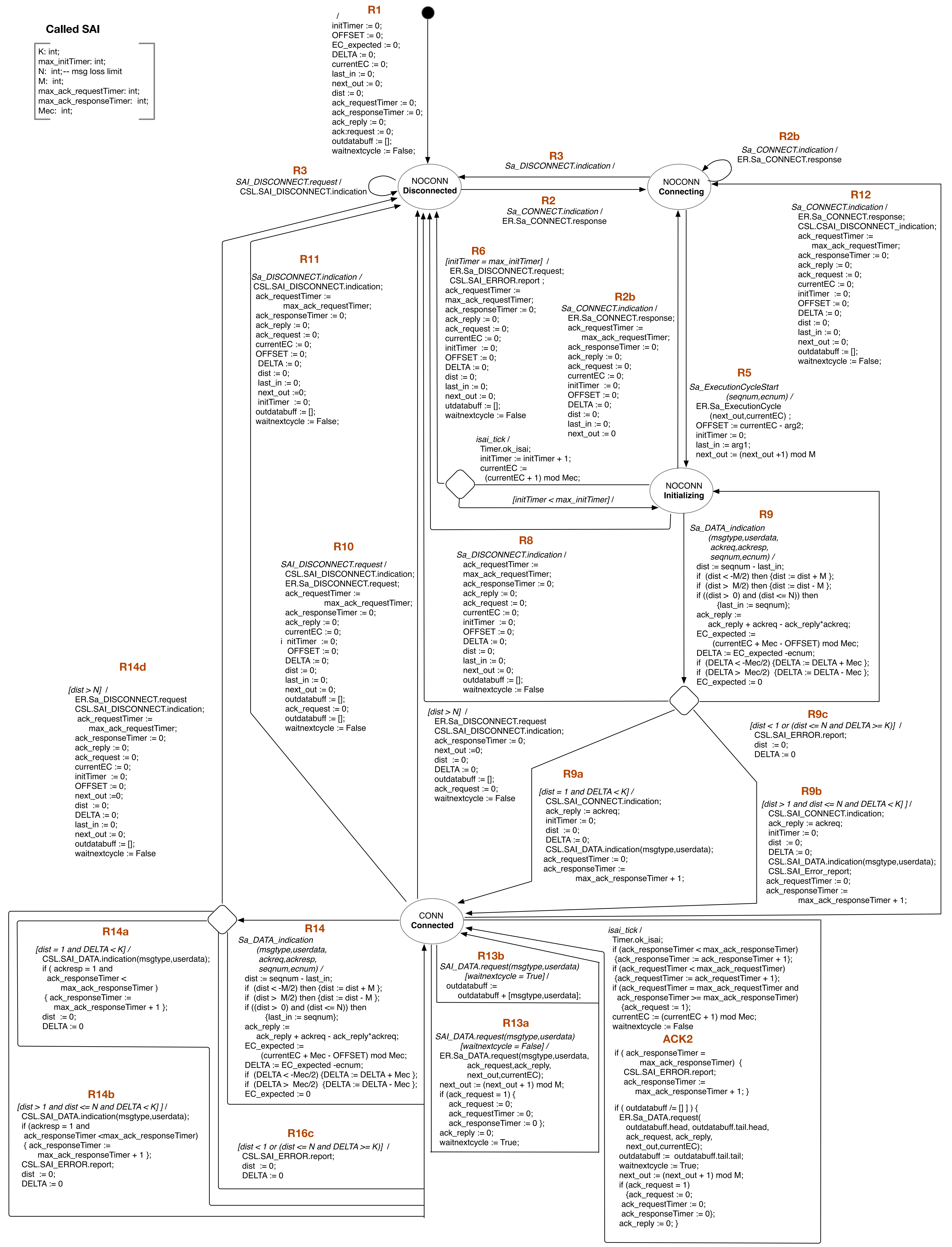

```
K: int;
max_initTimer: int;
N:  int;-- msg loss limit
M:  int;
max_ack_requestTimer: int;
max_ack_responseTimer:  int;
Mec:  int;
```

R1

```
/
initTimer := 0;
OFFSET := 0;
EC_expected := 0;
DELTA := 0;
currentEC := 0;
last_in := 0;
next_out := 0;
dist := 0;
ack_requestTimer := 0;
ack_responseTimer := 0;
ack_reply := 0;
ack:request := 0;
outdatabuff := [];
waitnextcycle := False;
```

R3

```
SAI_DISCONNECT.request /
 CSL.SAI_DISCONNECT.indication
```

R3

```
Sa_DISCONNECT.indication /
```

R2

```
Sa_CONNECT.indication /
ER.Sa_CONNECT.response
```

R2b

```
Sa_CONNECT.indication /
ER.Sa_CONNECT.response
```

R12

```
Sa_CONNECT.indication /
  ER.Sa_CONNECT.response;
  CSL.CSAI_DISCONNECT_indication;
  ack_requestTimer :=
       max_ack_requestTimer;
  ack_responseTimer := 0;
  ack_reply := 0;
  ack_request := 0;
  currentEC := 0;
  initTimer  := 0;
  OFFSET := 0;
  DELTA := 0;
  dist := 0;
  last_in := 0;
  next_out := 0;
  outdatabuff := [];
  waitnextcycle := False;
```

R11

```
Sa_DISCONNECT.indication /
 CSL.SAI_DISCONNECT.indication;
  ack_requestTimer :=
           max_ack_requestTimer;
  ack_responseTimer := 0;
  ack_reply := 0;
  ack_request := 0;
  currentEC := 0;
  OFFSET := 0;
   DELTA := 0;
   dist := 0;
   last_in := 0;
   next_out :=0;
   initTimer  := 0;
  outdatabuff := [];
  waitnextcycle := False;
```

R6

```
[initTimer = max_initTimer]  /
  ER.Sa_DISCONNECT.request;
  CSL.SAI_ERROR.report ;
ack_requestTimer :=
max_ack_requestTimer;
ack_responseTimer := 0;
ack_reply := 0;
ack_request := 0;
currentEC := 0;
initTimer  := 0;
OFFSET := 0;
DELTA := 0;
dist := 0;
last_in := 0;
next_out := 0;
utdatabuff := [];
waitnextcycle := False
```

R2b

```
Sa_CONNECT.indication /
  ER.Sa_CONNECT.response;
  ack_requestTimer :=
       max_ack_requestTimer;
  ack_responseTimer := 0;
  ack_reply := 0;
  ack_request := 0;
  currentEC := 0;
  initTimer  := 0;
  OFFSET := 0;
  DELTA := 0;
  dist := 0;
  last_in := 0;
  next_out := 0
```

R5

```
Sa_ExecutionCycleStart
        (seqnum,ecnum) /
  ER.Sa_ExecutionCycle
      (next_out,currentEC) ;
  OFFSET := currentEC - arg2;
  initTimer := 0;
  last_in := arg1;
  next_out := (next_out +1) mod M
```

```
isai_tick /
 Timer.ok_isai;
 initTimer := initTimer + 1;
 currentEC :=
   (currentEC + 1) mod Mec;
```

```
[initTimer < max_initTimer] /
```

R9

```
Sa_DATA_indication
      (msgtype,userdata,
       ackreq,ackresp,
       seqnum,ecnum) /
  dist := seqnum - last_in;
  if  (dist < -M/2) then {dist := dist + M };
  if  (dist >  M/2) then {dist := dist - M };
  if ((dist >  0) and (dist <= N)) then
         {last_in := seqnum};
  ack_reply :=
     ack_reply + ackreq - ack_reply*ackreq;
  EC_expected :=
       (currentEC + Mec - OFFSET) mod Mec;
  DELTA := EC_expected -ecnum;
  if  (DELTA < -Mec/2) {DELTA := DELTA + Mec };
  if  (DELTA >  Mec/2)  {DELTA := DELTA - Mec };
  EC_expected := 0
```

R10

```
SAI_DISCONNECT.request /
  CSL.SAI_DISCONNECT.indication;
  ER.Sa_DISCONNECT.request;
  ack_requestTimer :=
            max_ack_requestTimer;
  ack_responseTimer := 0;
  ack_reply := 0;
  currentEC := 0;
i  nitTimer  := 0;
   OFFSET := 0;
  DELTA := 0;
  dist := 0;
  last_in := 0;
  next_out := 0;
  outdatabuff := [];
  ack_request := 0;
  outdatabuff := [];
  waitnextcycle := False
```

R8

```
Sa_DISCONNECT.indication /
  ack_requestTimer :=
  max_ack_requestTimer;
  ack_responseTimer := 0;
  ack_reply := 0;
  ack_request := 0;
  currentEC := 0;
  initTimer  := 0;
  OFFSET := 0;
  DELTA := 0;
  dist := 0;
  last_in := 0;
  next_out := 0;
  outdatabuff := [];
  waitnextcycle := False
```

```
[dist > N]  /
  ER.Sa_DISCONNECT.request
  CSL.SAI_DISCONNECT.indication;
  ack_responseTimer := 0;
  next_out :=0;
  dist  := 0;
  DELTA := 0;
  outdatabuff := [];
  ack_request := 0;
  waitnextcycle := False
```

R9c

```
[dist < 1 or (dist <= N and DELTA >= K)]  /
  CSL.SAI_ERROR.report;
  dist  := 0;
  DELTA := 0
```

R9a

```
[dist = 1 and DELTA < K] /
  CSL.SAI_CONNECT.indication;
  ack_reply := ackreq;
  initTimer := 0;
  dist  := 0;
  DELTA := 0;
  CSL.SAI_DATA.indication(msgtype,userdata);
 ack_requestTimer := 0;
 ack_responseTimer :=
              max_ack_responseTimer + 1;
```

R9b

```
[dist > 1 and dist <= N and DELTA < K] ] /
  CSL.SAI_CONNECT.indication;
  ack_reply := ackreq;
  initTimer := 0;
  dist  := 0;
  DELTA := 0;
  CSL.SAI_DATA.indication(msgtype,userdata);
  CSL.SAI_Error_report;
 ack_requestTimer := 0;
 ack_responseTimer :=
              max_ack_responseTimer + 1;
```

R14d

```
[dist > N]  /
  ER.Sa_DISCONNECT.request
  CSL.SAI_DISCONNECT.indication;
   ack_requestTimer :=
        max_ack_requestTimer;
  ack_responseTimer := 0;
  ack_reply := 0;
  ack_request := 0;
  currentEC := 0;
  initTimer  := 0;
  OFFSET := 0;
  next_out :=0;
  dist  := 0;
  DELTA := 0;
  last_in := 0;
  next_out := 0;
  outdatabuff := [];
  waitnextcycle := False
```

R14a

```
[dist = 1 and DELTA < K] /
  CSL.SAI_DATA.indication(msgtype,userdata);
  if ( ackresp = 1 and
    ack_responseTimer <
         max_ack_responseTimer )
   { ack_responseTimer :=
        max_ack_responseTimer + 1 };
  dist  := 0;
  DELTA := 0
```

R14

```
Sa_DATA_indication
      (msgtype,userdata,
       ackreq,ackresp,
       seqnum,ecnum) /
  dist := seqnum - last_in;
  if  (dist < -M/2) then {dist := dist + M };
  if  (dist >  M/2) then {dist := dist - M };
  if ((dist >  0) and (dist <= N)) then
         {last_in := seqnum};
  ack_reply :=
     ack_reply + ackreq - ack_reply*ackreq;
  EC_expected :=
       (currentEC + Mec - OFFSET) mod Mec;
  DELTA := EC_expected -ecnum;
  if  (DELTA < -Mec/2) {DELTA := DELTA + Mec };
  if  (DELTA >  Mec/2)  {DELTA := DELTA - Mec };
  EC_expected := 0
```

R13b

```
SAI_DATA.request(msgtype,userdata)
       [waitnextcycle = True] /
  outdatabuff :=
     outdatabuff + [msgtype,userdata];
```

R13a

```
SAI_DATA.request(msgtype,userdata)
       [waitnextcycle = False] /
 ER.Sa_DATA.request(msgtype,userdata,
        ack_request,ack_reply,
        next_out,currentEC);
 next_out := (next_out + 1) mod M;
 if (ack_request = 1) {
    ack_request := 0;
    ack_requestTimer := 0;
    ack_responseTimer := 0 };
 ack_reply := 0;
 waitnextcycle := True;
```

```
isai_tick /
 Timer.ok_isai;
 if (ack_responseTimer < max_ack_responseTimer)
  {ack_responseTimer := ack_responseTimer + 1};
 if (ack_requestTimer < max_ack_requestTimer)
  {ack_requestTimer := ack_requestTimer + 1};
 if (ack_requestTimer = max_ack_requestTimer and
   ack_responseTimer >= max_ack_responseTimer)
    {ack_request := 1};
 currentEC := (currentEC + 1) mod Mec;
 waitnextcycle := False
```

ACK2

```
 if ( ack_responseTimer =
           max_ack_responseTimer)  {
    CSL.SAI_ERROR.report;
    ack_responseTimer :=
        max_ack_responseTimer + 1; }

 if ( outdatabuff /= [] ] ) {
   ER.Sa_DATA.request(
     outdatabuff.head, outdatabuff.tail.head,
     ack_request, ack_reply,
     next_out,currentEC);
   outdatabuff :=  outdatabuff.tail.tail;
   waitnextcycle := True;
   next_out := (next_out + 1) mod M;
   if (ack_request = 1)
     {ack_request := 0;
    ack_requestTimer := 0;
    ack_responseTimer := 0};
   ack_reply := 0; }
```

R14b

```
[dist > 1 and dist <= N and DELTA < K] ] /
 CSL.SAI_DATA.indication(msgtype,userdata);
 if (ackresp = 1 and
  ack_responseTimer <max_ack_responseTimer)
  { ack_responseTimer :=
       max_ack_responseTimer + 1 };
 CSL.SAI_ERROR.report;
 dist  := 0;
 DELTA := 0
```

R16c

```
[dist < 1 or (dist <= N and DELTA >= K)]  /
  CSL.SAI_ERROR.report;
  dist  := 0;
  DELTA := 0
```