CHAPTER 5
QUALITY OF SERVICE ANALYSIS FOR AEERG PROTOCOL

In our work, in order to improve the Quality of Service (QoS), we implemented the WFQ in a modified manner. It is termed as m-WFQ. The metrics of QoS includes efficient Bandwidth utilization, effective Power management and Minimum delay with greater throughput. By modified WFQ, all these metrics are achieved.

5.1 m-WFQ ALGORITHM

This is a modification of WFQ algorithm. It is given in Figure 5.1 implementation of m-WFQ in the proposed algorithm is explained with the help of following steps.

Figure 5.1 Flow chart of m-WFQ Algorithm
Send the packets from source node to destination node. At destination, PDR is calculated. This value is sent back to source as a feedback factor. Then the procedure is done as per following steps:

**Step 1:** At source node:

Calculate the time of arrival of Feedback value about packet delivery ratio. From the value, the source node decides

- short path to destination (Time of arrival T1)
- longer path to destination (Time of arrival T2)
- longest path to destination. (Time of arrival T3)

**Step 2**

Feedback factor value (Packet delivery ratio) is compared with Threshold value

If the value of packet delivery ratio is greater than threshold value

Intermediate sleep nodes are not triggered to active mode. Active nodes in short path to destination are used for data transmission.

Else

Intermediate nodes in the longer and longest paths are also triggered to active mode.

**Step 3**

Identified the short path is assigned with maximum weight.
Step 4

In the link, the number of path is calculated. The following procedure is applied on the path.

(i) Separate the packets which contain two different destinations address. Group them and formulate in one queue.

(ii) Determine remaining energy in the battery of every active node. If it is less than 30% of its initial energy, then their packet will be grouped into the same queue which has been already assigned with maximum weight.

(iii) Assign weight $q$ for the above specified queue (comparatively greater than any other queues and the sum of weight assigned for all queues should not exceed one) that determines how many packets are transmitted from the queue. Assuming all the packets are variable (since the packets arrive from different source nodes) in its size.

(iv) Compute the time at which the packet can complete the service using General Processor Sharing.

(v) Service packets based on time of completion of General Processor Sharing.

The scheduler maintains two variables

a) Current round number and

b) Highest per queue finish number. Current round number means number of rounds of service. Highest per queue finish number means current number of active connections

In m-WFQ, among the various queues, a packet is selected from a more weighted queue initially. The selected packet can be processed and sent through output port. This works as follows. Each arriving packet is given
virtual start and finish times. The virtual start time $S(n, j)$ and the virtual finish time $F(n, j)$ of the $n^{th}$ packet in $j^{th}$ queue are computed as follows:

$$S(n, j) = \max (F(n-1, j), V\{a(n, j)\})$$

$$F(n, j) = S(n, j) + \frac{L(n, j)}{r(j)} \text{ with } F(0, j) = 0,$$

where $a(n, j)$ and $L(n, j)$ are the arrival time and the length of the packet respectively. $r(j)$ is the rate at which packets are processed.

$m$-WFQ supports the fair distribution of bandwidth for variable-length packets by approximating a generalized processor sharing system. While generalized processor sharing is a theoretical scheduler that cannot be implemented, its behavior is similar to a weighted bit-by-bit round-robin (WRR) scheduling discipline. In a WRR scheduling discipline the individual bits from packets at the head of each queue are transmitted in a WRR manner. This approach supports the fair allocation of bandwidth, because it takes packet length into account (Srinivasan Keshav 1991). As a result, at any moment in time, each path receives its configured share of output port bandwidth. Transmitting packets from different paths as one bit at a time can be supported by a TDM network, but it cannot be supported by a statistically multiplexed network. If we assume that there is a packet reassembler at the far end of the link, then the assembly of full packet depends upon the way in which the last bit of each packet is transmitted. This is referred to as the packet's finish time.

In $m$-WFQ, let us consider a weighted bit-by-bit round-robin scheduler servicing three queues. Assume that queue 1 is assigned 60 percent of the output port bandwidth and that queue 2 and queues 3 are each assigned 20 percent of the bandwidth. The scheduler transmits three bits from queue 1, one bit from queue 2, one bit from queue 3, and then returns to queue 1. As a result of the weighted scheduling discipline, the last bit of the 600-byte packet is transmitted before the last bit of the 300-byte packet, and the last bit of the
300-byte packet is transmitted before the last bit of the 450-byte packet. This causes the 600-byte packet to finish (complete reassembly) before the 300-byte packet, and the 300-byte packet to finish before the 450-byte packet.

m-WFQ approximates this theoretical scheduling discipline by calculating and assigning a weight to each channel. Given the bit rate of the output port, the number of active queues, the relative weight assigned to each of the queues, and the length of each of the packets in each of the queues, it is possible for the scheduling discipline to calculate and assign a finish time to each of the arriving packet. The scheduler then selects and forwards the packet that has the earliest (smallest) finish time, among all of the queued packets. It is important to understand that the finish time is not the actual transmission time for each packet. Instead, the finish time is a number assigned to each packet that represents the order in which packets should be transmitted on the output port.

In our proposed protocol, we have made three channels from source S to Destination D. According to the time of arrival of feedback factor, shortest path is estimated. That path has been assigned with maximum weight of 0.6. The other two links having time of arrival as T2, T3 are assigned with weight each as 0.2 respectively. We have simulated the packets in the shortest path both in larger and as well as in smaller size (which are having same destination) compared to other path packets. The packets originated from the nodes which are having reduced energy level are also scheduled within the same shortest path. Since, the weight assigned as 0.6 for shortest path, packets in that path irrespective of their size, they have been processed earlier. For example, if m-WFQ determines that packet A is in shortest path and packet B is in second queue and packet C is in third queue then packet A is transmitted before packet B or packet C. (Figure 5.2).
Figure 5.2 An example of the m-WFQ algorithm

When each packet is classified and placed into its queue, the scheduler calculates and assigns a finish time for the packet. As the m-WFQ scheduler services its queues, it selects the packet with the earliest (smallest) finish time as the next packet for transmission on the output port. For example, if m-WFQ determines that packet A has a finish time of 30, packet B has a finish time of 70, and packet C has a finish time of 135, then packet A is transmitted before packet B or packet C. In Figure 5.2, we can observe that the appropriate weighting of queues allows a m-WFQ scheduler to transmit two or more consecutive packets from the same queue.

5.2 m-WFQ benefits and limitations

Modified weighted fair queuing has two primary benefits:

- m-WFQ provides protection to each service class by ensuring a minimum level of output port bandwidth independent of the behavior of other service classes.

- When combined with traffic conditioning at the edges of a network, m-WFQ guarantees a weighted fair share of output port bandwidth to each service class with a bounded delay.

However, modified weighted fair queuing comes with following limitations:
• Highly aggregated service classes means that a misbehaving flow within the service class can impact the performance of other flows within the same service class.

• m-WFQ implements a complex algorithm that requires the maintenance of a significant amount of per-service class state and iterative scans of state on each packet arrival and departure.

• Computational complexity impacts the scalability of m-WFQ when attempting to support a large number of service classes which have same RTT and different IP destination address.

• On high-speed interfaces, minimizing delay to the granularity of a single packet transmission may not be worth the computational expense if you consider the insignificant amount of serialization delay introduced by high-speed links and the lower computational requirements of other queue scheduling disciplines.

• Finally, m-WFQ may be better for other queue scheduling disciplines, the delay bounds can still be quite large.

5.2.1 m-WFQ implementations and applications

m-WFQ is deployed at the boundaries of the network to provide a fair distribution of bandwidth among a number of different service classes. In proposed protocol, m-WFQ is configured to support the following range of behaviors:

• m-WFQ is configured to classify packets into a relatively large number of queues using the source/destination UDP/TCP port numbers, and the IP ToS byte.
m-WFQ is configured to allow the system to schedule a limited number of queues that carry aggregated traffic flows. For this configuration option, the system uses QoS policy. The three low-order IP precedence bits in the IP ToS byte to assign packets to queues.

Each of the queues is allocated a different percentage of output port bandwidth based on the weight that the system calculates. The assignment of weight is based on Round Trip Time of the path for each of the service classes. But the total weight, assigned to all flows must be equal to one.

This approach allows the system to allocate different amounts of bandwidth to each queue based on the QoS policy group or to allocate increasing amounts of bandwidth to each queue as the IP precedence increases.

The work is summarized as follows: In our proposed protocol, assumption is made as that there is only one shortest path, to reach their destination. If the same path has been chosen by all packets, congestion will occur. In order to avoid congestion, four of the queue disciplines (FIFO, RED, PQ and WFQ) are used and the same are explained in a detailed manner. These queueing techniques are incorporated in proposed routing protocol and its QoS metrics are analyzed. In the same chapter, modified Weighted Fair Queuing (m-WFQ) is explained with its flow chart. Modified WFQ is implemented for this devised routing protocol. It is a slightly different technique taking into account of bandwidth allocation proportional to packet size so that effectively bandwidth utilization is done. Weights assigned for the path is based on its Round Trip Time along with its destination IP. The path with small Round Trip Time has been assigned with maximum weight. However, the total weights of all paths must be equal to one. Its performances have been compared with other queue disciplines. It has proved that modified WFQ gives better performances due to its speed in processing due to maximum assigned weight for that path and less in delay. So, by using m-WFQ, we can have better QoS. Simulation results are discussed in the next chapter.