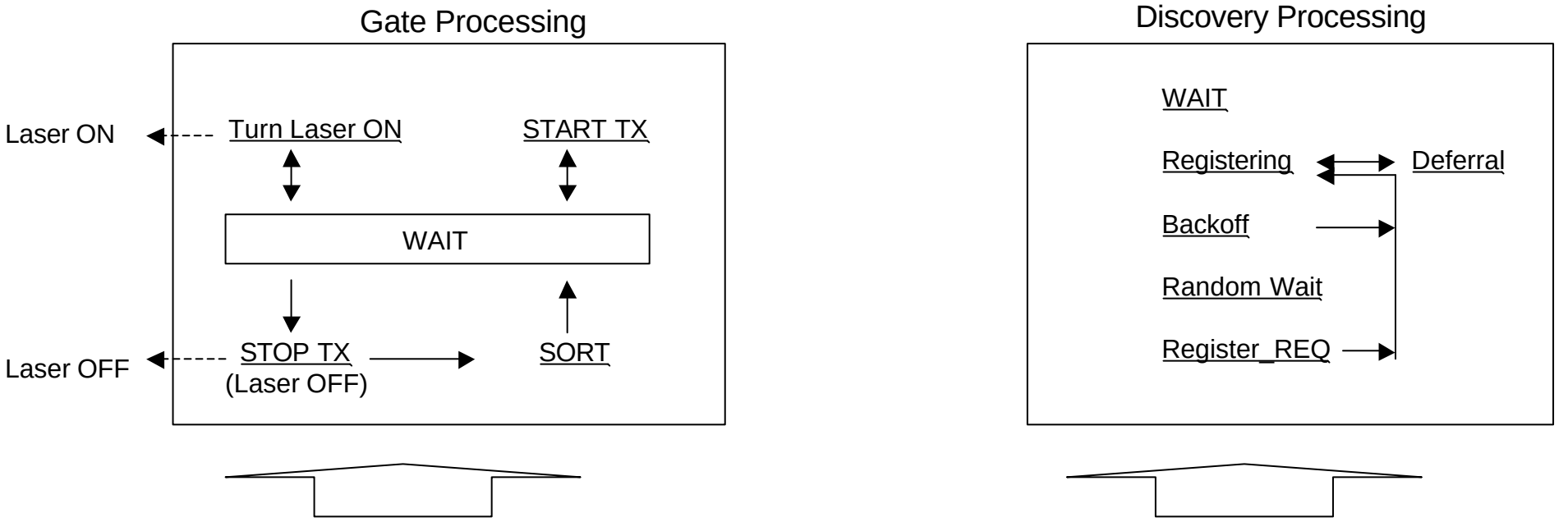
“Laser Control” into Discovery Processing State Diagram

Yasuo Ogura NTT
Yukihiro Fujimoto NTT

Current state diagram : No relationship between Gate and Report

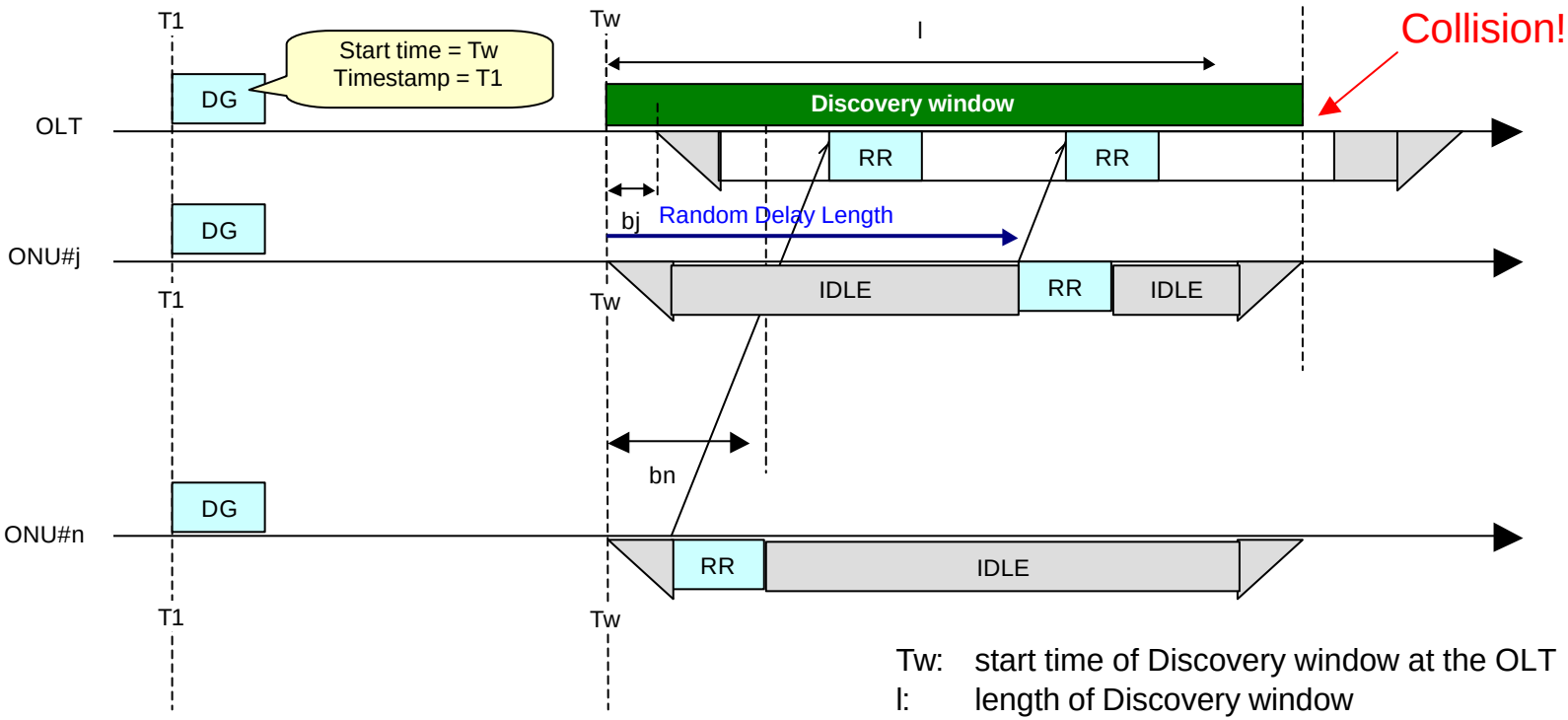


Gate Processing control “Laser ON/OFF”.

Report Processing decide to send REGISTER_REQ MPCPDU or not.

There is no relationship between “Gate Processing” and “Report Processing”, so that Laser will be ON/OFF independent to send Register_REQ or not.

Problem : Collision in the Discovery window

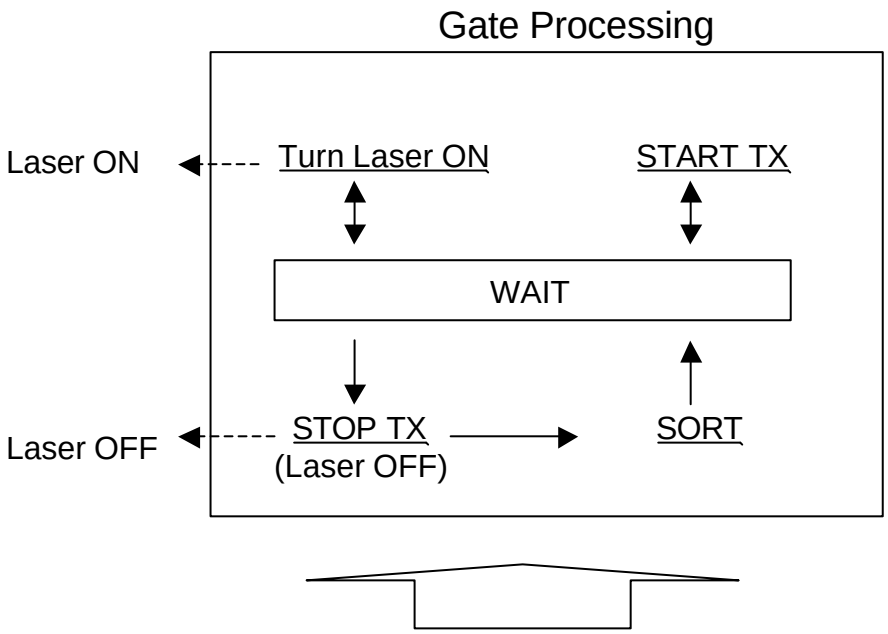


Tw: start time of Discovery window at the OLT
l: length of Discovery window
bj: round trip propagation delay of ONU#j
bn: round trip propagation delay of ONU#n

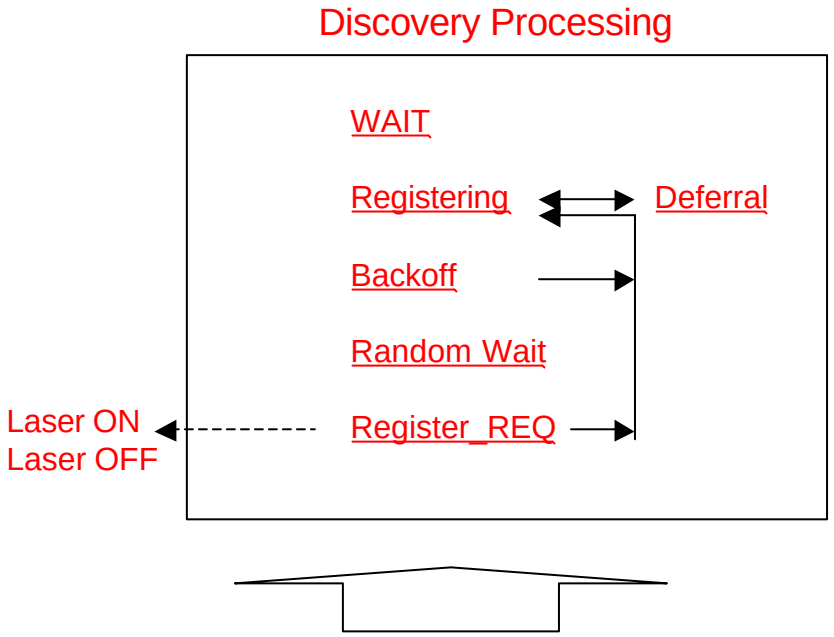
DG: Discovery GATE
RR: Register_REQ

When ONU#j and ONU#n turn Laser ON, collision will happen in the Discovery window.
It seems difficult to process AutoDiscovery successfully when multiple ONUs are power ON.

Proposed state diagram : Discovery Processing controls Laser ON/OFF



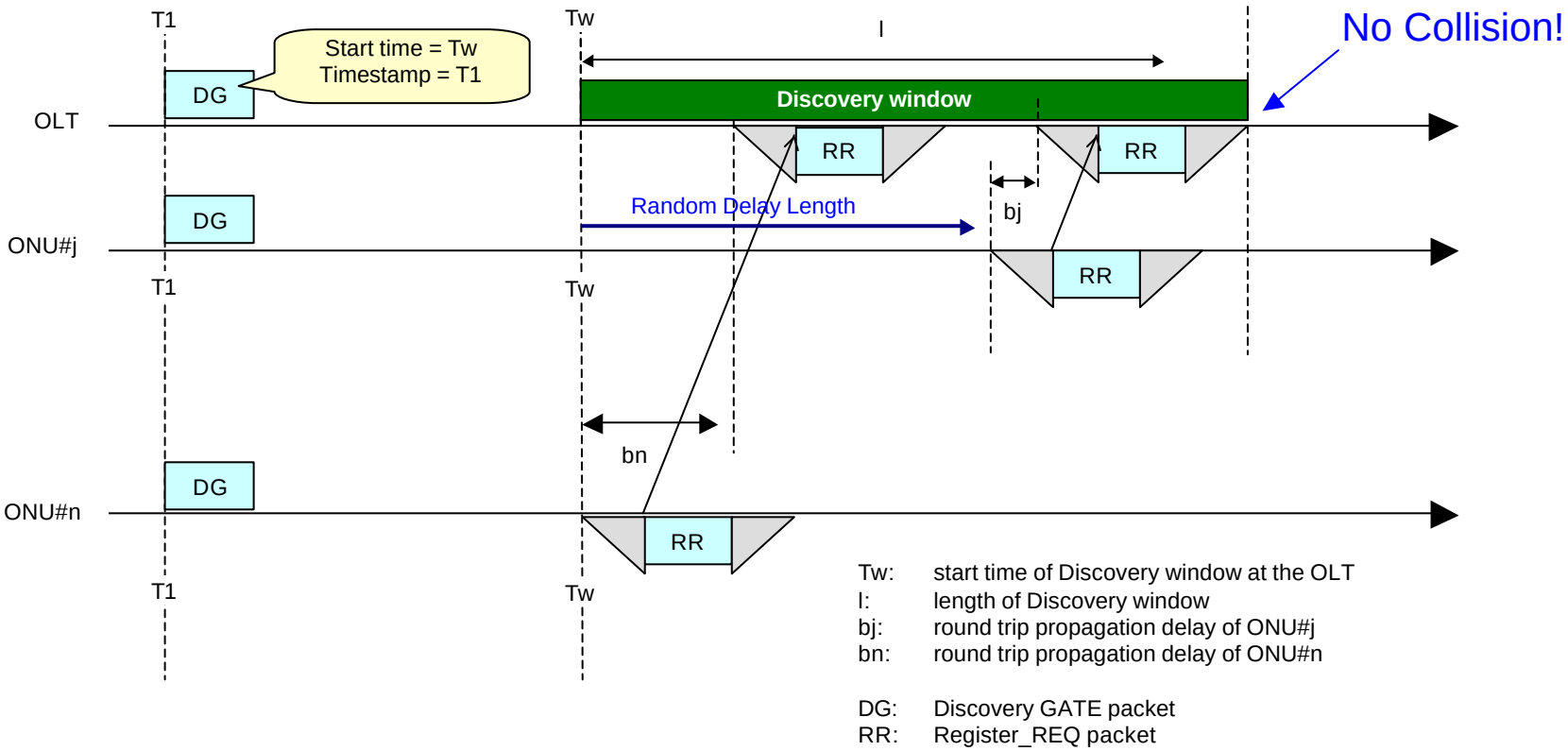
Gate Processing control “Laser ON/OFF” for Normal Gate.



Report Processing sends REGISTER_REQ MPCPDU and controls Laser ON/OFF for Register.

Turn laser ON while ONU is sending a Register_REQ only.

Solution:Laser ON/OFF during Register_REQ



Turn laser ON while ONU is sending a Register_REQ only.
It seems easy that OLT receive multiple Register_REQ in a single Discovery Window.

Proposal: OMP Parser/Multiplexer RX State Diagram

P.126 56.3.3.1.6 State Diagram

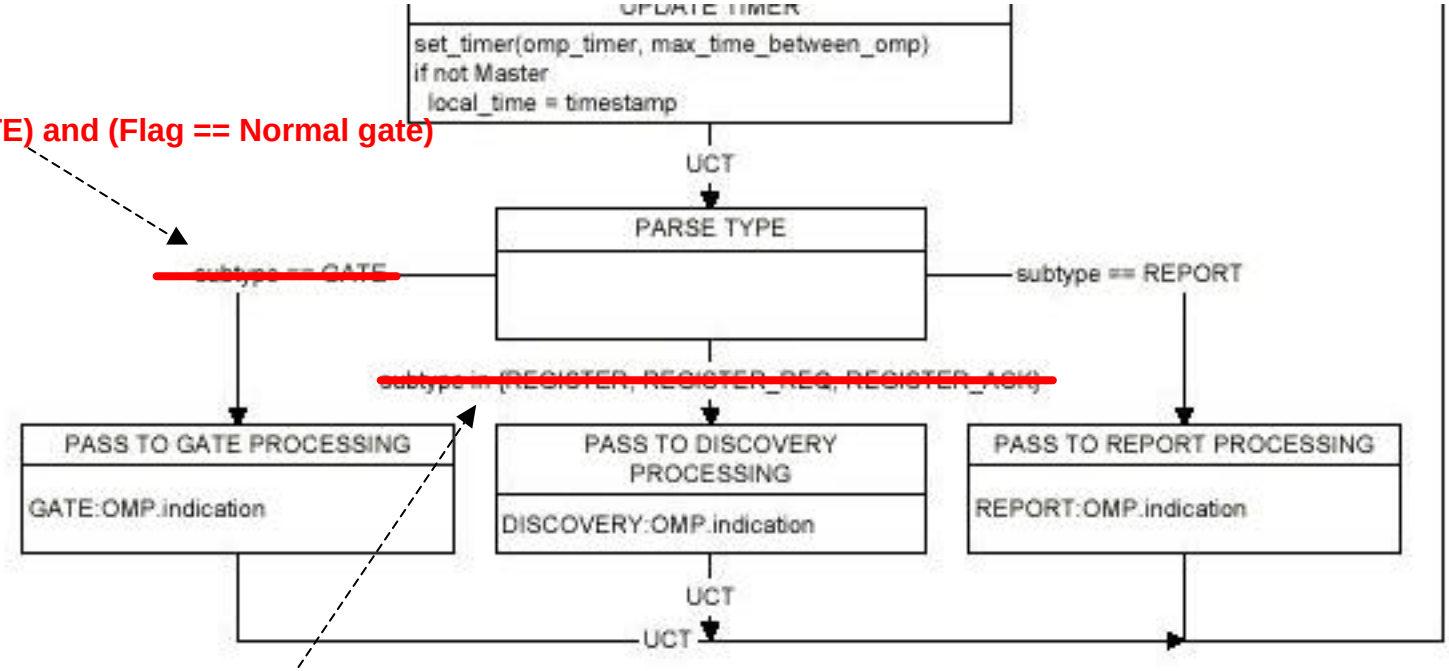


Figure 56–13—OMP Parser/Multiplexer RX State Diagram

(Subtype == GATE) and (Flag == Normal gate)

(subtype in { REGISTER, REGISTER_REQ, REGISTER_ACK})
or
(Subtype == GATE) and (Flag == Discovery gate)

Proposal : Discovery Processing Slave State Diagram 1

P.135 56.3.4.1.6 State Diagram

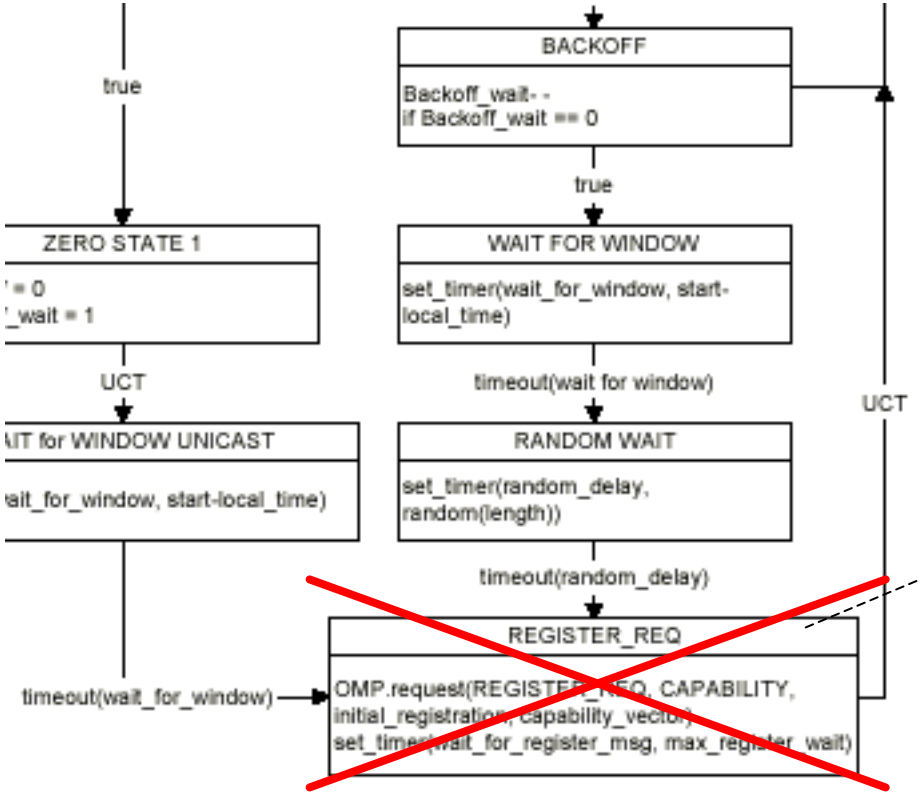


Figure 56–17—Discovery Processing Slave

timeout(wait_for_window)

timeout(random_delay)

TURN LASER ON

LaserControl = true
set_timer(IDLE_timer, IDLE_time)

timeout(IDLE_timer)

START TX

TXAllow = true

set_timer(grant_window, register_req_length)

REGISTER_REQ

OMP.request(REGISTER_REQ, CAPABILITY, initial_registeration, capability_vector)

set_timer(wait_for_register_msg, max_register_wait)

timeout(grant_window)

STOP TX

LaserControl = false
TXAllow = false

56.3.4.1.2 Variables

……..
……..
……..

LaserControl

This variable is used to control the transmit path. It is set to true when the transmit path is enabled, and is set to false when the transmit path is being shut down. LaserControl is always on for the OLT, except when disabled, and changes its value according to the state of the Gate Processing sublayer.

TYPE: on/off
DEFAULT VALUE: off for ONU
on for OLT

TXAllow

This variable is used to control PDU forwarding in the transmit path. It is set to true when the transmit path is enabled, and is set to false when the transmit path is being shut down. TXAllow is always true for the OLT, and changes its value according to the state of the Gate Processing sublayer.

TYPE: boolean
DEFAULT VALUE: false for ONU
true for OLT

IDLE_time

This variable holds the time required to stabilize the receiver at the OLT. It counts in time_quanta units from the point where transmission output is stable to the point where it is decodable.
This value is set following registration, as it is broadcast by the OLT.

TYPE: 32 bit unsigned
DEFAULT VALUE: 00-00-00-10 (256 nano seconds)

56.3.4.1.1 Constants

……

……

……

register_req_length

This constants is used for setting the grant_window timer. It represent the length of a REGISTER_REQ MPCPDU and a preamble.

TYPE: 32 bit unsigned

DEFAULT VALUE: 00-00-00-24(64+8 bytes)

56.3.4.1.4 Timers

……

……

IDLE_timer

This timer is used to wait for the event signaling the end of the period where no PDUs are allowed transmission inside the grant window. This period, where only IDLE symbol-pairs are transmitted is used to allow clock synchronization acquisition for the receiving entity.

VALUE:

grant_window

This timer is used to wait for the event signalling the end of a grant window.

VALUE: