**Eksamen TDAT2004 – Datakom 02.06.2017 SENSORVEILEDNING**

**Eksamen gjennomført digitalt på Inspera**

**1. Lagmodell**

1. Lag en liste over lagene i den 5-delte lagmodellen og plasser disse i rett rekkefølge. Forklar tjenester og protokoller i denne sammenhengen.

Lagene listes.

Tjenester: Tjenester tilbys fra laget under til laget over. Eks TCP tilbyr pålitelig overføring, IP tilbyr upålitelig overføring

Protokoller: Standarder for tolking og håndtering av datapakker som overføres mellom likestilte lag. Eks HTTP, IP

1. Hva er typiske egenskaper ved IP-adresser, portadresser og MAC-adresser?

IP: IPv4 og IPv6, Størrelse/antall bit, nett-id og host-id, organisering i adresseområder eks private, multicast. Eks på skrivemåter

Port: Størrelse/Antall bit, organisering i portområder (velkjente, system og dynamiske). Eks på skrivemåte port 80, reservert for HTTP

MAC: Størrelse/Antall bit, delt mellom fabrikantnummer og løpenummer, broadcastadresse. Eks på skrivemåte

**2. WEB**

1. HTTP/1.1 ble utformet i 1999 og brukes fortsatt. Forklar hvorfor protokollen er så tilpasningsdyktig.

Man kan føye til nye headerlinjer på formen «kolonseparerte navn-verdi par» for ny funksjonalitet. Klient og tjener velger å overse linjer man ikke «skjønner», derfor er systemet også bakoverkompatibelt.

1. Beskriv formål, virkemåte og eksempler på headerlinjer ved bruk av informasjonskapsler og vedvarende oppkopling.

Infokapsler: Skape tilstandsfull forbindelse (hukommelse). Det gir mulighet for å sette preferanser, bruke nettbutikk, være pålogget i sesjoner. Tjener lager cookie som sendes til klient: **Set cookie: innhold**. Ved klientens neste oppslag på denne URL sender klient med denne cookie som headerlinje. **Cookie: innhold**.

Vedvarende oppkopling: Kunne laste flere objekter over samme TCP-forbindelse. HTTP 1.0 koplet sesjonen ned etter hvert objekt. Vedvarende gir raskere respons. Klient sender forespørsel **Connection: keep alive**. Tjener kan godta med **Keep-alive: tid, antall objekter**

**3. DNS**

1. Beskriv hva som typisk skjer når man gjør et navneoppslag i DNS.

Pc program (resolver) sender DNS-oppslag til lokal navnetjener. Hvis info i cache, svar. Hvis ikke, lokal navnetjener «snur seg rundt» og spørre videre på vegne av klient. Spør først rottjener, som svarer med navnet til toppnivå navnetjener. Spør videre til toppnivåtjener, som igjen peker på hvem man kan spørre videre. Helt til man får svar fra autoritativ tjener eller om det er lagrest i cache under veis. Lokal navnetjener jobber rekursivt.

1. Hva kan ressursrecords av type MX og A ha til felles, og hvorfor er det hensiktsmessig?

Kan ha felles domenenavn for ulike tjenester. Eks både web og epostadresse for ntnu.no. Dette gir mulighet for forskjellig IP for ulike tjenester på samme domene, altså kjøre tjenere på ulike maskiner.

**4. TCP**

1. Beskriv kort hva som er hovedprinsippet i TCP for pålitelig overføring.

Hovedprinsippet er å sende kvitteringer. For å kunne gjøre det må man etablere en forbindelsesorientert og tilstandsfull sesjon. Under overføring av data holdes det nøyaktig oversikt over hva som er sendt og kvittert mottatt fra mottaker ved hjelp av flagg, sekvensnummer og kvitteringsnummer for hver byte. OBS: Glidende vindu, meningskontroll og flytkontroll er innretninger for ytelse, ikke pålitelighet.

1. Hva skjer ved oppkopling av en forbindelse?

Det gjennomføres et 3-way handshake:

Part A åpner og sender en TCP-pakke med flagget SYN satt. Det forteller motpart av det ønskes en oppkopling. I SYN-pakken følger startverdien av sekvensnummer-tlelleren (ISN, Initial Sequence number). Part A setter av plass i eget buffer for sending og mottak

Part B svarer med en pakke med både SYN-flagget, med eget ISN, og ACK-flagget satt. Det betyr at det initieres en forbindelse fra B 🡪 A, samt at den mottatte pakken kvitterer.

Part A kvitterer til slutt med en ACK-pakke på SYN-forespørsel fra B

Begge parter har nå satt av buffer for sending og mottak av segmenter, og har gitt verdier til sine sekvenstellere.

**5. TCP**

1. Beskriv bruk av tellere og flagg når det overføres to segmenter på 100 byte hver fra A til B

Prinsippet er at sekvensnummer viser posisjon til første byte i nyttelasten i senderetning, mens kvitteringsnummer viser neste forventede byte i mottakerretning.

Velger relativt startposisjon sekv.nr=1 (første segment med nyttelast). SYN-flagg=0 (kun satt ved oppsett forbindelse). Nyttelast kun fra A 🡪 B (sendes ikke inn returdata i eksempelet)

Kan også svare med akkumulert kvittering for to påfølgende pakker, men da er sekv.nr >1 for A 🡪 B

|  |  |
| --- | --- |
| A🡪 B (A til B) | A 🡨 B (B til A) |
| sekv.nr = 1  kvitteringsnr = 1  ACK-flagg = 1  Nyttelast 100 byte |  |
|  | Sekv.nr=1  kvitteringnr=101  ACK=1 |
| sekv.nr = 101  kvitteringsnr = 1  ACK-flagg = 1  Nyttelast 100 byte |  |
|  | Sekv.nr=1  kvitteringnr=201  ACK=1 |

1. Hva er hensikten med Glidende vindu? Beskriv virkemåte og hva som skjer dersom en pakke går tapt

Hensikten er å utnytte linjekapasiteten, kunne sende flere påfølgende segmenter uten å måtte vente på kvittering for hver pakke separat. (Kan bety en forskjell fra mindre enn 10% til mer enn 90% utnyttelse av linjekapasitet)

Sender har segmenter i et sendebuffer. Sender fortløpende til maks vindusstørrelse. Vindu «glir over» buffer etter hvert som segmenter blir kvittert ut, dvs sender stadig flere segmenter etter hvert som kvitteringer kommer.

Sendevindu reguleres av RTT (round trip time), metning og flytkontroll. (Krever ikke detaljer)

Ved uteblitte kvitteringer (tapte pakker eller forsinkelser) nedjusteres sendevindu til 1 hver gang, for så å øke eksponentielt for hver vellykkede overføring.

Ved pakketap må pakker sendes på nytt (retransmisjon). I hovedsak Go-Back-N, dvs sende alt på nytt fra og med tapte segment.

**6. Nettlaget**

1. Når bruker en PC sin nettmaske og hvordan skjer det?

En PC bruker nettmasken for å avgjøre om en pakke kan sendes direkte til mottaker på eget IP-nett eller om den må sendes til ruter for å nå et eksternt IP-nett

Bruker logisk AND mellom nettmaske for egen IP og mottakers IP for å finne nettadressene

1. Hva kjennetegner et IP-nett?

Alle noder på et IP-nett har samme nettadresse og nettmaske. Noder kan sende direkte til hverandre uten å gå via ruter. Nodene er på samme kringskastingsdomene. Nodene på IP-nettet har en felles ruter for kommunikasjon mot Internett. (Om IP-nettet bruker private IP-adresser mot en NAT-ruter endrer ikke på disse kjennetegnene)

**7. Nettlaget**

1. En PC er koplet til både kablet og trådløst Internett. Beskriv hvordan det avgjøres hvilken vei pakker skal sendes.

Når en PC er koplet mot to ulike IP-nett (kablet og trådløst) vil dette vises som to separate innslag i rutingstabell på PC. For å avgjøre hvilket nett man skal sende mot gjøres oppslag i rutingstabellen. Ruten med gunstigste metrikk (lavest kostand) velges. (Metrikk settes ved tilkopling til IP-nettet, ikke justert dynamisk etter trafikkforhold)

1. Hva kjennetegner autonome systemer (AS) på Internett? Hvordan er de koplet sammen?

Et autonomt system (AS) eies og driftes av en organisasjon (ISP), eks Uninett og Telenor. Alle rutere i AS bruker samme protokoll for dynamisk oppdatering av rutevalg. Hvert AS har et unikt AS-nummer.

Alle AS er koplet sammen i samtrafikkpunkter. Dette er svitsjer som overfører pakker fra et AS til et annet. Kalles Internet eXchange, og i Norge blir det NIX. Norge har flere slike (for å unngå single point of failure). Om et AS skulle miste forbindelsen til NIX kan brukere internt i AS fortsatt kommunisere med hvernadre, men ikke til resten av Internett.

(Ikke påkrevd i svaret: Et AS kan også ha direkte tilkoplet mindre AS under seg, stub-AS)

**8. Lenkelaget**

1. Hvordan fungerer bruk av preamble-signalet i forkant av rammer på lenkelaget?

Preamble er en sekvens av bit (101010101… og avsluttes med 11). brukes for å «vekke opp mottaker og synkronisere samplingsfrekvens slik at hvert bit den påfølgende rammen kan samples (avgjøres 0 eller 1) på optimalt tidspunkt.

1. Hvilke to prinsipper, avhengig av trafikk, kan trådløs overføring veksle mellom for overføring av datapakker?

Lav trafikk: Lyter, sende en pakke og vente på kvittering fra aksesspunkt

Stor trafikk: RTS/CTS. Node sender en kort request om tillatelse til å sende (RTS). Aksesspunkt sender CTS (Clear to send), som betyr at nå må alle andre noder være «stille» en viss periode. Unngår «skjult node» problem

Mekanismen kalles CSMA/CA (Collision Avoidance)

**9. Funksjoner**

1. Beskriv formål og virkemåte for Traceroute

Formål: kartlegge rute fra node A til B over Internett.

Virkemåte: bruker TTL-felt i IP-pakkeheader. Første pakke TTL=1, som dekrementeres (telles ned) i første ruter. TTL=0 har den effekten at ruter forkaster pakken og ICMP sender en feilmelding tilbake til avsender. Dermed fanger man opp ruters IP og første ruter er da «oppdaget». Går videre med trinnvis økning av TTL til målnoden svarer.

1. Beskriv formål og virkemåte for Address Resolution Protocol (ARP)

Formål: Finne MAC-adresse til en mottakers IP for å sende pakker på eget IP-nett.

Virkemåte: Avsendernode sender først MAC-broadcast og spør «Hvem har IP nn» og får svar. I svaret ligger nodens MAC-adresse. Koplingen MAC-IP adresser vedlikeholdes i ARP-tabell, og oppdateres jevnlig.

**10. Diverse**

1. Hvilken organisasjon står bak utviklingen av Internett-standardene (TCP/IP-familien), og hvordan skjer det?

Utvikles av IETF. Bruker RFC (request for Comment) som meode, dvs frivillige brukergrupper som diskuterer, tester og lager innstillinger.

1. Hva er hovedendringene i IPv6 sammenliknet med IPv4?

Først og fremst økt adresserom fra 32 til 128 bit. Må nevne strømlinjeforming av pakkeheader (fjerne fragmentering, kutte ut sjekksum, fast pakkehodestørrelse)