også indikasjoner fra nettet om metningstilstanden der. Vi har to alternativer, nemlig RED og DEC.
- g.1) gi en kort forklaring av prinsippene bak disse to metodene
- g.2) diskuter fordeler og ulemper med de to alternativene. Hvilken av de to metodene kan benyttes uten å kreve endringer i TCP?

Oppgave 4. Spesifikasjon av protokoll for IP telefoni

Figuren illustrerer et nett med variabel forsinkelse. Vi betrakter to vertsmaskiner tilkoplett nettet og som kommuniserer med hverandre. Applikasjonen er telefoni – analog til digital omforming av tale i maskin A, og den motsatte prosessen i B. Den digitaliserte talen innpakkes først i en transportlags pakke og deretter i en IP pakke for transport over nettet frem til B. Vi ser bort fra Fysisk/Link-lagene i våre betraktninger.



Noen karakteristiske data for denne konfigurasjonen:

- IP leverer en "best-effort" tjeneste til transportlaget. For å gjøre ting enklere, antar vi at alle pakker som blir sent fra A kommer frem til B.
- Nettets forsinkelse varierer fra 0,1 til 0,15 sek. Vi ser bort fra andre mulige effekter som kan berøre forsinkelsen.
- Brutto datarate mellom de to transportlagene er 1 Mb/s.
- Applikasjonsprosessen i A leverer fra seg en blokk digital tale (1280 bit) hvert 0,02 Sek
- Applikasjonsprosessen i B krever å få en tilsvarende blokk digitale data for utspilling hvert 0.02 Sek.
- Vi antar et socket-liknende grensesnitt mellom transportlaget og applikasjonen, og for enkelthets skyld antar vi at det alt er etablert en assosiasjon mellom de to applikasjonsprosessene.

1. Fortell kort hvilke oppgaver transportlaget må kunne utføre med hensyn til overføring av den digitaliserte talen.
2. Vil den variable forsinkelsen i nettet ha betydning for disse oppgavene?
3. Spesifiser de viktigste feltene i transportprotokollhodet slik at transportentitetene skal kunne ivareta de oppgavene du har beskrevet i deloppgave 1 ? Begrunn dine valg av protokollfelder!
4. Spesifiser nå i pseudo-kode en prosedyre i transportlaget på sendersiden som formaterer transport segmentet (T-PDU). Kall prosedyren T-Send (). Den kalles fra applikasjons-prosessen. Når T-Send har formatert T-PDUen, kaller den en IP-Send () prosedyre i IP laget. Hvilke parametere trenger de to prosedyrene når prosedyrene kalles?
5. Videre skal du spesifisere to prosedyrer T-Ipop () og T-Apop () i transportlaget på mottakersiden. T-Ipop kalles fra IP laget og leverer en T-PDU til transportlaget, prosesserer denne og legger den til rette for applikasjonen. T-Apop kalles fra applikasjonsprosessen og henter inn en blokk med digitalisert tale for utspilling. Hvilke parametere trenger de to prosedyrene når de kalles?