

SIGNAL

Open_Channel_Request, Open_Channel_Confirm, Open_Channel_Fail,
 Open_Request(t_Open_Request_paramframe), Open_Confirm(t_Open_Confirm_paramframe),
 Close_Channel_Request, Close_Channel_Confirm, Close_Channel_Fail,
 FIN(t_FINframe), FINACK(t_FINACKframe),
 Send_Packet_Request, Send_Packet_Confirm,
 Flow_Continue, Packet_Loss_Notification, Nothing_to_send(t_nothingtosendframe),
 Packet_Receive, DataFrame(t_slacp_frame), FEC_Data(t_FEC_Dataframe), ACK(t_ACKframe),
 FEC_ACK(t_FEC_ACKframe), Sender_FEC_Redundancy(t_FEC_Redundancyframe),
 Receiver_FEC_Redundancy(t_FEC_Redundancyframe), BufferAvailable,
 BufferAvail, No_MacBuf, No_MBuf, Send_Packet_Dropped, Send_Packet_Quench,
 RXMTPKT_LOSS(integer), Channel_Closing, Channel_Closed;

/* slacp frame*/

```
newtype t_slacp_frame struct
  frame_type integer;
  frame_seq_no integer;
  frame_length integer;
  ip_pkt_data_pkt;
endnewtype t_slacp_frame;
```

/* data packet */

```
syntype indexsort = integer
  constants 0:6
endsyntype;
newtype data_pkt array (indexsort, integer)
endnewtype;
```

```
newtype t_FEC_Dataframe struct
  frame_type integer;
  FEC_block_no integer;
  FEC_seq_no integer;
  Data t_slacp_frame;
endnewtype;
```

/* Queue of slacp frames */

```
newtype t_slacp_queue array(integer, t_slacp_frame)
endnewtype;
```

```
newtype t_nothingtosendframe struct
  frame_type integer;
  frame_seq_no integer;
endnewtype;
```

```
newtype t_FEC_Redundancyframe struct
  frame_type integer;
  FEC_rframe_count integer;
  Sender_Redundancy boolean;
  FEC_block_no integer;
  FEC_seq_no integer;
  redundant_data t_slacp_frame;
endnewtype;
```



system slap_protocol

2(3)



```
newtype t_sack_block struct
hole boolean;
seq_no integer;
endnewtype;
newtype t_sacks array(integer,t_sack_block);
endnewtype;
```

```
newtype t_fsack_block struct
hole boolean;
seq_no integer;
endnewtype;
newtype t_fsacks array(integer,t_fsack_block);
endnewtype;
```

```
newtype t_ACKframe struct
frametype integer;
Cum_seq_no integer;
holes_exist boolean;
sacks t_sacks;
endnewtype;
```

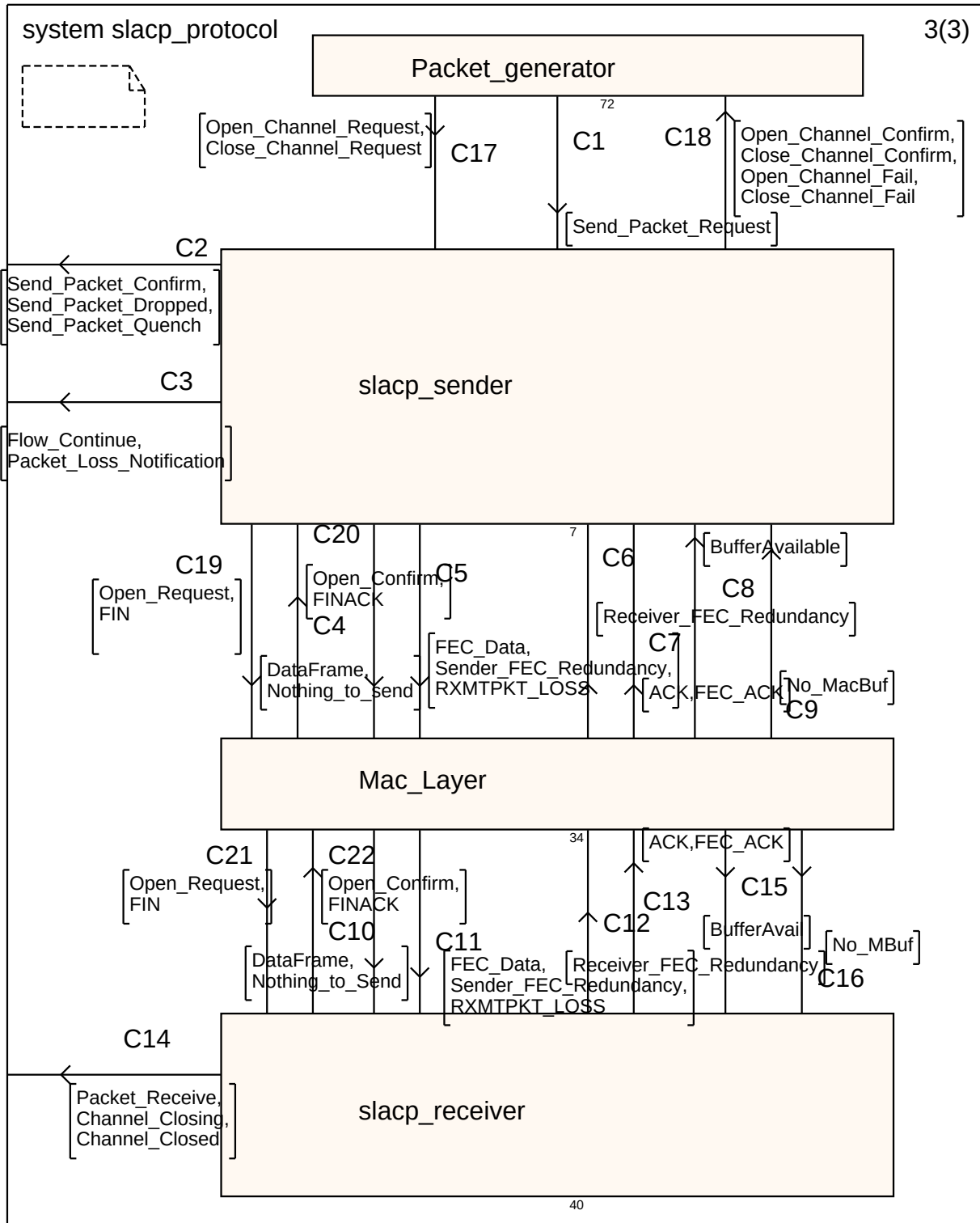
```
newtype t_FEC_ACKframe struct
frametype integer;
fec_block_no integer;
fec_seq_no integer;
holes_exist boolean;
fsacks t_fsacks;
endnewtype;
```

```
newtype t_Open_Request_paramframe struct
seq_no integer;
parameter integer;
endnewtype;
```

```
newtype t_Open_Confirm_paramframe struct
seq_no integer;
parameter integer;
endnewtype;
```

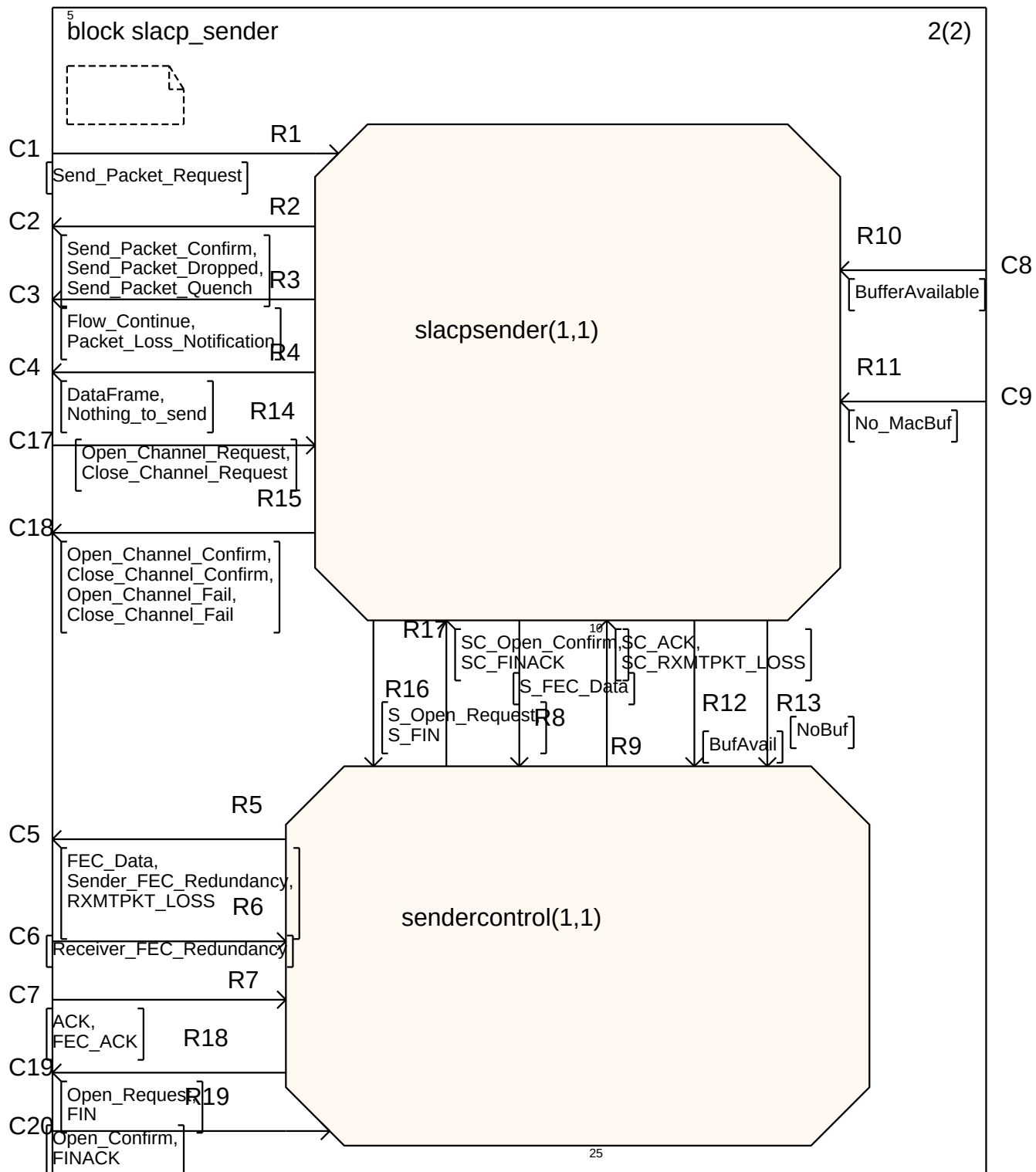
```
newtype t_FINframe struct
seq_no integer;
ACKset boolean;
endnewtype;
```

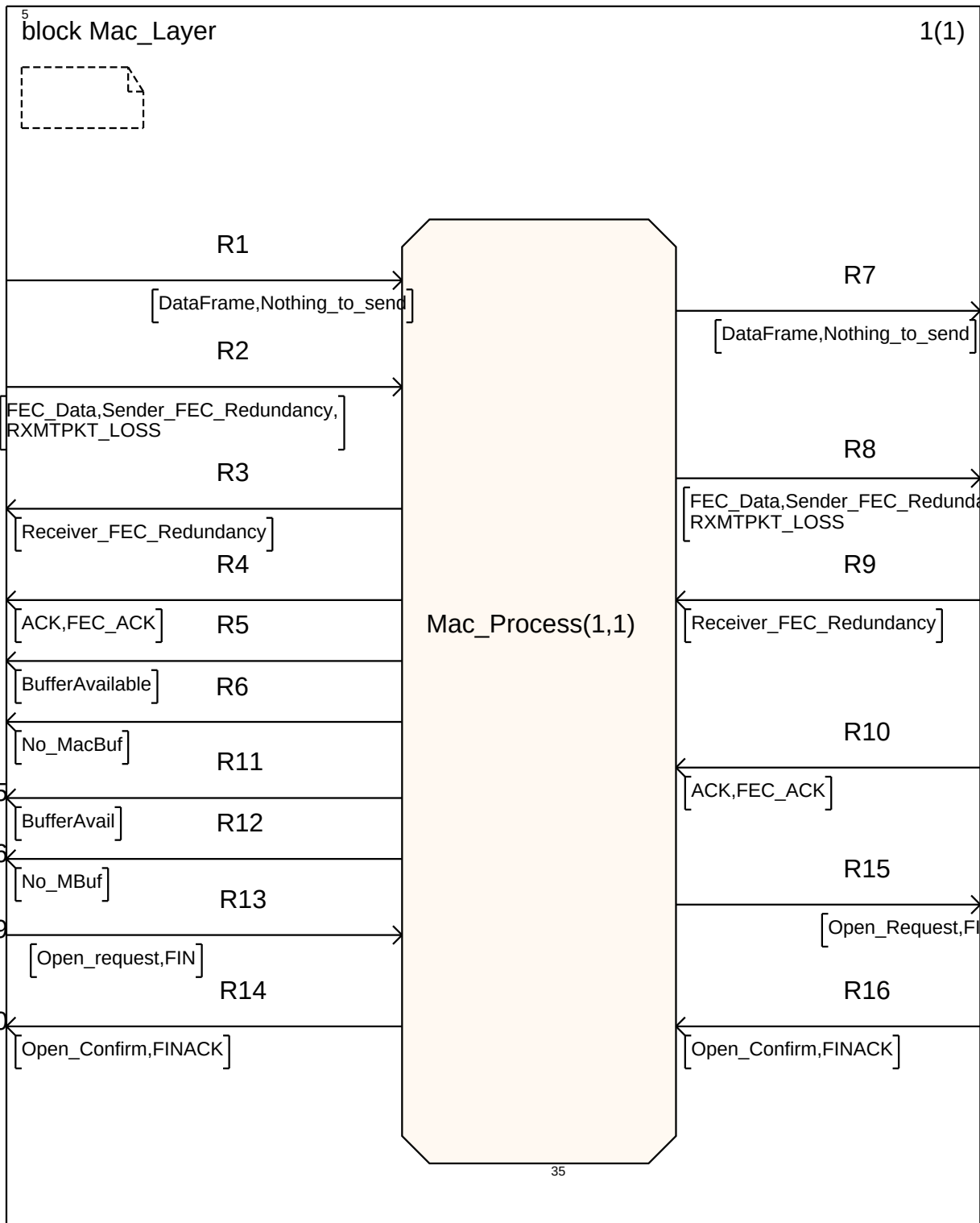
```
newtype t_FINACKframe struct
seq_no integer;
endnewtype;
```





```
SIGNAL
S_Open_Request(t_Open_Request_paramframe),
SC_Open_Confirm(t_Open_Confirm_paramframe),
S_FEC_Data(t_slacp_frame),
SC_ACK(t_ACKframe),
SC_RXMTPKT_LOSS(t_slacp_frame),BufAvail,NoBuf,
S_FIN(t_FINframe),SC_FINACK(t_FINACKframe);
```





process Mac_Process

1(5)



```

/* Mac_queue variables */
dcl Mac_Buf t_Mac_queue;
dcl Mac_queue_front integer:=0;
dcl Mac_queue_tail integer:=0;

```

```

newtype t_Macframe choice
a t_slacp_frame;
b t_ACKframe;
c t_FEC_Dataframe;
d t_FEC_Redundancyframe;
e t_FEC_ACKframe;
f t_nothingtosendframe;
g t_Open_Request_paramframe;
h t_Open_Confirm_paramframe;
i t_FINframe;
j t_FINACKframe;
k integer;
endnewtype;

```

```

Synonym Max_MacBuf integer=70;
Synonym MacLowWaterMark integer=40;
Synonym Max_Macqueue_size integer=10;
dcl slacp_frame t_slacp_frame;
dcl MacBuf_Len integer:=0;
dcl Frame_Len integer:=10;
dcl Mac_counter integer:=0;
dcl delay_duration:=1;
dcl NoBuf_on boolean:=false;
dcl Bufavail_on boolean:=true;

```

```

dcl ACKframe t_ACKframe;
dcl FEC_ACKframe t_FEC_ACKframe;
dcl FEC_Rframe t_FEC_Redundancyframe;
dcl FEC_Dataframe t_FEC_Dataframe;
dcl Macframe t_Macframe;
dcl ntframe t_nothingtosendframe;
dcl FINframe t_FINframe;
dcl FINACKframe t_FINACKframe;
dcl seq_no integer;
Timer Link_Timer ;

```

```

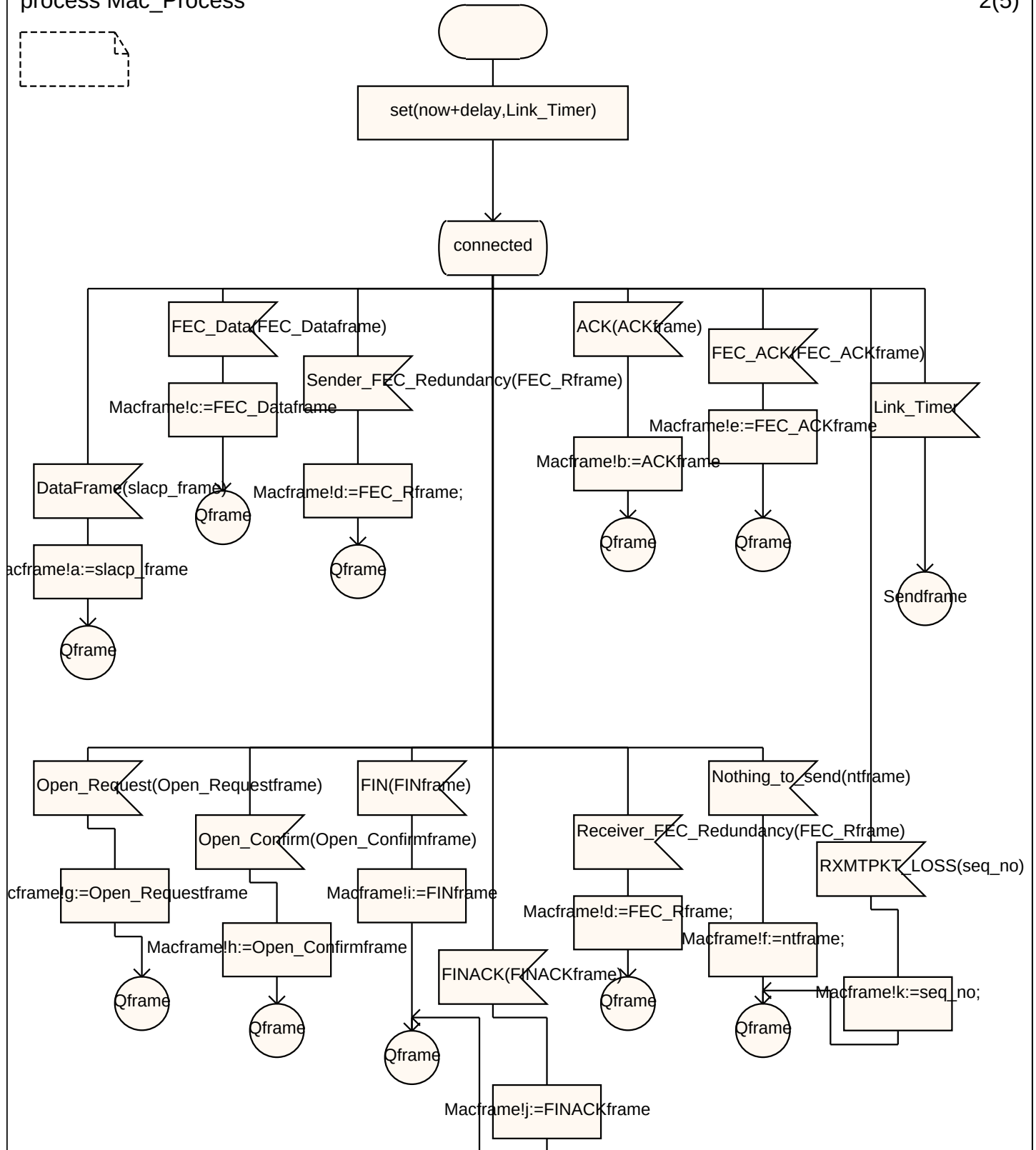
/* Queue of Mac frames */
newtype t_Mac_queue array(integer,t_Macframe)
endnewtype;

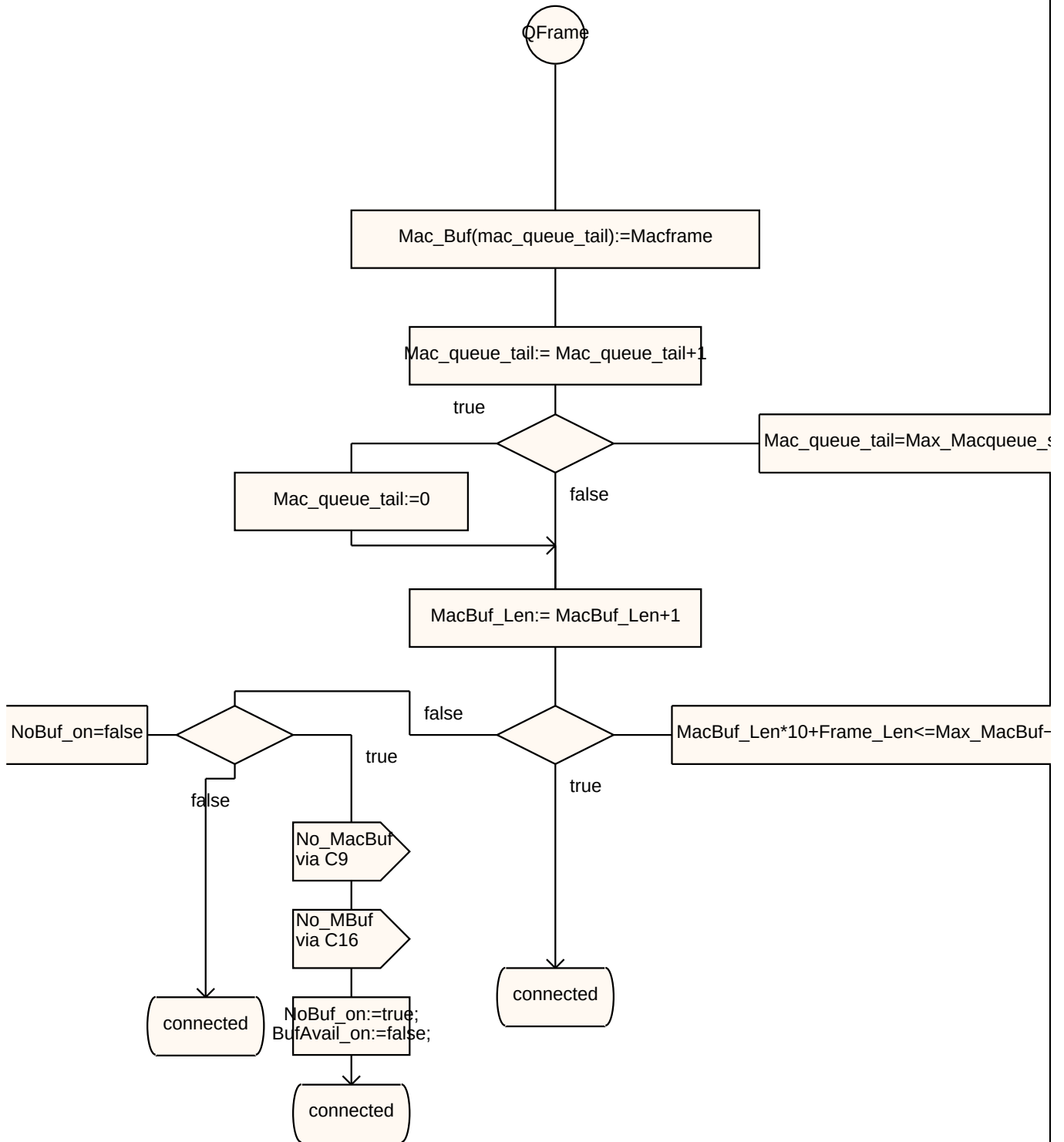
```

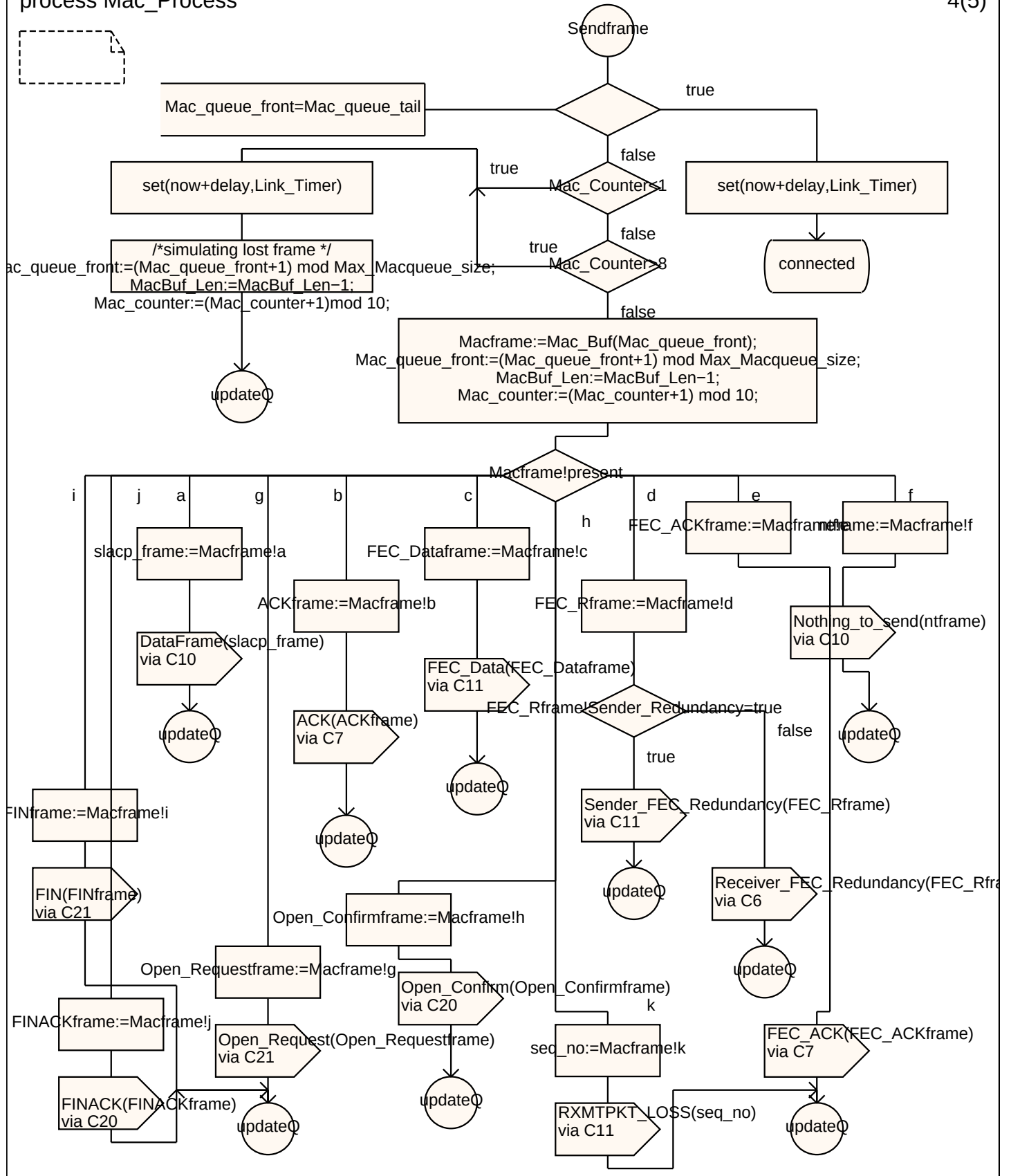
```

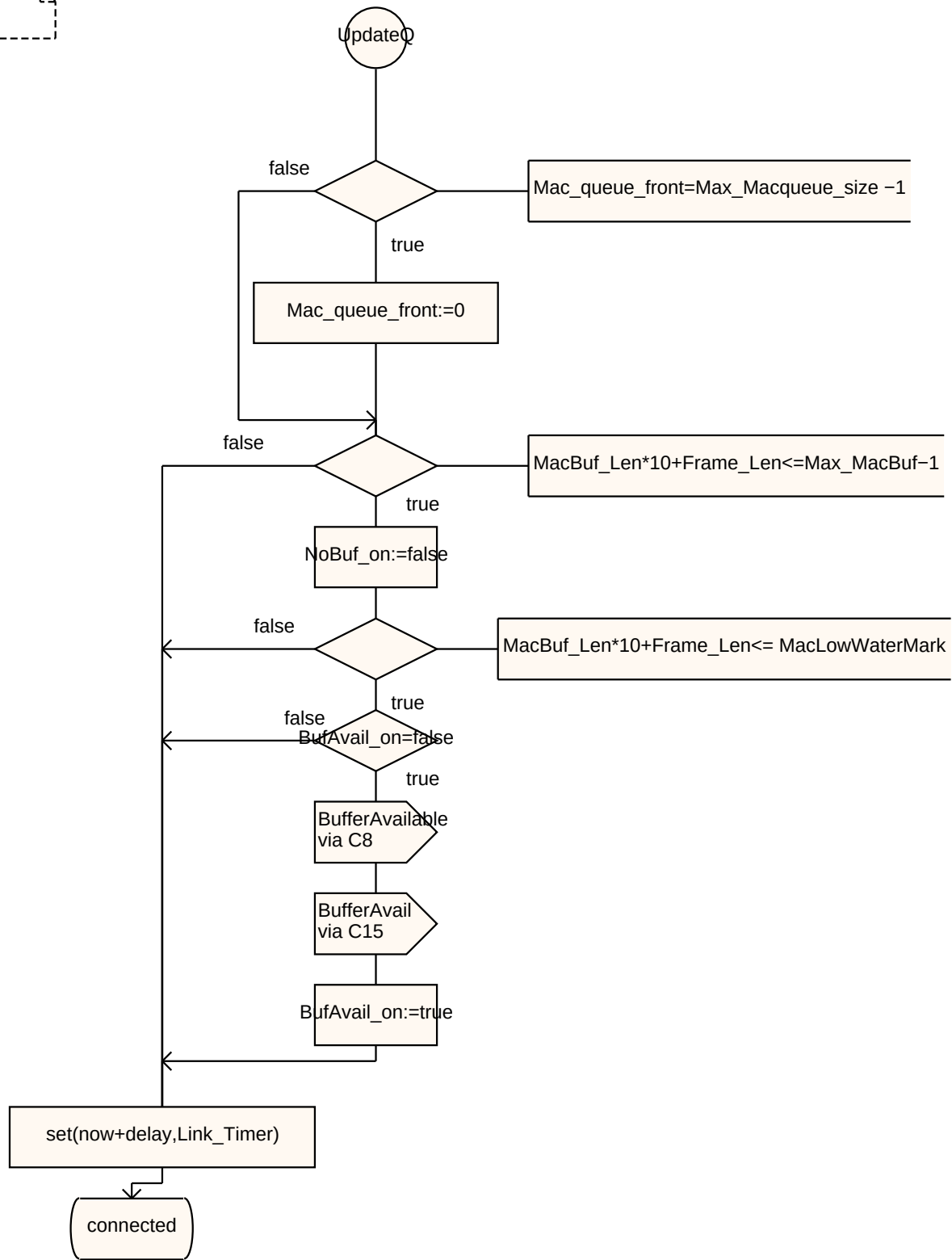
dcl Open_Requestframe t_Open_Request_paramframe;
dcl Open_Confirmframe t_Open_Confirm_paramframe;

```



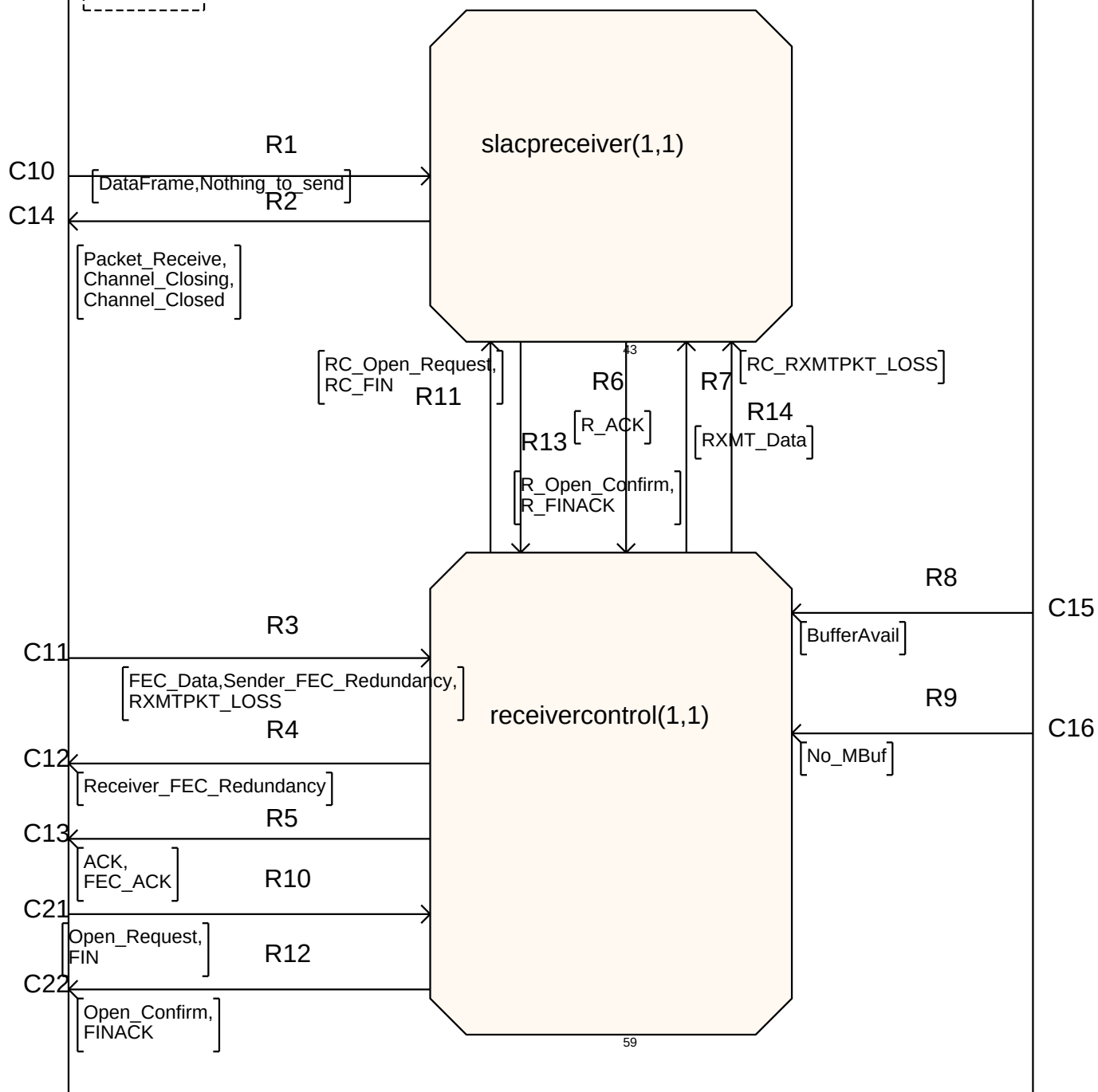






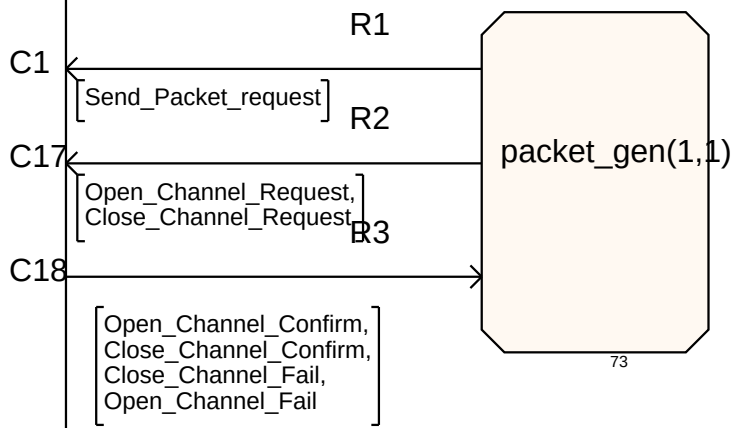


```
SIGNAL
RC_Open_Request(t_Open_Request_paramframe),
R_Open_Confirm(t_Open_Confirm_paramframe),
R_FINACK(t_FINACKframe), RC_FIN(t_FINframe),
R_ACK(t_ACKframe),RXMT_Data(t_slacp_frame),
RC_RXMTPKT_LOSS(integer);
```



⁵block Packet_Generator

1(1)

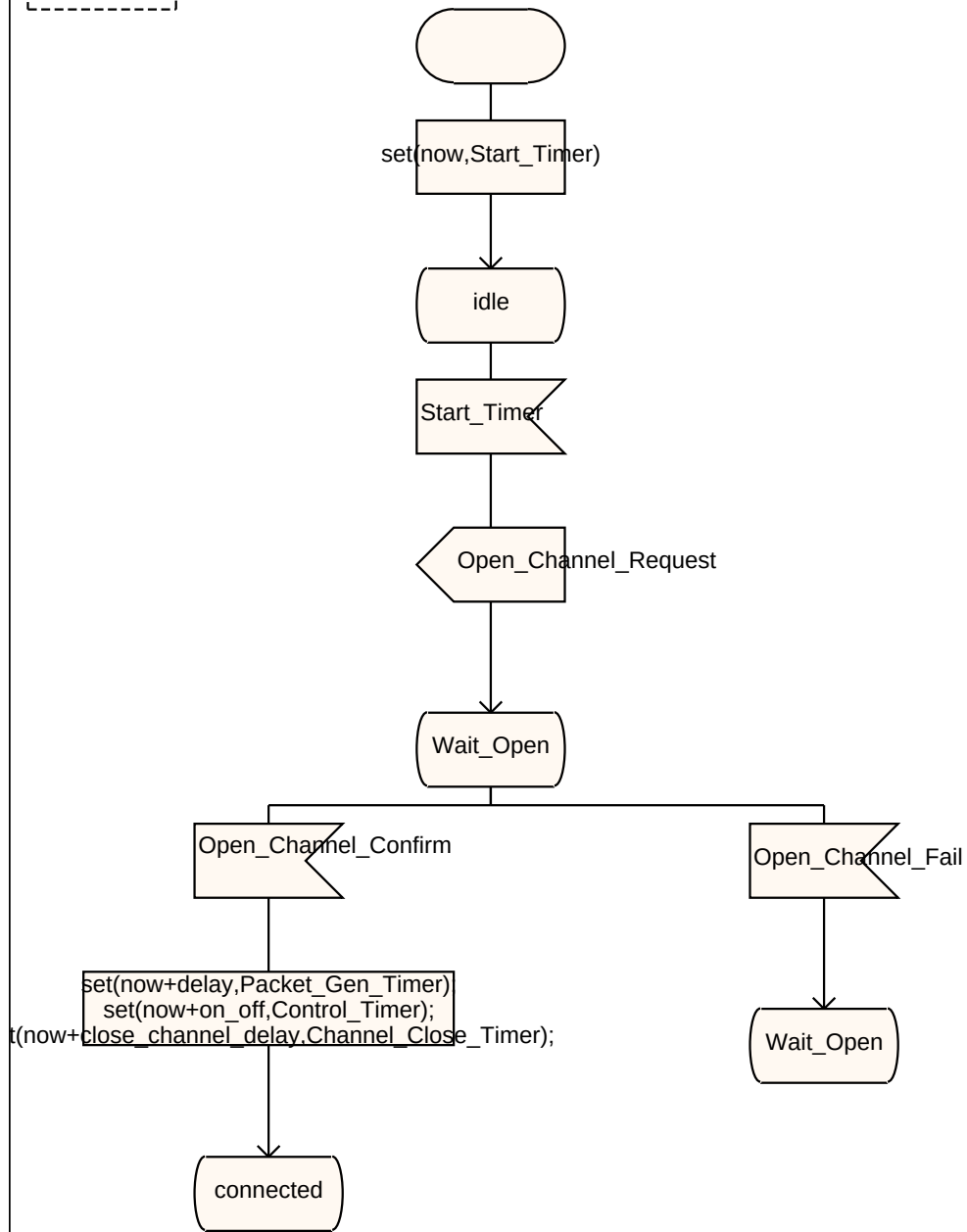


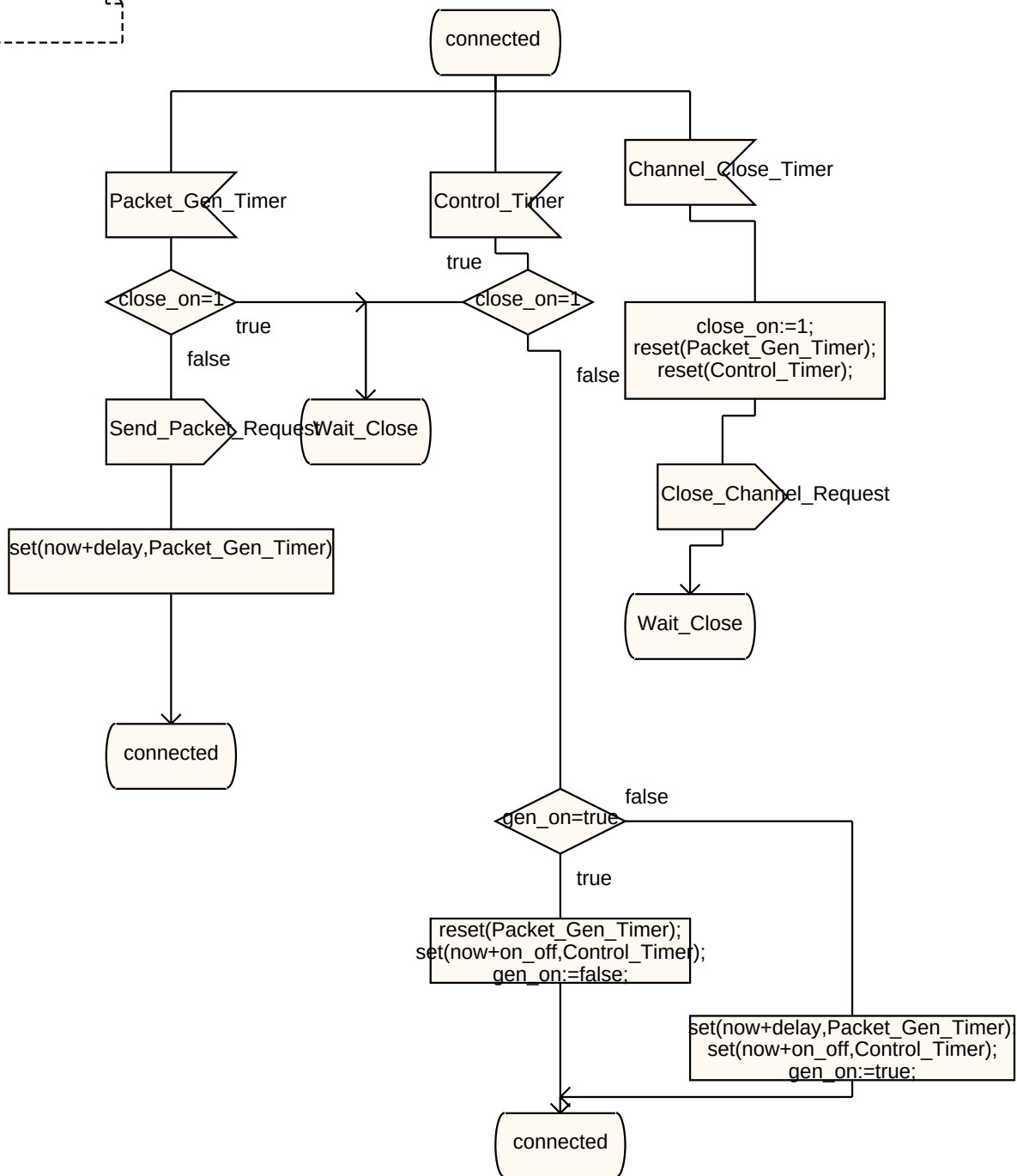
process packet_gen

1(4)



```
Timer Packet_Gen_Timer;  
dcl delay duration:=1;  
Timer Control_Timer;  
dcl on_off duration :=2;  
dcl gen_on boolean :=true;  
Timer Channel_Close_Timer;  
Timer Start_Timer;  
dcl close_channel_delay duration :=50;  
dcl close_on integer :=0;
```





process packet_gen

