

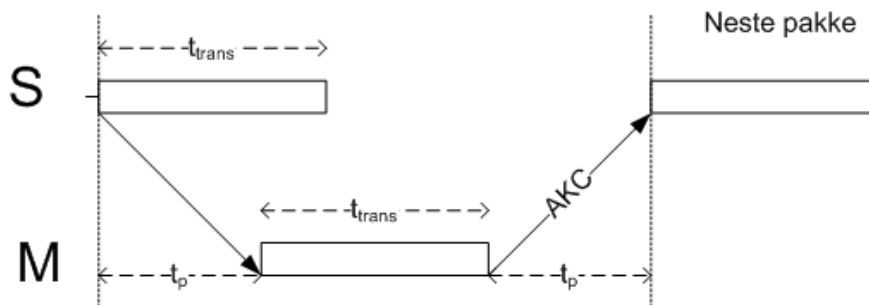
Effektivitet.

Når PC-A skal sende en datapakke til PC-B, vil det gå noe tid før pakken kommer fram. Tidsforsinkelsen mellom PC-A og PC-B er avhengig av den fysiske avstanden, og hvor mange noder (routere/switcher o.l.) pakken må gå igjennom.

Hvis protokollen krever et positivt svar (ACK) før den sender neste pakke (idle-RQ = stopp-og-vent), vil en stor del av tiden gå til å vente på svar. Det medfører at bare en liten del av tiden blir brukt til å faktisk sende data. Du vil oppleve det som at datahastigheten blir mindre, selv om den «fysiske» datahastigheten er stor.

Effektiviteten U er gitt av forholdet mellom tiden som brukes til å sende data, og tiden til neste pakke (eller serie med pakker) kan sendes. Vi setter tiden for å sende en pakke til t_{trans} og tiden signalet bruker for å komme fra sender (S) til mottager (M) og tilbake igjen; RTT (Round Trip Time). $RTT=2 \cdot t_p$
 L =pakkestørrelsen og R =bitshastigheten. Da blir U , ved stopp-og-vent overføring (idle RQ):

$$U = \frac{t_{trans}}{RTT + t_{trans}} = \frac{L/R}{RTT + L/R}$$



RTT mellom PC-A og PC-B kan man finne med kommandoen ping:

```
PING princeton.edu
```

```
Pinging princeton.edu [140.180.223.22] with 32 bytes of data:
```

```
Reply from 140.180.223.22: bytes=32 time=114ms TTL=243
```

```
Reply from 140.180.223.22: bytes=32 time=116ms TTL=243
```

```
Reply from 140.180.223.22: bytes=32 time=114ms TTL=243
```

```
Reply from 140.180.223.22: bytes=32 time=115ms TTL=243
```

Her ser du time, som ligger rundt 115 ms. Dette er RTT til www.princeton.edu.

Datahastigheten kan finnes ved å multiplisere verdien for U med den «fysiske» datahastigheten.

Hvis f.eks pakkestørrelsen er 8000 bit, den «fysiske» datahastigheten er 100 Mbit/s, og RTT er 115 ms, får vi:

$$L/R = 8000 / 100 \cdot 10^6 = 8 \cdot 10^{-5} \text{ [s]} = 80 \mu\text{s} \rightarrow U = 80 \mu\text{s} / 115 \text{ ms} = 7,0 \cdot 10^{-4}$$

Det gir datahastigheten = $7,0 \cdot 10^{-4} \cdot 100 \text{ Mbit/s} = 70 \text{ kbit/s}$

Hvis man kan sende flere datapakker, f.eks K pakker, før svar på den første må komme, vil effektiviteten øke.

Hvis f.eks $K=32$, blir $U = 32 \cdot 7,0 \cdot 10^{-4} = 22,4 \cdot 10^{-3}$, som gir en datahastighet på $22,4 \cdot 10^{-3} \cdot 100 \text{ Mbit/s} = 2,24 \text{ Mbit/s}$

Hvis det blir feil på en pakke, kan den sendes på nytt. Da er det to måter det kan gjøres på:

- 1) Selective retransmission: Kun den pakken som det var feil i sendes på nytt. Da vil forstyrre rekkefølgen av de pakkene som er mottatt. Da må mottageren få ordnet pakkene i riktig rekkefølge, før de sendes opp til laget over.
- 2) Go-Back-N: Pakkene sendes alltid i riktig rekkefølge. Hvis det er en feil i en pakke, må den pakken, og alle etterfølgende pakker sendes på nytt, selv om de etterfølgende pakkene har vært feilfri. På den måten vil mottageren alltid motta pakkene i riktig rekkefølge.

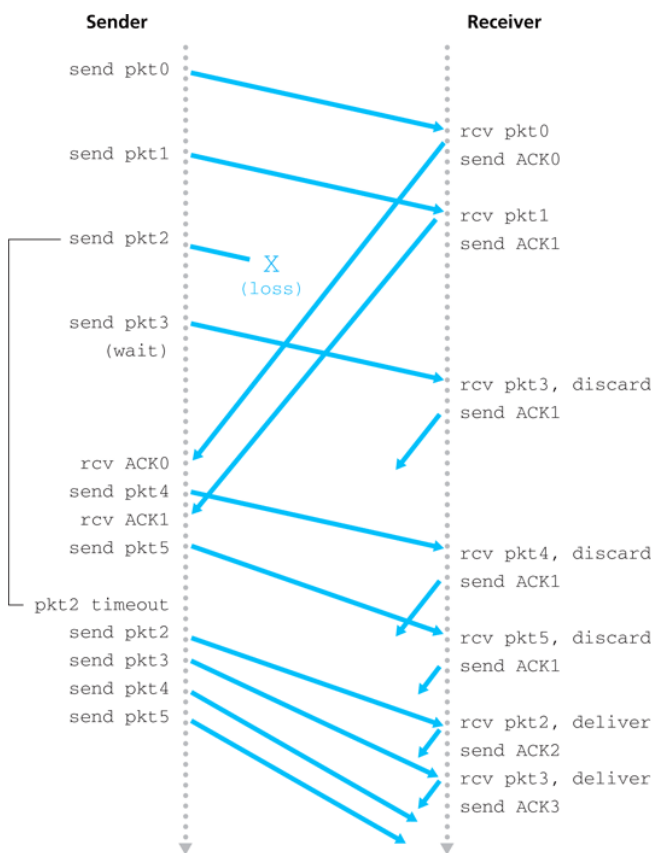


Figure 3.22 ♦ Go-Back-N in operation

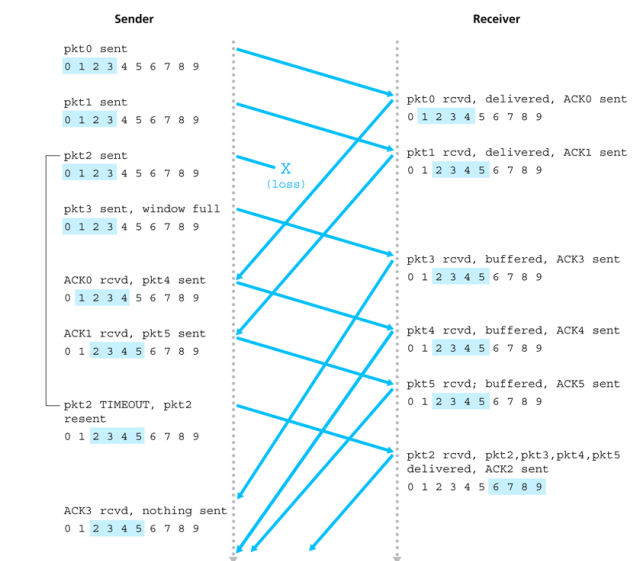


Figure 3.26 ♦ SR operation