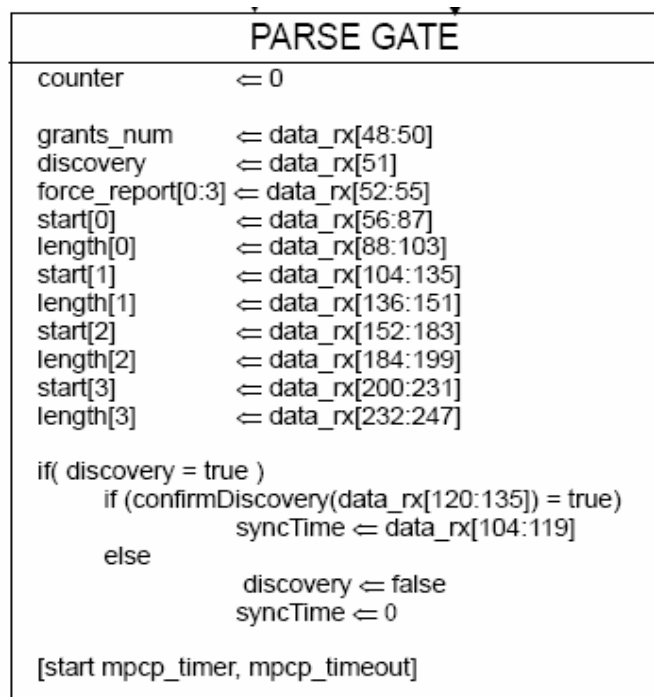


Rationale for modifications to state diagram in Figure 93-28

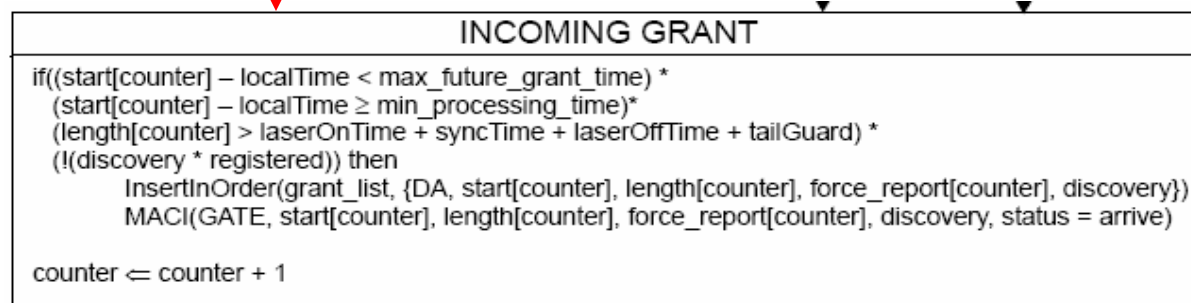
Marek Hajduczenia, Nokia Siemens Networks S.A.

Current (D1.3) of Fig. 93-28



- Parse incoming GATE MPCPDU
- decide whether the carried grants should be accepted and inserted into grant list or not

- if the size / start time of the grant is OK and the grant was received when
 - ONU is unregistered and discovery grant was received
 - ONU is registered and regular grant was received



Current (D1.3) of Fig. 93-28

- an issue was identified with the **if** statement. Observe the Karnaugh map under the assumption that the slot size and start time is OK i.e. **(start[counter] – localTime < max_future_grant_time) * (start[counter] – localTime ≥ min_processing_time) * (length[counter] > laserOnTime + syncTime + laserOffTime + tailGuard)** is true

	discovery	registered	!(discovery & registered)
1	0	0	1
2	0	1	1
3	1	0	1
4	1	1	0

```
INCOMING GRANT
if((start[counter] – localTime < max_future_grant_time) *
(start[counter] – localTime ≥ min_processing_time)*
(length[counter] > laserOnTime + syncTime + laserOffTime + tailGuard) *
(!(discovery * registered)) then
  InsertInOrder(grant_list, {DA, start[counter], length[counter], force_report[counter], discovery})
  MACI(GATE, start[counter], length[counter], force_report[counter], discovery, status = arrive)
counter ← counter + 1
```

- ONU is not registered and still could accept a GATE
- probably not dangerous (no LLID in RS anyway)
- if possible, plug this hole when correcting Fig. 93-28 state diagram

Current (D1.3) of Fig. 93-28

PARSE GATE	
counter	← 0
grants_num	← data_rx[48:50]
discovery	← data_rx[51]
force_report[0:3]	← data_rx[52:55]
start[0]	← data_rx[56:87]
length[0]	← data_rx[88:103]
start[1]	← data_rx[104:135]
length[1]	← data_rx[136:151]
start[2]	← data_rx[152:183]
length[2]	← data_rx[184:199]
start[3]	← data_rx[200:231]
length[3]	← data_rx[232:247]
if(discovery = true)	
if (confirmDiscovery(data_rx[120:135]) = true)	
syncTime ← data_rx[104:119]	
else	
discovery ← false	
syncTime ← 0	
[start mpcp_timer, mpcp_timeout]	

- issue identified in PARSE GATE state:
- imagine an ONU with 1G upstream
 - OLT schedules 10G discovery window
 - ONU cannot register in this window:
 - confirmDiscovery returns FALSE
 - discovery ← FALSE
 - registered ← FALSE
 - INCOMING GRANT state accept such a grant (see previous slide)
 - solution needed to avoid such situation

Proposed changes to Fig. 93-28

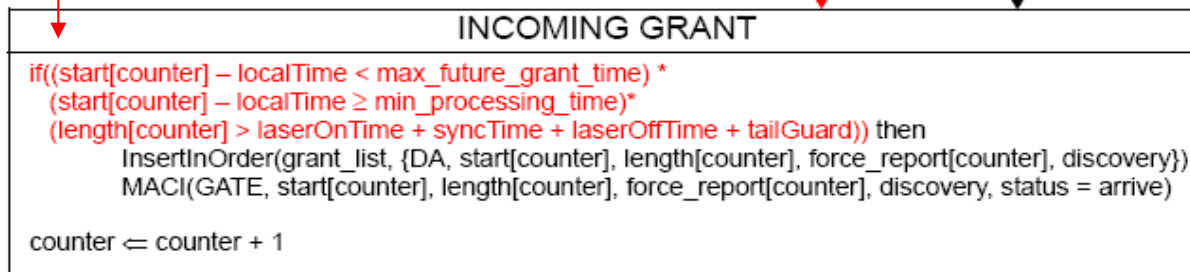


Changes proposed to PARSE GATE state:

- a new Boolean flag **gate_accepted** to indicate whether the GATE grant(s) are to be accepted or dropped
- conditions:
 - if the ONU is not registered and the incoming GATE is a Discovery GATE and confirmDiscovery evaluates to TRUE > set gate_accepted to TRUE and fetch syncTime
 - if the ONU is registered and the incoming GATE is a regular granting GATE > set gate_accepted to TRUE
 - otherwise gate_accepted is kept with FALSE value
- based on gate_accepted, either WAIT or INCOMING GRANT state is called

Current (D1.3) of Fig. 93-28

- the conditional statement is simplified, since the incorrect GATEs are filtered at the end of the PARSE GATE state;
- the if statement can be changed to **if ((start[counter] – localTime < max_future_grant_time) * (start[counter] – localTime ≥ min_processing_time) * (length[counter] > laserOnTime + syncTime + laserOffTime + tailGuard))**
 - only slot size and slot start time needs to be checked
 - insertion into the grant list is performed exactly like in D1.3
 - exit conditions from INCOMING GRANT state are exactly like in D1.3



The whole state diagram is presented in 3av_0805_hajduczenia_6.pdf