Dynamic Bandwidth Allocation in Ethernet PON

Chan Kim, Tae Whan Yoo

ETRI

Need for DBA

- Efficient use of upstream bandwidth is needed to provide quality of service for mixture of upstream traffic

- In EPON, ONU cannot send upstream data if not granted by the OLT

- Network operators can add more customers due to increased efficiency

- Customers can enjoy enhanced services requiring bandwidth peaks beyond traditional fixed bandwidth

- Quality of service is related to delay and loss, and for upstream traffic, it is determined by the grant allocation (sometimes, delay causes loss)

Need to specify DBA ?

- Is DBA a target for standardization?
    - Yes for the framework
    - No for the allocation algorithm
- The "hook" for DBA should be designed to accommodate a wide range of DBA algorithms
- The current baselines limits the possibility of efficient DBA algorithms

Problem in Current baselines

- reports and gates don't match
  - report is for maximum 8 priority queues
  - but gate is aggregate for 8 such queues
  - gate usage for queues service is up to ONU
  - Is this enough?

- report is not always initiated by the OLT but also can be initiated by ONUs
  - Some algorithms may use systematic, managed report gathering and not want ONU-initiated reports.
  - There should be a way to prevent these ONU-initiated reports.(ex: set during auto-discovery)
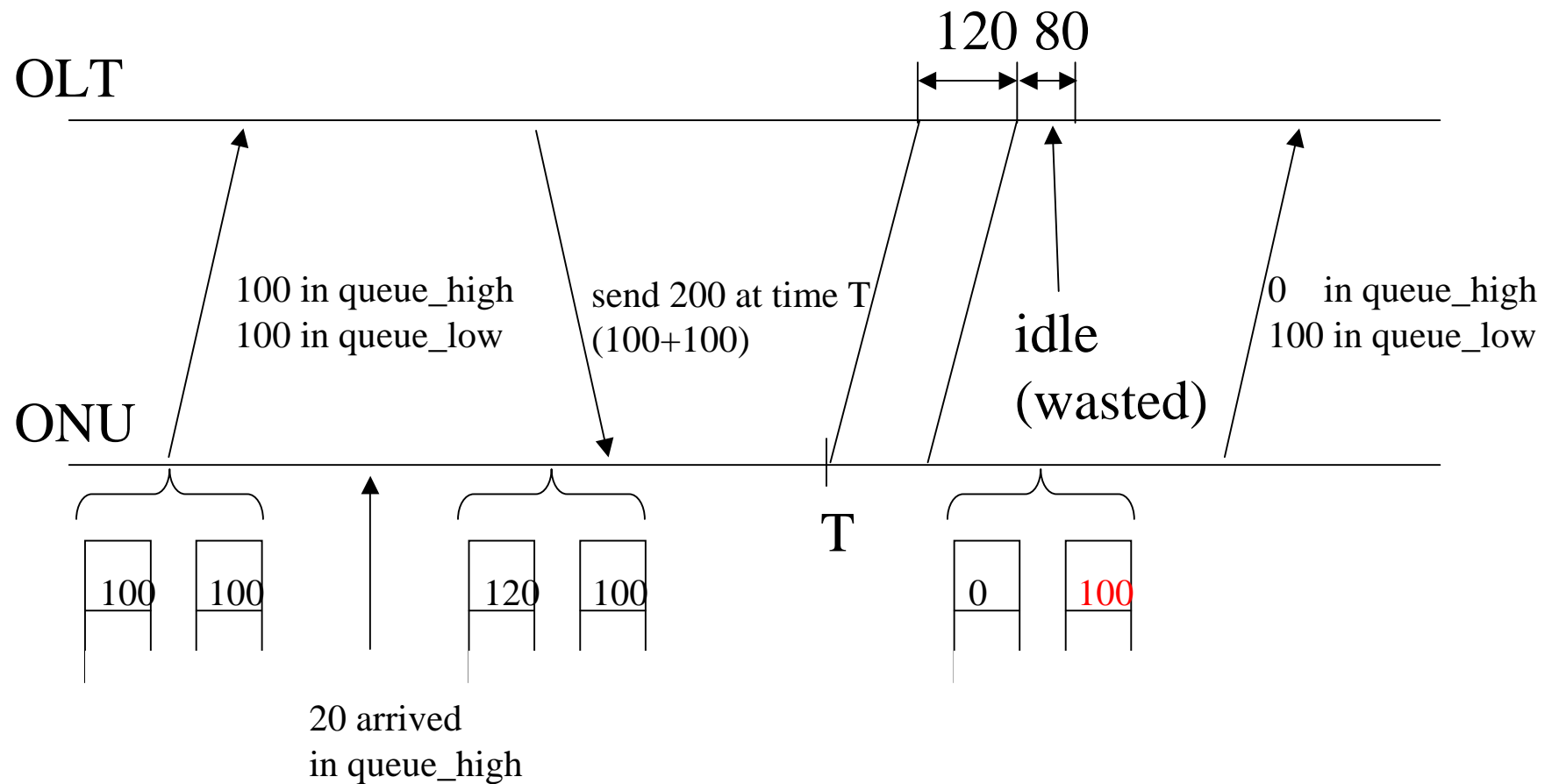
Problem in EPON DBA

- in ONU, at the reporting time or gate application time

  queue level = last reported bytes + newly arrived bytes – serviced bytes after the previous report

  (for each priority queue)

=> The OLT should know the serviced bytes of each priority queue to extract the new arrivals from the report

=> But the ONU does not know OLT's algorithm and uses the received gates in its best discretion

=> This in turn, makes the OLT not know how much were serviced for each priority queue at the ONU between reports

=> false assignment between priority => a vicious cycle!

Problem example

OLT

120 80

ONU

100 in queue_high
100 in queue_low

send 200 at time T
(100+100)

idle
(wasted)

0   in queue_high
100 in queue_low

T

100   100

120   100

0   100

20 arrived
in queue_high

excluding unreported data from service?

- Of course, there are ways to prevent unreported data from being serviced
  - remembering "up to where each queue was reported"
  - This will make the ONU design more complex than simply modifying the gate to have explicit priority

- But this isn't enough because ONU doesn't know OLT's algorithm!
  - within reported data, OLT doesn't know which queues have been serviced with the gate (even if we exclude unreported data from the service)

What is proposed here

- modify the report and gate to incorportate explicit E-PON specific "class queues"
  - report on individual class queue of ONU(vLink)
  - gate on individual class queues of ONU(vLink)
  - Note : 802.1D priority queues are already aggregate for the users. The SLA should be met by effort's of all the intermediate nodes.

- Assuming that the gate type includes "gate for report", report method should be negotiated during auto-discovery too.

Need for intermediate queue level

- There can be different types of OLTs and ONUs
  - supported number of priority queues in ONU
  - supported number of priority in OLT algorithm
  - ONU's grant usage capability (scheduling is needed here too)

- Using a logical class queues to accommodate all OLT, ONU types and all traffic types
  - for common objects of DBA algorithms (different vendors..)
  - for algorithm simplicity

- The logical queues can be named "Traffic Class(TC)" or another. Only TC is used in report and gate.

- Actual 1Q priority queues can remain separate as before (whether it's many FIFOs or shared memory). The new logical queues are used only in reporting and using gates.

TC mapping proposal

- Adopt similar scheme as in ATM-PON(ITU-T G.983.4)

- common DBA parameters can be negotiated between OLT and ONUs

- 802.1D priority queues are mapped to 5 TCs

- The TC definition can be modified to better meet the Ethernet traffic requirement (less stringent than ATM?)
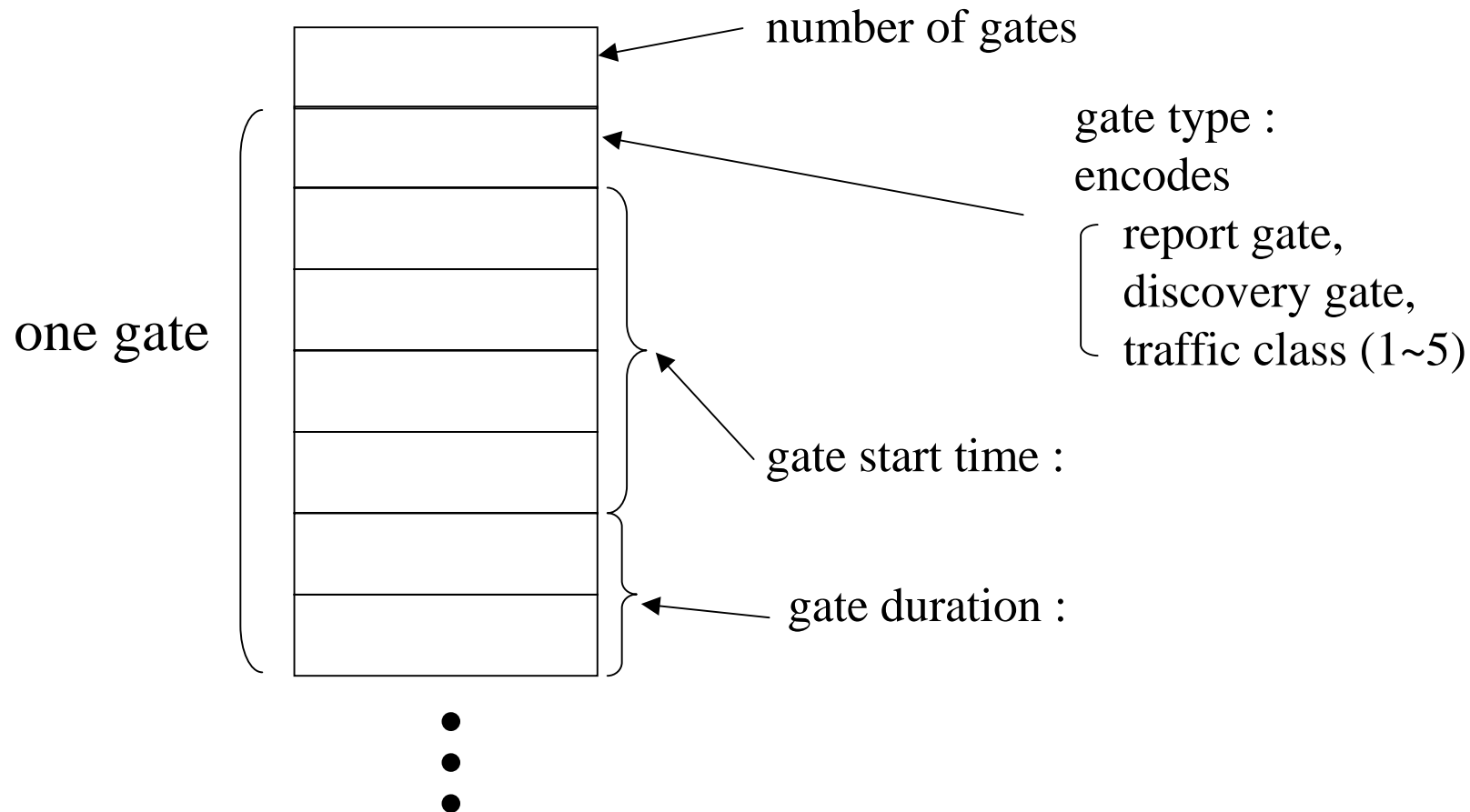
TC mapping example (G.983.4)

	Delay Sensitive	Assignment Type	Applicable T-CONT Type				
			Type 1	Type 2	Type 3	Type 4	Type 5
Fixed	Yes	Provisioning	X				X
Assured	No	Provisioning		X	X		X
Non-assured	No	Dynamic			X		X
Best-effort	No	Dynamic				X	X
Maximum	No	Provisioning			X	X	X

Proposed Gate Format change

number of gates

gate type :
encodes

report gate,
discovery gate,
traffic class (1~5)

one gate

gate start time :

gate duration :

Need for report mode setting

- "Gate for Report" Suggested in the May meeting
- But it allows the ONU to report on its own
- Some algorithms may want the controlled report only
- This would be very convinient for OLT's scheduling.(for most algorithms I think)
- Another suggestion : put the allowed report mode in the OLT capability in the gate (for discovery) message

Conclusion-1

- It is impossible to avoid algorithm failure with current baseline proposals

- This "report and gate with explicit traffic class indication" prevents the algorithm failures. (what ever the algorithm is)

- Allowed report method should be negotiated during the auto-discovery using OLT's capability field

Conclusion-2

- An intermediate traffic class queue is desirable in DBA for algorithm simplification and flexibility
  - For OLTs and ONUs with different DBA capabilities
  - For common object for various algorithms (think of OLTs and ONUs from different vendors)
  - The mapping of physical queues to the EPON traffic class can be programmable

- This suggestion is not an bandwidth allocation algorithm but a "hook" for EPON DBA.