

1)

Eksempler på slike forbindelser kan være SSH (Secure Shell) og filoverføringsprogrammet SCP. (Secure Copy). Hvis en av disse protokollene blir benyttet er forbindelsen sikker. Da kan ikke innholdet ses/ nås uten å ha en nøkkel.

For å kunne få tilgang til kryptert innhold må det benyttes nøkler. Det finnes to typer slik: public og private. **Public** nøkkel sendes til de som skal sende noe kryptert innhold tilbake til deg. For å dekode dette innholdet tilbake brukes en **private** nøkkel.

2)

Routerne prøver alltid å finne den beste veien gjennom et nettverk. Eller den med minst kostnad. Router algoritmene sin oppgave er derfor å finne gode veier (ruter) fra avsendere til mottakere gjennom et nettverk av routere. Det er en rekke faktorer som spiller inn når algoritmene skal finne den beste veien fra avsender til mottaker, men i praksis ser man også at andre ting spiller inn, slik som policy issues. Eksempelvis at et firma disponerer en router som ikke kan ta imot/ sende pakker fra et annet firma etc.

De to mest vanlige/kjente routingsalgoritmene er:

### **Link State (LS) Routing Algorithm:**

Denne algoritmen kartlegger hele nettet på forhånd. Altså alle noder (innenfor et AS) kjenner hele nettet. Dette er mulig ved at hver node sender Link-State pakker til alle andre noder. Disse pakkene inneholder identitet/kostnad hvor den respektive linken. Alle noder har dermed lik og full oversikt over nettet. (En annen måte å si det på: hver node har muligheten til å kjøre algoritmen for å regne ut den minst kostbare veien til hver av de andre nodene)

Denne algoritmen tar utgangspunkt i Dijkstra 's algoritme. Hvis man eksempelvis har seks noder i vil man ha fem mulige destinasjoner fra hver av de forskjellige nodene. Etter fem iterasjoner algoritmen vil da korteste vei til hver node være kjent.

### **Distance- Vector (DV) Routing Algorithm:**

Denne metoden fungerer på dette prinsippet: Hver av nodene mottar informasjon fra en eller flere direkte tilknyttede noder, gjør nødvendige utregninger (i henhold til mest effektive vei) og sender denne informasjonen tilbake til de respektive nodene. Algoritmen er iterativ og stopper når det ikke er mer informasjon å utveksle. En interessant del ved denne algoritmen er at det aldri går noe signal om at den skal stoppe, det skjer av seg selv!

Det finnes derfor noen forskjeller mellom disse to. Blant annet er DV-metoden lettere å implementere og håndtere. Det fordi LS- metoden er mer kompleks. (De baserer seg også selvsagt på forskjellige algoritmer/ prinsipper). En viktig forskjell er også at DV regner ut beste rute basert på avstand, mens LS på overall kostnad.

Fordi LS-metoden er mer kompleks, er det ofte slik at CPU og RAM på PC- i større grad blir utnyttet når denne kjører, i motsetning til DV.

3)

De tre mest vanlige rutingsprotokollene er:

-**IGP** (interior gateway protocols)

-**RIP** (Routing Information Protocol)

-**OSPF** (Open Shortest Path First)

Disse protokollene (som andre protokoller) spesifiserer hvordan rutere kommuniserer med hverandre. De distribuerer informasjon mellom rutere som gjør de i stand til å velge mellom to noder i et nettverk. (Det er algoritmene som tar den faktiske avgjørelsen)

Det er også verdt å nevne at routere som ligger i grensen mellom to AS bruker protokollen BGP (Border Gateway Protocol).

4)

Alle routere innenfor et AS må benytte seg av samme routingsalgoritme og routingsprotokoll. Algoritmen som brukes **innenfor** et bestemt AS kalles også Intra -AS routing algoritme/protokoll. Dette er altså algoritmen som tar seg av kommunikasjon mellom noder innenfor et bestemt AS. Denne protokollen ignorerer altså internett på utsiden av det spesifikke AS-et.

På den andre siden må også de forskjellige AS-ene kommunisere med hverandre. Protokollen som tar seg av dette kalles for Inter-AS routing algoritme/protokoll. Denne protokollen tar, i motsetning til overnevnte, utgangspunkt i at nettet kun består av de forskjellige AS-ene.

5)

Når en datapakke skal sendes fra en node A til en node B vil det alltid ta en tid før pakken kommer frem. Denne tiden avhenger hovedsakelig av to ting. Den fysiske avstanden mellom nodene, og hvor mange noder pakken må innom på veien til mottaker. Noen protokoller krever også et svar på om pakken kan sendes før den faktisk sendes, da vil det også gå med en del tid på dette.

Effektiviteten på en dataoverføring er ofte gitt ved  $U$ . Dette er forholdet mellom tiden det tar å sende en pakke, og tiden som går til neste pakke kan sendes. Tiden det tar for pakken å komme frem er en del av formelen, mens tiden det tar for signalet å komme tilbake fra mottaker (altså en rundtid) er den andre delen i formelen. Denne gjelder altså ved en stopp- og vent overføring. (Avsender må avvente svar fra mottaker før neste pakke kan sendes)

$$U = \frac{t_{trans}}{RTT + t_{trans}}$$
 her er altså  $t_{trans}$  = tiden det tar før pakken å komme frem, mens  $RTT$  er rundetiden.

6)

$$L = 2048 * 8$$

$$R = 50 * 10^6$$

$$\frac{L}{R} = \frac{2048 * 8}{50 * 10^6} = 3.28 * 10^{-4}$$

*RTT* leser jeg av oppgaven. (Velger average tid) som er 216ms =  $216 * 10^{-3} s$

Vi har også et sendevindu på 32 pakker, vi må da ta hensyn til dette i formelen.

$$U = 32 \left( \frac{3.28 * 10^{-4}}{216 * 10^{-3} * + 3.28 * 10^{-4}} \right) \approx 0.0485$$

$$U = \mathbf{0.0485}$$

7)

Datahastigheten finner vi ved å gange effektiviteten med den fysiske datahastigheten. (Hastigheten ut av PC-en)

$$Datahastighet = 0.0485 * 50 \approx 2.43$$

$$\mathbf{Datahastighet = 2.43 Mbps}$$