# Kontrol Spørgsmål

#### C1.Application layer

What is a socket? How can it be used?

Et applikations oprettet OS-kontrollerede interface (**en slags "dør"**), til hvilken applikationsprocessen kan både **sende og modtage beskeder** til / fra en anden applikationsproces. **Socket** er lokalt **identificeret** med et **portnummer**.

En socket er en slags **dør** imellem end **system programmet og transport lage**t. Før programmering er det vigtigt for udvikleren at bestemme hvilken port programmet skal bruge, **fx port 80 (standard for HTTP).**

#### C2. TCP/UDP

Isnt TCP always preferable to UDP since TCP provides a reliable data transfer

service and UDP does not?

**TCP (Transmission Control Protocol)** konstant **checker** for om alle **datapakker modtages** og derefter **justerer** **hastigheden** efter hvad der går godt, så kan man man sagtens komme ud for at køre med lavere hastighed end forventet og egentlig også end der er behov for, idet **routning** mellem **servere** på **internettet** konstant ændrer sig.

**Ved UDP** (User Datagram Protocol) er det **applikationerne**, der skal **checke** via den **checksum** protokollen leverer om **alle pakkerne er modtaget korrekt**.

UDP kan derfor være hurtigere end TCP til overførsel, og derfor kan det være godt at bruge i situationer hvor tab af enkelte pakker ikke betyder så meget (***f.eks. diverse streaming***)

#### C3. Routing

What is the diﬀerence between routing and forwarding?

**Forwarding:**

Forwarding af pakker, betyder at pakkerne bliver sendt i kopier til flere computere eller noder på netværket. Dette kan fx være broadcasting i form af internet radio eller internet broadcasting. Dette betyder også at det er modtageren (destinationen), som skal afslutte transmissionen medmindre at afsendings serveren går ned.

**Routing:**

Routing er mere målrettet. Ved routing bliver datagrammet på et tidspunkt afleveret fra source til destinationen som et resultat af gengivet forwarding. Her er det adresserne og netværkets opbygning der bestemmer hvor dataen skal sendes hen. Fx hvis man har en IP adresse der hedder 192.168.0.10, så ved man at computeren har host adressen 0.10 og at computeren befinder sig på netværket 192.168.0.0.

#### C4. DHCP

What is DHCP? How can it be used?

**Tildeler ipadresser, subnets og gateway adresser.**

**DHCP (Dynamic Host Configuration Protocol)**

En DHCP server står og lytter på et lokalt netværk på UDP port 67 og sørger for at klienter på netværket får tildelt en korrekt ip adresse og den rigtige information om netværkstruktur, altså subnet og gateways således at klienten kan kommunikere med andre computere på netværket. Forløbet er:

1. Klient starter op, altså maskinen tændes.
2. Netkortet undersøger om der er en DHCP server på netværket.
3. DHCP server svarer og sender sin IP adresse.
4. Netkortet spørger DHCP server: kan jeg få tildelt en IP adresse
5. DHCP server svarer og sender en IP adresse

#### C5. MAC layer

What are the pros and cons of parity and CRC checks?

**MAC addresser er FYSISKE addresser** på **hardware** enheder.

Ved anvendelse af **et 4 bytes CRC-felt** (Cyclic Redundancy Check) kan **man opdage eventuelle fejl** I en modtagen pakke. Feltet beregnes på basis af hele pakken, altså **inclusive** **adresse**- og **type**-felterne.

Hvis et **netkort** modtager en datapakke med **fejl** I, bliver pakken **kastet bort.** Det er så op til afsenderen evt. At blive træt af at vente på svar og sende pakken igen.

**Parity check** er en parietets bit, der altså kun kan være **0 eller 1**. Det gælder for en **hel transmission** og betyder at hvis **parietets checket fejler**, **skal hele transmissionen gentages** og ikke kun den enkelte pakke som det er tilfældet ved CRC.

#### C6. Wireless LANs

Why does 802.11 not check for collisions? Why does it handle retransmissions?

802.11 bruger teknikker til at slippe for kollision. Derudover er det også for dyrt at bygge hardware der kan opdage kollisioner.

Den bruger en ***link-layer aknowledgement/retransmission***. Når en bruger sender en frame, så er der en risiko for at dette frame ikke når til destinationen intakt.

For at løse dette fejl problem, bruger man et link-layer acknowledgement.

1. Når destinationen modtager en frame, venter den et lille stykke tid (kaldet en SIFS) og sender derefter ***en acknowledgement frame tilbage***.
2. Hvis den ikke er blevet modtaget i et bestemt tidsrum, ***så sender den en fejl og retransmitter framen*** ved brug af CSMA/CS protokollen.
3. Hvis det stadig ikke virker, så timer den ud.