

Tilakoneet selective repeat -algoritmissa

Tässä esityksessä olen pyrkinyt käyttämään Kurosen ja Rossin käyttämiä muuttujanimiä. Itse koodissahan on sitten toiset nimet. Lähettäjä (kuva 1) ylläpitää seuraavia muuttujia:

- `send_base` (koodissa `vanhin`): vanhimman lähetetyn mutta kuittaamattoman paketin sekvenssinumero, lähetysikkunan alaraja
- `nextseqnum` (koodissa `seur`): seuraavaksi lähetettävän paketin sekvenssinumero.

Vastaanottaja (kuva 2) puolestaan ylläpitää muuttujaa `rcv_base` (koodissa `vanhin`), joka on vastaanottoikkunan alaraja, siis pienin vastaanottoikkunaan kuuluva sekvenssinumero. Sekä lähetys- että vastaanottoikkunan koko on N (koodissa `ikkuna`).

Sekvenssinumeroita on käytännössä tietysti vain äärellinen määrä (koodissa tämä määrä on tallennettu attribuuttiin `max`), minkä vuoksi esim. muuttujan `nextseqnum` arvot jossain vaiheessa niin sanotusti pyörähtävät ympäri. Koneiden yhteydessä olevissa vertailuoperaatioissa (esim. `nextseqnum < send_base + N`) ei ole kuitenkaan – selvyiden vuoksi – otettu tätä huomioon.

Muuttujat `send_base`, `rcv_base` ja `nextseqnum` alustetaan nolllaksi.

Kuva 1: Lähettäjän tilakone.

Kutsuva ohjelma tarjoaa dataa lähetettäväksi.

```
-----  
if nextseqnum < send_base + N: # Jos lähetysikkunassa on tilaa:  
    # Paketoi data, sekvenssinumeroksi nextseqnum. Lähetä data.  
    # Käynnistä ajastin sekvenssinumerolla nextseqnum.  
    nextseqnum++
```

else:

Pyydä ohjelmaa yrittämään hetken päästä uudestaan.

```
# Saapuu kuittaus, jonka  
# sekvenssinumero on ackno.  
-----
```

notcorrupt = onko bittivirheetön?

if notcorrupt and ackno >= send_base and

ackno < send_base + N:

Pysäytä ajastin, joka vastaa sekvenssinumeroa

ackno. Merkitse vastaava paketti kuitatuksi.

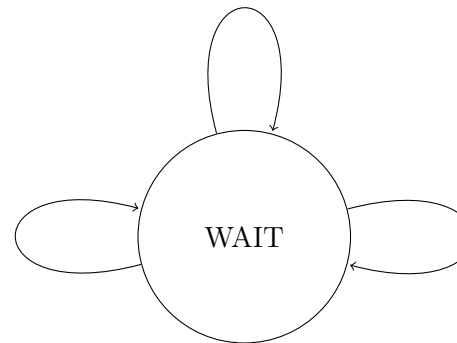
if ackno == send_base:

Kasvata lukua send_base niin paljon, että se

on pienin kuittaamattoman paketin sekvenssinumero.

else:

Älä tee mitään.



Ajastimen aikakatkaisu,
ajastimen numero seqno.

```
-----  
# Lähetä numeroa seqno  
# vastaava paketti uudelleen.  
# Käynnistä uudestaan ajastin,  
# jonka numero on seqno.
```

Kuva 2: Vastaanottajan tilakone.

```
# Saapuu bittivirheetön paketti,  
# jonka sekvenssinumerolle seqno pätee  
# seqno >= rcv_base ja  
# seqno < rcv_base + N.
```

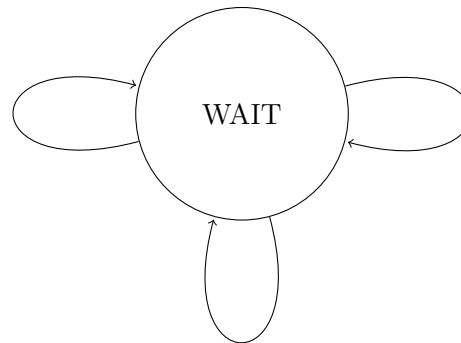
```
-----  
# Lähetä kuittaus.
```

```
if seqno == rcv_base:
```

```
    # Välitä eteenpäin paketti ja kaikki  
    # muut vastaanotetut ja puskuroidut  
    # paketit, joiden sekvenssinumerot  
    # muodostavat jonon  
    # rcv_base, rcv_base+1, ..., rcv_base+n,  
    # missä n >= 0 (ja siis mahdollisimman suuri).  
    rcv_base += n + 1
```

```
else:
```

```
    # Puskuroi paketti.
```



```
# Saapuu bittivirheetön paketti, jonka  
# sekvenssinumerolle seqno pätee  
# seqno >= rcv_base - N ja  
# seqno < rcv_base.
```

```
-----  
# Lähetä kuittaus.
```

```
# Saapuu mikä tahansa muu paketti.
```

```
-----  
# Älä tee mitään.
```