

Appendix G:

Protokolbeskrivelse for vagtcentraltilslutning

Indhold:

1. Formater	3
2. Operationskoder	4
3. Hvis en kvittering udebliver.....	5
4. Primær station	7
5. Sekundær station	17

Protokolbeskrivelse for vagtcentral-tilslutning

Ved vagtcentral-tilslutningsformen ALC anvendes ALC (Asynchronous Line Control) protokollen, fuld duplex, i det følgende omtalt som ALC/FD.

Protokollen sætter et brugermodul i vagtcentralen i stand til at udveksle data med et brugermodul i AlarmNettet (nærmere bestemt en VC-connector i terminalstationen (TS)), som det er illustreret i fig. G.1 nedenfor.

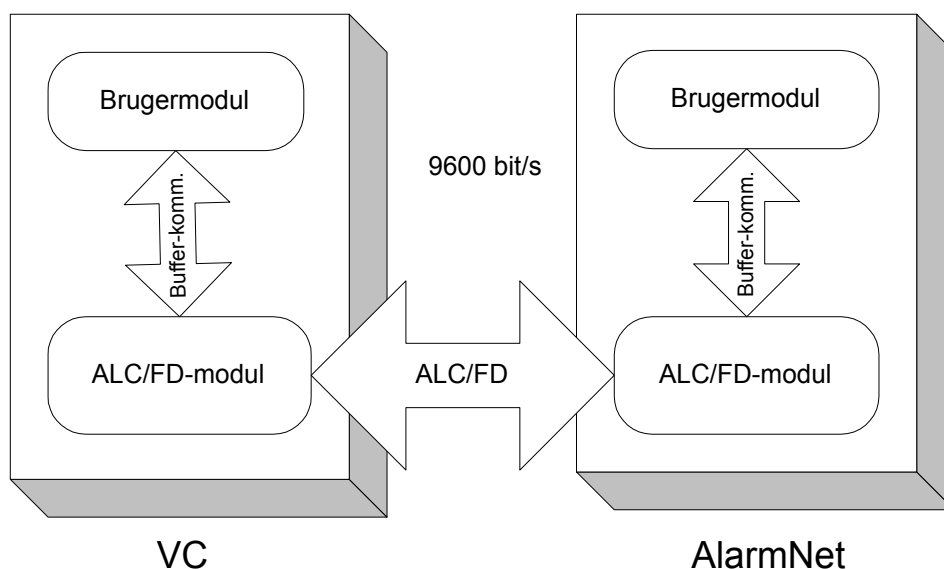


Fig. G.1: ALC/FD protokollen

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. G.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.

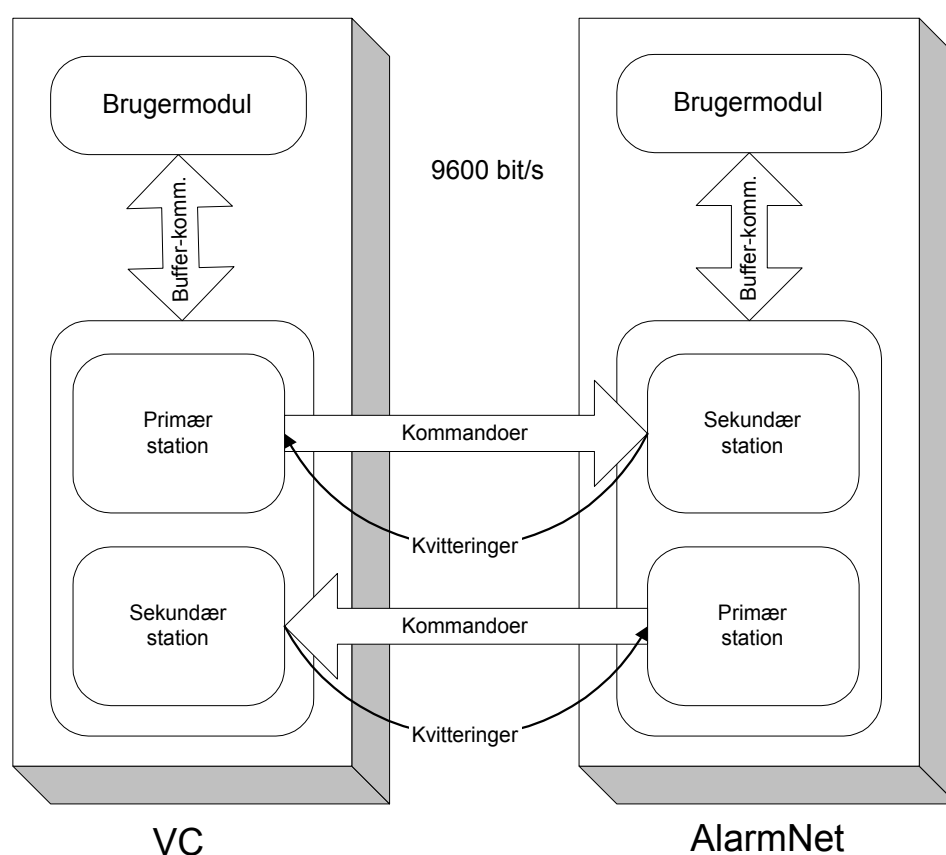


Fig. G.2. ALC/FD - primær og sekundær station

1. Formater

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

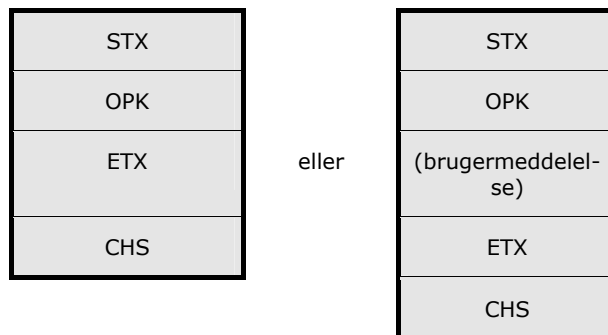


Fig. G.3: Pakkeformater

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. G.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 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.

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.

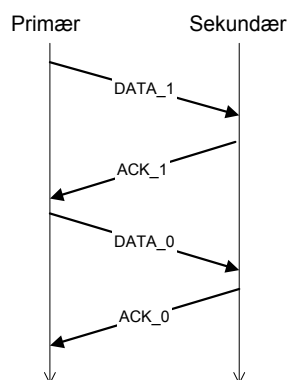


Fig. G.4: Normal dataoverførsel med brug af sekvensnumre

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. G.5:

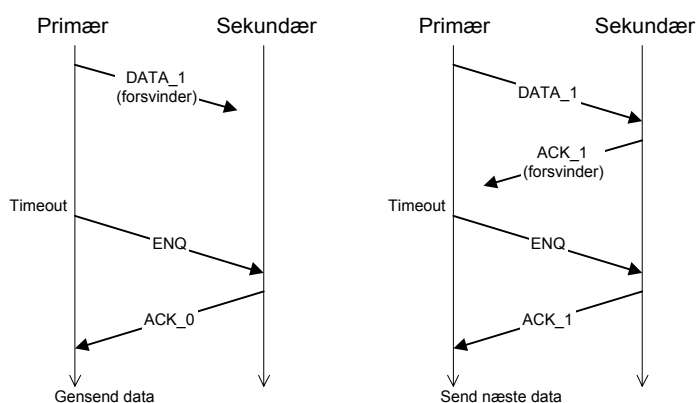


Fig. G.5: Brug af ENQ og sekvensnummer

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.

4. 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. G.6). Cirklerne symboliserer tilstande, pilene symboliserer begivenheder, og rektanglerne symboliserer handlinger.

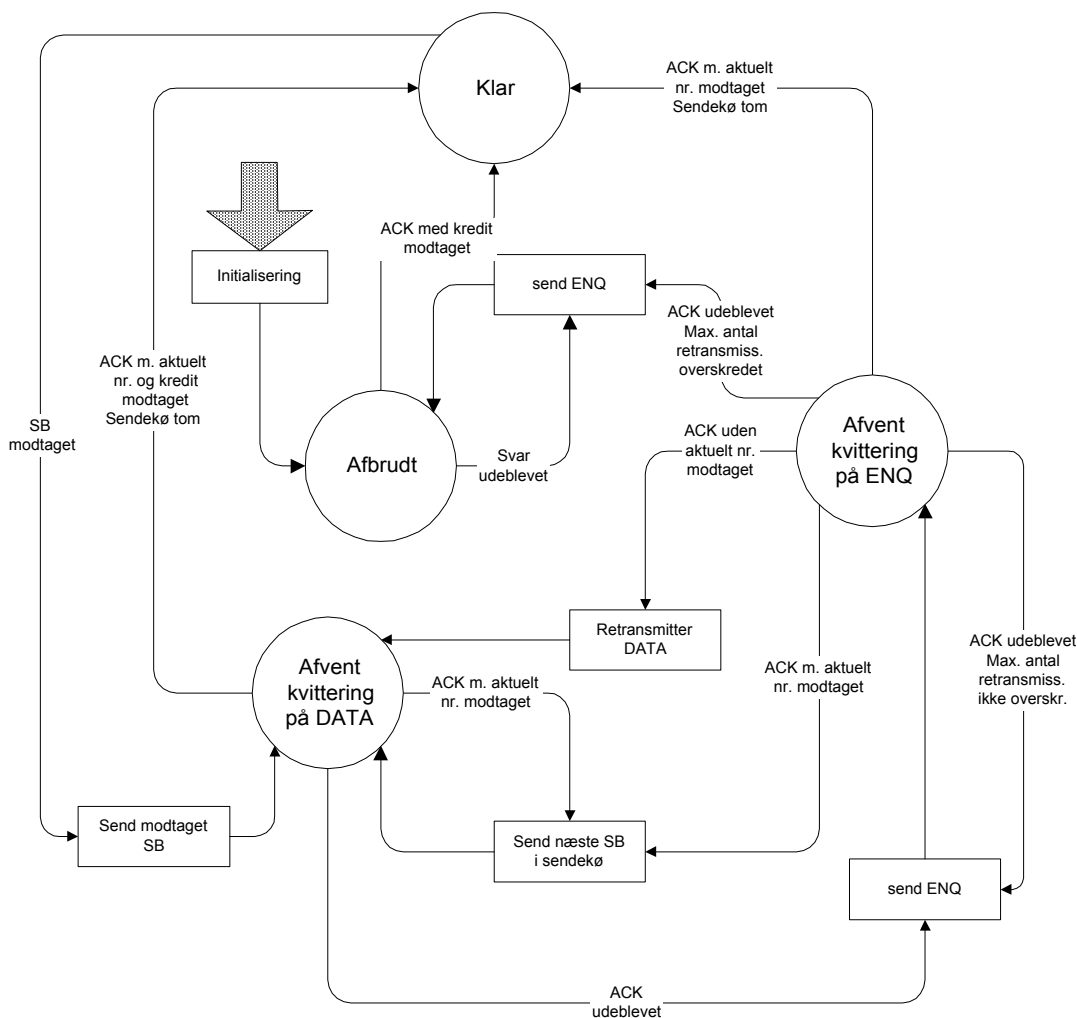


Fig. G.6: Tilstands-handlingsbeskrivelse for primær station

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 verdana-skrift**, mens pseudokode-delene er skrevet med **<kursiveret verdana i kantede parente-**

ser>. 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	2	3	4
	A	B	B	A
B: Klar	5	6	7	8
	C	B	B	B
C: Afvent kvittering på DATA	9	10	11	12
	C	C	B, C	A, D
D: Afvent kvittering på ENQ	13	14	15	16
	D	C	B, C	A, D

Fig. G.7: Tilstands-handlingstabel for primær station

	Primær station Initialisering
Handling	<pre> { sekvensnr = 1; tilstand = afbrudt; <send ENQ> <start ENQ timeout periode> antal_transmissionsforsøg = 1; } </pre>

Tilstand:	afbrudt
Begivenhed:	SB (der er modtaget en sendebuffer fra brugermodul)
Handling 1:	<pre> { <returner buffer med resultat: ingen forbindelse> } </pre>

Tilstand:	afbrudt
Begivenhed:	RESET (der er modtaget RESET fra sekundærstationen)
Handling 2:	{ tilstand = klar; }

Tilstand:	afbrudt
Begivenhed:	ACK (der er modtaget en ACK fra sekundærstationen)
Handling 3:	{ sekvensnummer = <samme værdi som angivet i ACK-kvitteringen>; tilstand = klar; }

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:	<pre> { sekvensnummer = <næste sekvensnummer>; <send DATA-kommando med sekvensnummer> aktuel_sendebuffer = <afsendt sendebuffer>; <Start DATA timeout periode> antal_transmissionsforsøg = 0; tilstand = afvent_kvittering_på_DATA; } </pre>

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

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

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:	<pre> { hvis (<sekvensnummer i ACK-kvitteringen> == <sekvensnummer i den afsendte datakommando>) { <send aktuel sendebuffer tilbage til bruger med resultat ok> antal_transmissionsforsøg = 0 /* idet der kom svar */ hvis (antal_sendebuffere == 0) { tilstand = klar; } ellers { <Næste sendebuffer tages ud af køen> <udfør HANDLING 5> } } ellers /* ikke aktuel kvittering: synkroniseringsfejl */ { <gør intet> } } </pre>

Tilstand:	afvent_kvittering_på_DATA
Begivenhed:	Timeout (timeoutperiode er udløbet)
Handling 12:	<pre> { hvis (antal_transmissionsforsøg >= max_antal_transmissionsforsøg) { <returner aktuel sendebuffer til bruger med resultat opgivet> <send alle sendebuffer 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; } } </pre>

Tilstand:	afvent_kvittering_på_ENQ
Begivenhed:	SB (der er modtaget en sendebuffer fra brugermodul)
Handling 13:	<pre> { <sæt sendebuffer i kø til overførsel> } </pre>

Tilstand:	afvent_kvittering_på_ENQ
Begivenhed:	RESET (der er modtaget RESET fra sekundærstationen)
Handling 14:	<pre> { <retransmitter sidste DATA-kommando> antal_transmissionsforsøg += 1; <start DATA timeout periode> tilstand = afvent_kvittering_på_DATA; } </pre>

Tilstand:	afvent_kvittering_på_ENQ
Begivenhed:	ACK (der er modtaget ACK fra sekundærstationen)
Handling 15:	<pre> { hvis (sekvensnummer i ACK-kvitteringen == sekvensnummer i den afsendte datakommando) { <send aktuel SB tilbage til bruger med resultat ok> antal_transmissionsforsøg = 0; /* der kom svar */ 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; } } </pre>

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

5. 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.

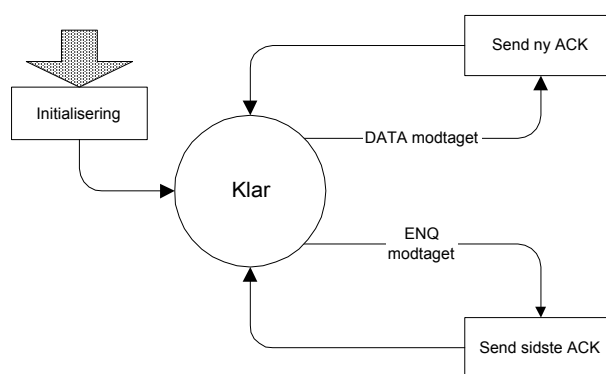


Fig. G.8: Tilstands-handlingsbeskrivelse for sekundær station

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.

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

Tilstand	Begivenhed	
	ENQ	DATA
A: Klar	1	2
	A	A

Fig. G.9: Tilstands-handlingstabel for sekundær station

	Sekundær station initialisering
Handling:	<pre>{ tilstand = klar; sidste_kvittering = RESET; }</pre>

Tilstand:	klar
Begivenhed:	ENQ (der er modtaget ENQ fra primærstationen)
Handling 1:	<pre>{ ACK_sekvensnummer = seneste_kvittering; hvis (antal_modtagebuffer > 0) { <ACK sendes til primærstation> } ellers { <gør intet> } }</pre>

Tilstand:	klar
Begivenhed:	DATA (der er modtaget DATA fra primærstationen)
Handling 2:	<pre> { hvis (antal_modtagebuffer > 0) { <kopier input til modtagebuffer og returner den til bruger> ACK_sekvensnummer = DATA_sekvensnummer; <ACK sendes til primærstation> } ellers { <gør intet> /* input ignoreres. */ } } </pre>