Design and implementation of an EPON DBA algorithm

Chan Kim, Tae Whan Yoo, Yool Kwon, Bong Tae Kim

ETRI, 161 Gajeong-dong Yuseong-goo Daejeon 305-350 Korea

1. EPON and DBA

Ethernet-PON(EPON) is the most promising access network due to its internet friendliness and low cost nature. EPON comprises an OLT in the central office and many ONU/ONTs in the residential area that are connected to the OLT in PON(passive optical network) topology using passive splitter(Figure.1).

The OLT should arbitrate the upstream transmission of the ONUs and allocate upstream bandwidth resource to them. The upstream bandwidth usage request is delivered using report frames containing the amount of data waiting in the upstream queues of ONUs and the grant information is delivered to the ONUs using downstream gate frames containing grant in the form of (start time, length) pair. There are timers at OLT and ONUs and timestamp is used to synchronize this report/grant operation. In ETRI, we developed an EPON master, slave chipset and trial service was held in the city of Gwang-joo. The DBA(Dynamic Bandwidth Allocation) employs a simple cycle-based water-filling algorithm but it's very stable and provides high performance of over 92% upstream bandwidth usage.

2. Water-Filling DBA Algorithm

The DBA scheme is cycle based and static short gates are generated for each registered ONUs to make sure reports are collected every cycle and the remaining time is dynamically allocated to the ONUs according to the reports. Allocating the remaining time to the ONUs is like water-filling in that unit length is subtracted from the available time and added to each ONU in a cyclic fashion until the requests are all satisfied or there is no available time left. CPU-generated gate frames are also considered because they will consume the same grant resource. This algorithm can also limit the maximum gate length and can guarantee minimum gate length set for each ONU. The water-filling is done for 2 phases of processing minimum guaranteed length and aggregate requested length. Figure 2 shows the example of the allocation result using the proposed and implemented dynamic gate allocation algorithm. It should be noted that by controlling gate length for each cycle, the bandwidth is also controlled.

3. Implementation of the DBA algorithm

The core of the DBA gate generation logic is composed of 4 engines each processing 16 ONUs' requests and a combining block which gathers information from the engines and provides aggregate information needed for each engine's processing(Figure 3). Total 64 ONUs can be processed with the DBA processing logic. DBA water-filling starts after reports are read and max, min lengths are setup. Each engine processes 16 ONUs' requests at the same pace. They look at the common available gate length resource, subtracting and adding unit length to their ONUs. The 4 engines process with same ONU index and same phase(minimum guaranteed processing phase, or aggregate request processing phase). Figure 4 shows the flow chart of the engines.

4. Performance

DBA performance depends on the number of ONUs, traffic type, and ONU queue size. Upstream delay is about 2 ms due to report and gate processing latency. The aggregate total upstream bandwidth is more affected by the overhead than DBA algorithm is always over 92% and can be deemed almost ideal. For 64 ONUs, the reports take 6.4 % of upstream bandwidth.

For each ONU, upstream throughput is limited by the ONU's queue size. In steady state for constant rate, when cycle is T and the traffic rate is R(bits/s), the gate turn-on time is $T*(R/R_L)$ where R_L is the line rate of 1Gbps. During the gate off time, the buffer rises at the rate of R for duration $T(1-R/R_L)$. The maximum rate without loss can be calculated by setting the maximum buffer level to ONU's buffer size. That is, $RT(1-R/R_L) = B$. When B and T is known, we can calculate the value of R. For example, when B = 128Kbits, the value of R is 146Mbps and this matches with the experiments.

5. Conclusion

With simple cycle based water filling algorithm and its implementation, stable DBA performance of over 92% upstream bandwidth for up to 64 ONUs was achieved without the DSP or CPU. It is stable since over-allocation is prevented by nature. This parallel, sequential approach is quick enough with control over maximum and minimum guaranteed bandwidth for each ONU by controlling the gate length every cycle.

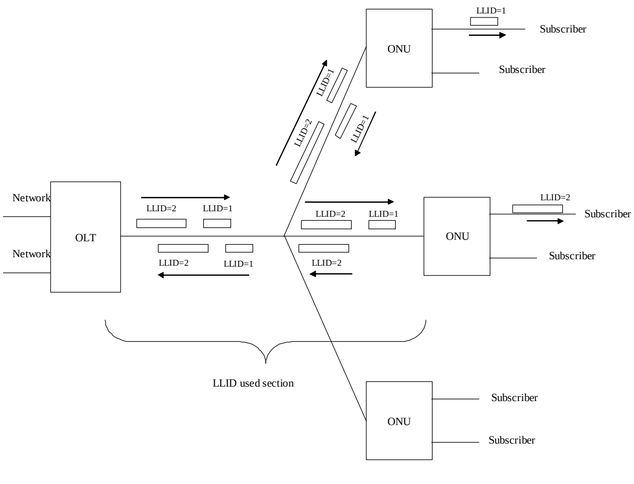
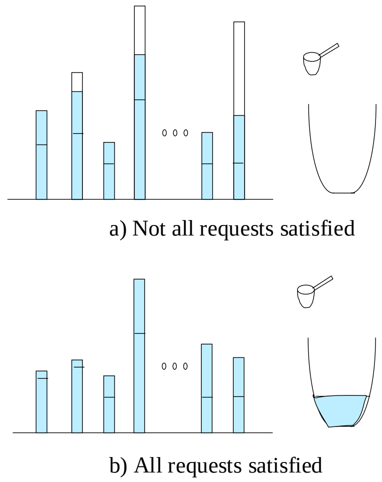


Figure 1. Usage of LLID in EPON



a) Not all requests satisfied

b) All requests satisfied

Figure 2. Water-filling DBA algorithm

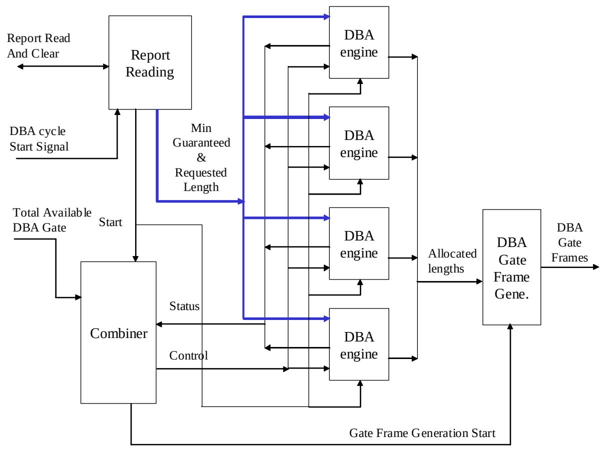


Figure 3. DBA gate generator

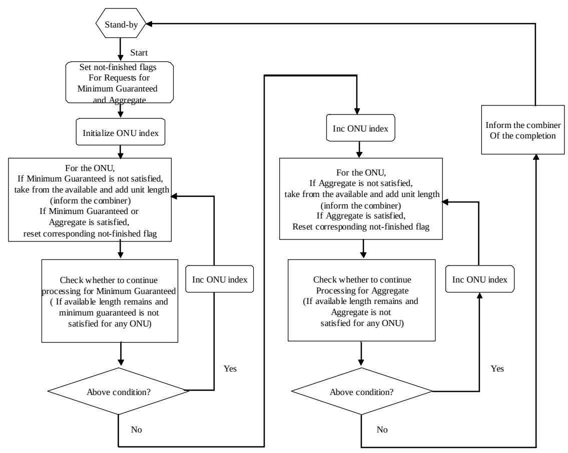


Figure 4. DBA engine's flow chart