Appendix C:

Protokolbeskrivelse for ATU-232 tilslutning

Indhold:

Protokolbeskrivelse for ATU-232 tilslutning

Ved tilslutningsformerne ATU-232 og ATU-422 anvendes ALC (Asynchronous Line Control) protokollen, fuld duplex med flowkontrol, i det følgende omtalt som ALC/FD, når den anvendte kommunikationsform er 232IF/I eller 422IF/I.

Protokollen sætter et brugermodul i alarmudstyret i stand til at udveksle data med et brugermodul i ATU, som det er illustreret i fig. C.1 nedenfor.

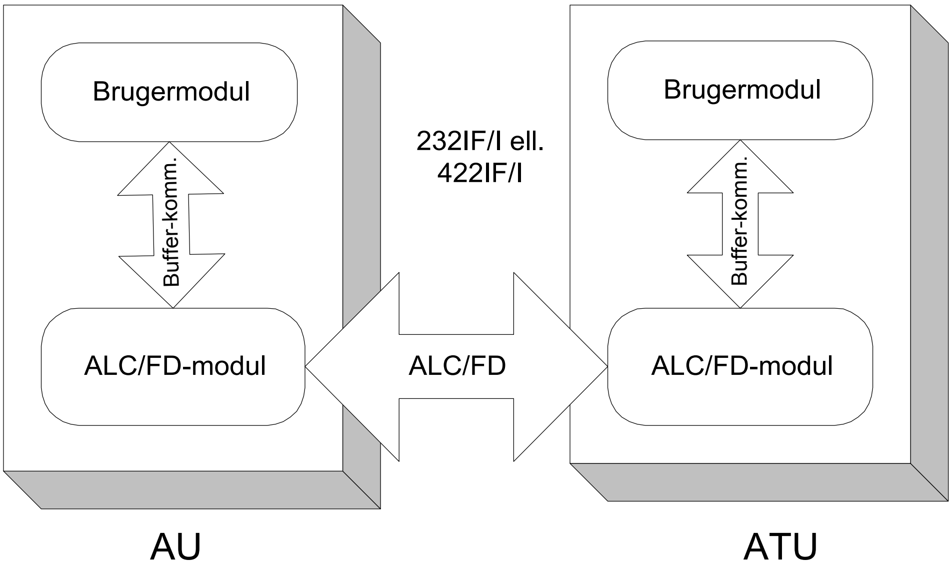

Fig. C.1: ALC/FD protokollen

ALC/FD protokollen er en byte-orienteret stop-og-vent protokol. Den består af to modsat rettede kanaler, som varetager overførslen af data i hver sin retning.

Hver kanal består af en primær station og en sekundær station. Den primære station sender kommandoer, det vil fortrinsvis sige data fra brugermodulet, til den sekundære station og den sekundære station kvitterer for disse til den primære station. Dette er skematisk vist på fig. C.2.

ALC/FD protokollen sikrer, at information kan føres gennem kommunikationsmediet

- uden at ændres
- uden at forsvinde
- uden at blive dubleret

Er det ikke muligt at overføre informationen indenfor et nøjere defineret tidsrum, opgives overførslen.

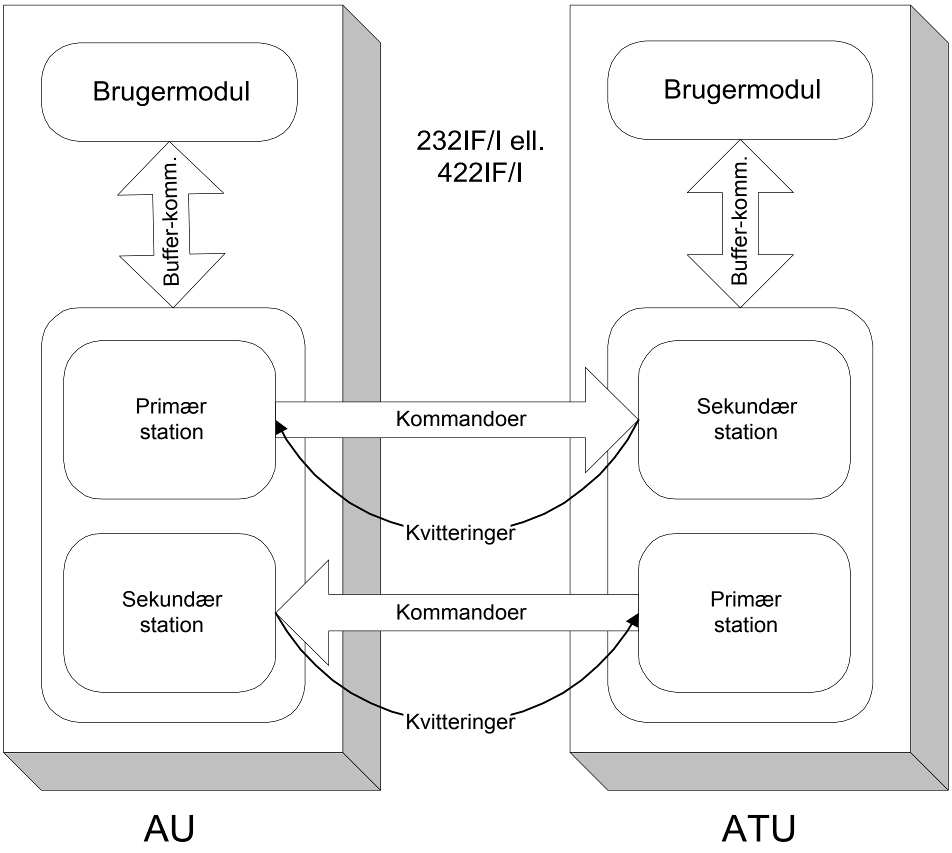

Fig. C.2: ALC/FD - primær og sekundær station

1. Formater

Den information, der overføres, er opdelt i pakker. Disse pakker kan have følgende udseende:

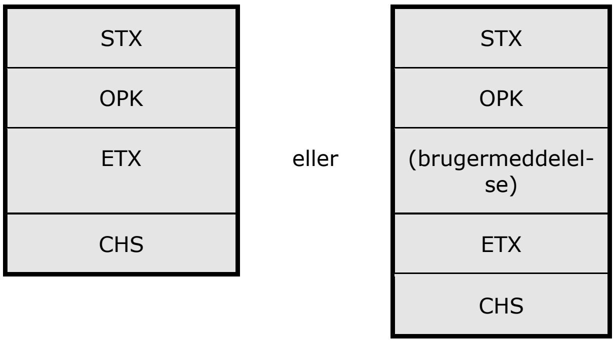

Fig. C.3: Pakkeformater

De enkelte felter i pakken har følgende betydning:

STX (Start of TeXt) markerer starten på en pakke.

OPK er en operationskode, der angiver om pakken skal opfattes som en kommando eller en kvittering.

ETX (End of TeXt) markerer afslutningen på pakken.

CHS checksum, der tjener til detektering af transmissionsfejl.

2. Operationskoder

Følgende operationskoder anvendes fra primær station:

- **ENQ**
- **DATA_0**
- **DATA_1**

DATA_0 og DATA_1 benyttes af den primære station til afsendelse af information. 0 og 1 kaldes sekvensnumre hvilket forklares senere. Kun når en af disse operationskoder anvendes, indeholder pakken en brugermeddelelse, som angivet i fig. C.3.

ENQ, enquiry, benyttes af den primære station til afhentning af sidst afgivne kvittering fra den sekundære station.

Følgende operationskoder anvendes fra den sekundære station:

- **ACK_0**
- **ACK_1**
- **RESET**
- **ACK_0_uden_kredit**
- **ACK_1_uden_kredit**
- **RESET_uden_kredit**

ACK_0 og ACK_1, acknowledge, benyttes som kvittering for modtagelse af henholdsvis DATA_0 og DATA_1 pakke, se forklaring senere vedrørende sekvensnummer. Med disse operationskoder overføres samtidig kredit, se nedenfor om flowkontrol.

RESET benyttes som svar på en ENQ pakke efter en restart af den sekundære station.

ACK_0_uden_kredit, ACK_1_uden_kredit og RESET_uden_kredit anvendes tilsvarende, men overfører ikke kredit til den primære station, se nedenfor om flowkontrol.

3. Hvis en kvittering udebliver...

Når den primære station har afsendt en datapakke, venter den et vist tidsrum på at modtage en modsvarende kvittering, hvilket vil sige en kvittering med samme sekvensnummer.

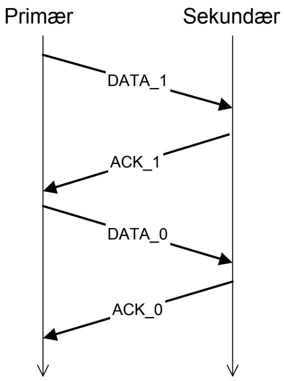

Fig. C.4: Normal dataoverførsel med brug af sekvensnumre

Udebliver kvitteringen, kan det have to årsager:

Den sidst afsendte pakke er ikke nået frem til den sekundære station

Den sidst afsendte pakke er nået frem, men kvitteringen fra den sekundære station er ikke nået tilbage til den primære station.

Den primære station sender nu en ENQ pakke, og modtager en kopi af den sidst afsendte kvittering fra den sekundære station.

Stemmer kvitteringens sekvensnummer med sekvensnummeret for den sidst afsendte pakke, er pakken nået frem, men den oprindelige kvittering gået tabt, dvs. situation 2 ovenfor. Den primære station kan herefter fortsætte med at sende den næste datapakke.

Har kvitteringen derimod ikke samme sekvensnummer som den sidst afsendte pakke, er det altså en kopi af kvitteringen for den forrige pakke, der er returneret. Det betyder, at den senest afsendte pakke ikke er modtaget af den sekundære station, dvs. situation 1 ovenfor. Den primære station må derfor gensende denne pakke.

Denne brug af enquiry er illustreret i fig. C.5:

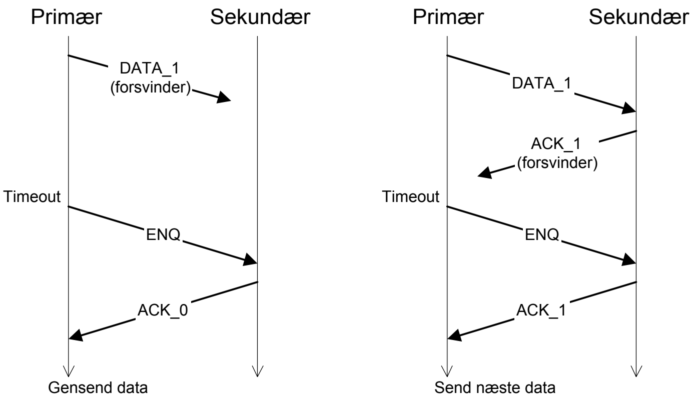

Fig. C.5: Brug af ENQ og sekvensnummer

Der opstår af og til transmissionsfejl. En sådan fejl indebærer, at en pakke forvanskes under transmissionen. Det vil sige, at mindst et af følgende er opfyldt:

- operationskoden er ukendt
- checksummen er ukorrekt
- pakkens længde er ikke korrekt

En forvansket pakke betragtes som ikke ankommet. Det vil sige at en station ikke må kvittere for den.

Der kan udtænkes scenarier, hvor forløbet bliver kompliceret af at f.eks. et enquiry og en kvittering krydser hinanden, men med en fornuftig timeout-værdi for den tid, den primære station venter på hhv. kvittering på data og svar på et enquiry, vil disse situationer normalt ikke kunne opstå. Beregningen af timeout-værdierne gennemgås senere i dette kapitel.

4. Flowkontrol

Som vist på fig. C.2, kommunikerer et brugermodul med både en primær station og en sekundær station. Kommunikationen foregår ved at brugermodulet afleverer fyldte sendebuffere til den primære station og stiller et antal tomme modtagebuffere til rådighed for den sekundære station.

Når den sekundære station modtager data, fyldes disse i en tom modtagebuffer. Hvis den sekundære station herefter ikke har flere ledige modtagebuffere, kvitteres der for modtagelsen af data med en af operationskoderne uden kredit. Er der derimod ledige modtagebuffere, anvendes den tilsvarende operationskode med kredit.

Hvis den primære station ikke har fået kredit fra den sekundære station, sendes et enquiry. Den primære station vil nu modtage en ACK-kvittering, og hvis den er med kredit, kan datapakken sendes, ellers sendes efter et tidsrum et nyt enquiry. Dette fortsætter indtil kredit er opnået.

Den sekundære station skal altså ikke blot besvare et enquiry ved at gensende den senest afsendte kvittering, men sørge for, at kreditinformationen er opdateret.

Dette udgør protokollens flowkontrol. Den primære station kun må sende data, hvis den har fået kredit fra den sekundære station - altså kun hvis den sekundære station er i stand til at behandle meddelelsen.

Flowkontrollen er ikke i sig selv et middel til en mere sikker kommunikation mellem primær og sekundær station. Var der ingen flowkontrol, kunne den sekundære station, når den ikke har en fri buffer, blot bortkaste de modtagne data og undlade at kvittere for disse. Den primære station vil så fortsætte med at gensende disse data, indtil den sekundære station atter kan tage fra.

Imidlertid skulle den sekundære station i disse situationer spilde værdifuld tid med at opsamle meddelelser, som alligevel ignoreres. Samtidig ville det uden flowkontrol ikke være muligt umiddelbart at detektere hvorvidt problemerne med at få overført data skyldes forbindelsen mellem primær og sekundær station eller kapacitetsproblemer i de bagved liggende systemer.

5. Timeout perioder

Timeout-perioderne konfigureres for den enkelte primærstation. Det betyder, at både på ATU- og alarmudstyrs-siden skal disse værdier fastlægges.

Umiddelbart skulle man tro, at mindste timeout-værdi skal svare til den tid, det tager at overføre en kommando med maksimal størrelse, behandle denne og overføre kvitteringen. Imidlertid er de fysiske linier delt af de to kanaler, således at den linie, som primærstationen i ATU overfører sine data på, også er den linie, sekundærstationen i ATU overfører kvitteringer. På tilsvarende vis deler primær- og sekundærstationerne i alarmudstyret en fysisk linie.

Ved beregningen af timeout-værdierne skal der derfor tages højde for, at der samtidig kan foregå trafik fra den anden kanal på samme fysiske linie.

Timeout-perioden efter afsendelse af en datapakke fra den primære station sættes derfor til minimum

- den mulige ventetid, mens den sekundære station på den anden kanal sender en kvittering
- plus den tid, det tager at overføre en datapakke med maksimal størrelse fra primær- til sekundærstationen
- plus den tid den sekundære station skal anvende på at behandle meddelelsen (dvs. kontrollere meddelelsens korrekthed og aflevere den til brugermodulet)
- plus den mulige ventetid, mens den primære station på den anden kanal sender en datapakke med maksimal størrelse
- plus den tid, det tager at overføre en kvittering fra sekundær- til primærstation.

Udtrykt som en formel vil det sige, at:

$$\mathbf{DATA\_TO} = \mathbf{2 * (PM + KL) / TH + PT}$$

hvor

DATA_TO = DATA TimeOut periode (i sekunder), dvs. det antal sekunder, den primære station venter efter afsendelsen af en datapakke, før et enquiry afsendes.

PM = Maksimal DATA-pakkelængde (i bit)

KL = Kvitteringslængde (i bit)

TH = Transmissionshastighed (i bit/s)

PT = Procestid i den sekundære station (i sekunder).

Eksempel: *Ved en maksimal pakkelængde på 86 tegn à 12 bit (databit, paritetsbit og stopbit), dvs. 1032 bit, en kvitteringslængde på 4 tegn à 12 bit dvs. 48 bit, en transmissionshastighed på 1200 bit/s og en procestid i den anden ende på f.eks. 1 sekund fås en timeout periode på 2.8 sekunder.*

Timeout-perioden efter afsendelse af ENQ-kommando kan på tilsvarende vis sættes til minimum den tid, den sekundære station skal anvende på at behandle meddelelsen, plus den tid det tager at transmittere én pakke med maksimal størrelse og en kvittering på den anden kanal, samt ét enquiry og en kvittering på egen kanal. Udtrykt som en formel vil det sige, at:

$$\textbf{ENQ-TO} = \mathbf{(PM + PE + 2 * KL) / TH + PT}$$

hvor

ENQ-TO = ENQ TimeOut periode (i sekunder), dvs. det antal sekunder, den primære station venter efter afsendelsen af et enquiry, før et nyt enquiry afsendes.

PM = Maksimal DATA-pakkelængde (i bit)

PE = Pakkelængde for ENQ (i bit)

KL = Kvitteringslængde (i bit)

TH = Transmissionshastighed (i bit/s)

PT = Procestid i den sekundære station (i sekunder).

***Eksempel**: Ved ENQ-pakkelængde på 4 tegn à 12 bit og samme størrelser som i eksemplet ovenfor fås en timeout periode på 1.98 sekunder.*

Der er som nævnt tale om mindsteværdier, altså kan større værdier anvendes. Specielt PT, procestiden, kan være vanskelig at estimere. Det anbefales, at følgende værdier anvendes:

	1200 bit/s	2400 bit/s	4800 bit/s	9600 bit/s
DATA timeout periode:	3 s	2 s	1,5 s	1,3 s
ENQ timeout periode:	2 s	1,5 s	1,3 s	1,2 s

Fig. C.6: Anbefalede timeout-værdier

6. Byte-timeout

Selvom der i pakkeformatet anvendes symbolerne STX og ETX, kan disse ikke entydigt bruges til at identificere start og slut på pakker, idet der kan komme data i brugermeddelelsen med disse værdier.

Brugermeddelelsen indeholder altid et længdefelt, men i fejlsituationer er det væsentligt, at det så hurtigt som muligt kan lykkes for den modtagende station at genoprette en korrekt fortolkning af de modtagne data.

Det anbefales derfor, at der i såvel primær- som sekundærstationen anvendes en timeout-værdi for modtagelsen af næste byte i en meddelelse. Overskrides denne timeout-værdi, bør stationen bedømme det hidtil modtagne som en forvansket meddelelse, og det forventes at næste byte vil være starten på en ny meddelelse.

Som byte-timeout anbefales en værdi på maksimalt det halve af ENQ timeout perioden.

7. Andre konfigurerbare parametre

Fastlæggelsen af nedenstående parametre vil være afhængigt af anvendelsen, dvs. alarmudstyret og brugermodulets mulige alternative handlinger.

Parameter:	Anbefalet værdi:
Max antal transmissionsforsøg	4
Max antal kredit ENQ	5

Fig. C.7: Anbefalede værdier for konfigurerbare parametre

8. Primær station

I det følgende beskrives, hvordan den primære station fungerer under opstart, normal dataoverførsel og i fejlsituationer.

Under opstart sender den primære station med jævne mellemrum en ENQ-kommando til den sekundære station. Når den primære station modtager enten en RESET-kvittering eller en ACK-kvittering, er forbindelsen etableret.

Som tidligere beskrevet er det sådan i ALC/FD protokollen, at et skift i sekvensnummeret (DATA_0 til DATA_1 eller omvendt, ACK_0 til ACK_1 eller omvendt) indikerer korrekt tilstand, mens fejltilstande resulterer i uændret sekvensnummer.

Under normal dataoverførsel sender den primære station en DATA-pakke indeholdende en brugermeddelelse og et sekvensnummer. Derefter modtager den en kvittering, der indikerer, at pakken er blevet korrekt modtaget hos den sekundære station, hvorefter den kan sende en DATA-pakke med en ny brugermeddelelse og et nyt sekvensnummer.

Primærstationens funktion kan beskrives som en række tilstande, primærstationen kan befinde sig i. I disse tilstande kan der indtræffe forskellige begivenheder, som resulterer i en handling, udført af primærstationen. Rent grafisk kan det beskrives med nedenstående tegning (fig. C.8). Cirklerne symboliserer tilstande, pilene symboliserer begivenheder, og rektanglerne symboliserer handlinger.

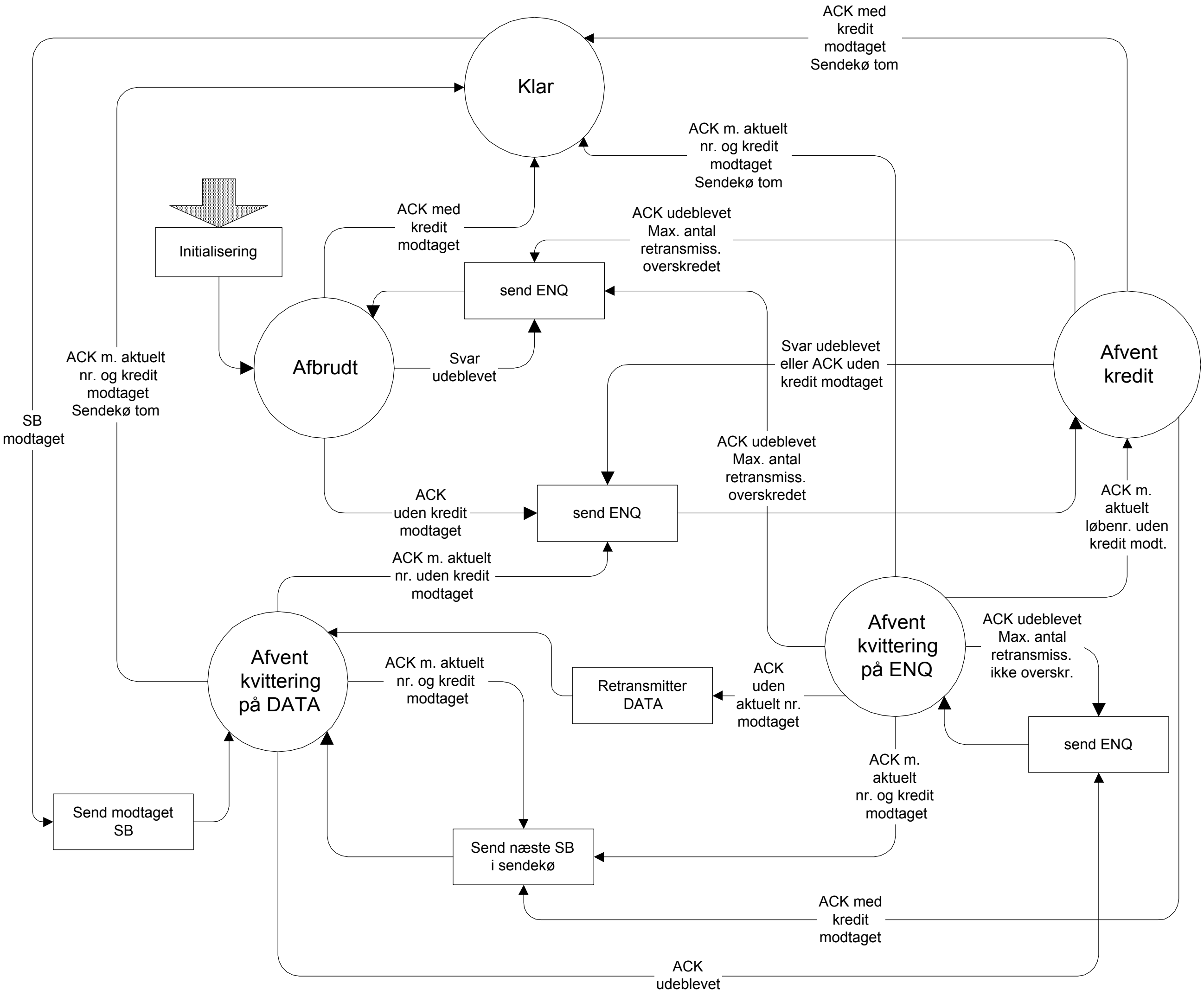

Fig. C.8: Tilstands-handlingsbeskrivelse for primær station

I det følgende beskrives først initialiseringen og dernæst de enkelte tilstande i detaljer: hvilke begivenheder, der kan indtræffe, og for hver af disse begivenheder beskrives de handlinger, begivenheden medfører, og hvilket tilstandsskift, der indtræder.

Beskrivelsen er skrevet i et pseudokode-sprog, baseret på en dansk udgave af kontrolstrukturerne fra programmeringssproget C. Disse dele er skrevet med **almindelig courier-skrift**, mens pseudokodedelene er skrevet med ***<kursiveret courier i kantede parenteser>.*** Kommentarer er - i overensstemmelse med syntaksen for C - omgivet af symbolerne “`/*`” og “`*/`”.

En række handlinger gentages i forskellige tilstande. Der refereres til en gentagelse af disse med pseudokode-sætningen “***udfør HANDLING x***”, hvor x er det nummer, der er anført ud for beskrivelsen af handlingen.

“***Start xxx timeout periode***” betyder i det følgende, at der bestilles et timeout den angivne periode senere. Hidtil bestilte - endnu ikke indtrådte timeouts - annulleres implicit. Den eneste tilstand, hvor der ikke er bestilt timeout, er tilstanden klar. Tidligere bestilte timeouts ignoreres blot i denne tilstand.

På oversigtsform kan de enkelte tilstande, begivenheder og handlinger beskrives med nedenstående skema, idet tallene refererer til handlingernes numre i de efterfølgende beskrivelser, og bogstaverne til de mulige tilstande efter handlingen:

Tilstand	Begivenhed			
	SB	RESET	ACK	Timeout
A: Afbrudt	1 A	2 B, E	3 B, E	4 A
B: Klar	5 C	6 B	7 B	8 B
C: Afvent kvittering på DATA	9 C	10 C	11 B, C, E	12 A, D
D: Afvent kvittering på ENQ	13 D	14 C	15 B, C, E	16 A, D
E: Afvent kredit	17 E	18 B, C, E	19 B, C, E	20 A, E

Fig. C.9: Tilstands-handlingstabel for primær station

	Primær station Initialisering
Handling	

```
{
  sekvensnr = 1;
  tilstand = afbrudt;

  <send ENQ>
  <start ENQ timeout periode>

  antal_transmissionsforsøg = 1;
  antal_kredit_ENQ = 0;
}
```

Tilstand:	afbrudt
Begivenhed:	SB (der er modtaget en sendebuffer fra brugermodul)
Handling 1:	

```
{
  <returner buffer med resultat: ingen forbindelse>
}
```

Tilstand:	afbrudt
Begivenhed:	RESET (der er modtaget RESET fra sekundærstationen)
Handling 2:	

```
{
 hvis (<operationskoden er uden kredit>)
 {
  kredittilstand = ingen_kredit;
 }
 ellers
 {
  kredittilstand   = kredit
 }
 antal_transmissionsforsøg = 0  /* idet der kom
                                  et svar      */

 hvis (kredittilstand == ingen_kredit)
 {
  tilstand = afvent_kredit;
  antal_kredit_ENQ   = 1;

  <send ENQ>
  <start ENQ timeout periode>
 }
 ellers /* kredit */
 {
  antal_kredit_ENQ = 0;
  tilstand = klar;
 }
}
```

Tilstand:	afbrudt
Begivenhed:	ACK (der er modtaget en ACK fra sekundærstationen)
Handling 3:	

```
{
 sekvensnummer = <samme værdi som angivet
                i ACK-kvitteringen>;

 <udfør HANDLING 2>
}
```

Tilstand:	afbrudt
Begivenhed:	Timeout (timeoutperiode er udløbet)
Handling 4:	**{** ***<send ENQ>*** ***<start ENQ timeout periode>*** **antal_transmissionsforsøg += 1;** **}**

Tilstand:	klar
Begivenhed:	SB (der er modtaget en sendebuffer fra brugermodul)
Handling 5:	**{** **sekvensnummer = *<næste sekvensnummer>*;** ***<send DATA-kommando med sekvensnummer>*** **aktuel_sendebuffer = *<afsendt sendebuffer>*;** ***<Start DATA timeout periode>*** **antal_transmissionsforsøg = 0;** **tilstand = afvent_kvittering_på_DATA;** **}**

Tilstand:	klar
Begivenhed:	RESET (der er modtaget RESET fra sekundærstationen)
Handling 6:	**{** ***<gør intet>*** **}**

Tilstand:	klar
Begivenhed:	ACK (der er modtaget ACK fra sekundærstationen)
Handling 7:	**{** ***<gør intet>*** **}**

Tilstand:	klar
Begivenhed:	Timeout (timeoutperiode er udløbet)
Handling 8:	**{** ***<gør intet>*** **}**

Tilstand:	afvent_kvittering_på_DATA
Begivenhed:	SB (der er modtaget en sendebuffer fra brugermodul)
Handling 9:	**{** ***<sæt sendebuffer i kø til overførsel>*** **}**

Tilstand:	afvent_kvittering_på_DATA
Begivenhed:	RESET (der er modtaget RESET fra sekundærstationen)
Handling 10:	**{** ***<gør intet>*** **}**

Tilstand:	afvent_kvittering_på_DATA
Begivenhed:	ACK (der er modtaget ACK fra sekundærstationen)

Handling 11:

```
{
 hvis (<sekvensnummer i ACK-kvitteringen> ==
      <sekvensnummer i den afsendte datakommando>)
 {
  <send aktuel sendebuffer tilbage
   til bruger med resultat ok>

  hvis (<ACK-operationskoden> == ACK_uden_kredit)
  {
   kredittilstand = ingen_kredit;
  }
  ellers
  {
   kredittilstand = kredit;
  }

  antal_transmissionsforsøg = 0 /* idet der kom svar */

  hvis (kredittilstand == ingen_kredit)
  {
   tilstand = afvent_kredit;
   antal_kredit_ENQ = 1;

   <send ENQ>
   <start ENQ timeout periode>
  }
  ellers /* kredit */
  {
   hvis (antal_sendebuffere == 0)
   {
    tilstand = klar;
   }
   ellers
   {
    <Næste sendebuffer tages ud af køen>
    <udfør HANDLING 5>
   }
  }
 }
 ellers /* ikke aktuel kvittering:
           synkroniseringsfejl */
 {
  <gør intet>
 }
}
```

Tilstand:	afvent_kvittering_på_DATA
Begivenhed:	Timeout (timeoutperiode er udløbet)
Handling 12:	

```
{
 hvis (antal_transmissionsforsøg >=
       max_antal_transmissionsforsøg)
 {
  <returner aktuel sendebuffer
   til bruger med resultat opgivet>
  <send alle sendebuffere i køen tilbage
   til bruger med resultat ingen forbindelse>

  antal_transmissionsforsøg = 1;

  <send ENQ>
  <start ENQ timeout periode>

  tilstand = afbrudt;
 }
 ellers
 {
  <send ENQ>
  <start ENQ timeout periode>

  antal_transmissionsforsøg += 1;
  tilstand = afvent_kvittering_på_ENQ;
 }
}
```

Tilstand:	afvent_kvittering_på_ENQ
Begivenhed:	SB (der er modtaget en sendebuffer fra brugermodul)
Handling 13:	

```
{
 <sæt sendebuffer i kø til overførsel>
}
```

Tilstand:	afvent_kvittering_på_ENQ
Begivenhed:	RESET (der er modtaget RESET fra sekundærstationen)
Handling 14:	

```
{
  <retransmitter sidste DATA-kommando>

  antal_transmissionsforsøg += 1;

  <start DATA timeout periode>

  tilstand = afvent_kvittering_på_DATA;
}
```

Tilstand:	afvent_kvittering_på_ENQ
Begivenhed:	ACK (der er modtaget ACK fra sekundærstationen)
Handling 15:	

Handling 15:

```
{
 hvis (sekvensnummer i ACK-kvitteringen ==
      sekvensnummer i den afsendte datakommando)
 {
  <send aktuel SB tilbage til bruger med resultat ok>

  antal_transmissionsforsøg = 0; /* idet der kom svar */

  hvis (ACK_operationskode == ACK_uden_kredit)
  {
   kredittilstand = ingen_kredit;
   tilstand = afvent kredit;
   antal_kredit_ENQ = 1;

   <send ENQ>
   <start kredit ENQ timeout periode>
  }
  ellers
  {
   kredittilstand = kredit;

   hvis (antal_sendebuffere == 0)
   {
    tilstand = klar;
   }
   ellers
   {
    <næste sendebuffer tages ud af køen>
    <udfør HANDLING 5> /* tilstand er nu
                       afvent_kvittering
                       på_DATA          */
   }
  }
 }
 ellers /* sekundær station modtog ikke sidste pakke */
 {
  <retransmitter sidste DATA-kommando>

  antal_transmissionsforsøg += 1;

  <start DATA timeout periode>

  tilstand = afvent_kvittering_på_DATA;
 }
}
```

Tilstand:	afvent_kvittering_på_ENQ
Begivenhed:	Timeout (timeoutperiode er udløbet)
Handling 16:	**{** ***<udfør handling 12>*** ***}***

Tilstand:	afvent_kredit
Begivenhed:	SB (der er modtaget en sendebuffer fra brugermodul)
Handling 17:	**{** ***<sæt sendebuffer i kø>*** ***}***

Tilstand:	afvent_kredit
Begivenhed:	RESET (der er modtaget RESET fra sekundærstationen)
Handling 18:	

```
{
  kreditttilstand = <kredittilstand i modtaget buffer>;
  antal_transmissionsforsøg      = 0  /* idet der kom svar */

  hvis (kredittilstand == ingen_kredit)
  {
    hvis (antal_kredit_ENQ >= max_antal_kredit_enq)
    {
      <returner alle SB med resultat busy>

      antal_kredit_ENQ = 1;
    }
    ellers
    {
      antal_kredit_ENQ += 1;
    }
    <afvent at ENQ timeout periode er slut,
     før næste ENQ sendes>
  }
  ellers   /* kredit */
  {
    antal_kredit_ENQ = 0;

    hvis (antal_sendebuffere == 0)
    {
      tilstand = klar;
    }
    ellers
    {
      <næste sendebuffer tages ud af køen>
      <udfør HANDLING 5> /* tilstand er nu
                     afvent_kvittering_på_DATA */
    }
  }
}
```

Tilstand:	afvent_kredit
Begivenhed:	ACK (der er modtaget ACK fra sekundærstationen)
Handling 19:	

```
{
  <udfør HANDLING 18>
  /* NB: modtaget kvitteringsnummer aflæses ikke */
}
```

Tilstand:	afvent_kredit
Begivenhed:	Timeout (timeoutperiode er udløbet)
Handling 20:	

```
{
 hvis (antal_transmissionsforsøg >=
      max_antal_transmissionsforsøg)
 {
  <returner alle sendebuffere med
   resultatet no connection>

  antal_transmissionsforsøg   = 1;
  tilstand = afbrudt;
 }
 ellers
 {
  antal_transmissionsforsøg += 1;
 }

 <send ENQ>
 <start ENQ timeout periode>
}
```

9. Sekundær station

Under normal dataoverførsel kvitterer den sekundære station for en DATA-kommando med en ACK-pakke med samme sekvensnummer. Ved modtagelse af en ENQ-kommando svarer den sekundære station med sidste kvittering, d.v.s. RESET, hvis stationen netop er startet op, ACK_0, hvis sidst modtagne DATA-kommando var DATA_0, og ACK_1, hvis sidste modtagne DATA-kommando var DATA_1.

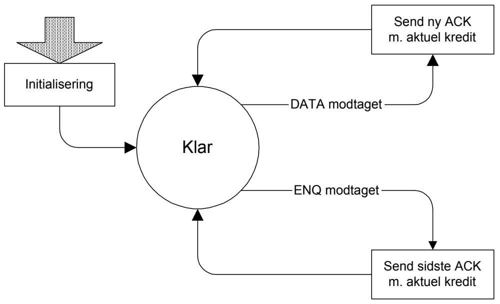

Fig. C.10: Tilstands-handlingsbeskrivelse for sekundær station

Kredittilstand indikeres i alle kvitteringer. Det vil sige, at hvis den seneste kvittering var ACK_0_uden_kredit, og der siden da er frigjort en modtagebuffer, svares nu med ACK_0, dvs. med kredit. Der er således to kredittilstande:

- Ingen kredit: antal modtagebuffere = 0
- Kredit: antal modtagebuffere > 0

Ved modtagelse af en forvansket kommando kvitterer den sekundære station ikke. Hvis der ikke findes en modtagebuffer til de læste data, kvitterer den sekundære station ikke. På grund af flowkontrollen med angivelse af kredittilstand i alle kvitteringer bør dette ikke forekomme.

Beskrevet på samme form som den primære station, er sammenhængen mellem tilstand, begivenheder og handlinger følgende:

Tilstand	Begivenhed	
	ENQ	DATA
A: Klar	1 A	2 A

Fig. C.11: Tilstands-handlingstabel for sekundær station

	Sekundær station initialisering
Handling:	

```
{
 tilstand = klar;
 sidste_kvittering = RESET;
}
```

Tilstand:	klar
Begivenhed:	ENQ (der er modtaget ENQ fra primærstationen)

Handling 1:

```
{
 ACK_sekvensnummer = seneste_kvittering;

 hvis (antal_modtagebuffere > 0)
 {
  <ACK sendes til primærstation> /* med kredit */
 }
 ellers
 {
  <ACK_uden_kredit sendes til primærstation>
 }
}
```

Tilstand:	klar
Begivenhed:	DATA (der er modtaget DATA fra primærstationen)

Handling 2:

```
{
 hvis (antal_modtagebuffere > 0)
 {
  <kopier input til modtagebuffer og
   returner den til bruger>

  ACK_sekvensnummer = DATA_sekvensnummer;

  hvis (antal_modtagebuffere > 0)
  {
   <ACK sendes til primærstation> /* med kredit */
  }
  ellers
  {
   <ACK_uden_kredit sendes til primærstation>
  }
  seneste_kvittering = <operationskode for
                      den afsendte ACK>
                     /* der gemmes ingen oplysning om
                        eventuelt manglende kredit */
 }
 ellers
 {
  <gør intet> /* input ignoreres. Denne situation bør
               ikke indtræffe, idet primærstationen
               løbende er informeret om kredit-
               tilstand hos sekundær station */
 }
}
```