Appendix C
Quality of Service Algorithm

This appendix describes the Quality-of-Service (QoS) algorithm used by the HP ProCurve 9304M, 9308M, and 6308M-SX routing switches and the 6208M-SX switch. You can apply QoS priorities to the following:

- Ports
- VLANs
- Static MAC entries
- Layer 4 sessions
- AppleTalk sockets.

Overview of QoS

QoS on the 9304M, 9308M, and 6308M-SX routing switches and the 6208M-SX switch is based on packet-based prioritization. When you assign a QoS priority to a port, you are selecting the queue in which the outbound packets for that port will be placed.

Queues apply to outbound packets, not inbound packets.

You can select one of the following priorities:

- 0 or 1 – These priorities correspond to queue 0, the normal priority queue.
- 2 or 3 – These priorities correspond to queue 1, a higher priority queue.
- 4 or 5 – These priorities correspond to queue 2, a higher priority queue.
- 6 or 7 – These priorities correspond to queue 3, the highest priority queue.

Each port and each module has four queues for outbound traffic.

NOTE: On the 9304M and 9308M, each chassis module also has a queue in which the module places packets for prioritized transmission onto the backplane to other modules. However, you do not need to configure these queues. The prioritization queues you assign to interfaces are used for traffic waiting to be sent to a module for transmission on the interface.

QoS Algorithm

QoS queues are serviced using weighted fair queuing. This scheme services each of the four queues on a given port during a single queue cycle. A queue cycle is one full pass through all four queues. Each queue has a non-configurable weight that determines how many packets the system processes for each queue during a given cycle and how many times the queue is processed during the cycle.

You can calculate the minimum guaranteed percentage of a port's bandwidth a set of queue weights yields by using the following equations:

q3/(q3 + 1) = p3

(1 - p3) * q2/(q2 + 1) = p2

(1 - p3 - p2) * q1/(q1 + 1) = p1

(1 - p3 - p2 -p1) = p0

where:

- p3 is the percentage for queue 3
- p2 is the percentage of what is left over from queue 3 that is allocated to queue 2
- p1 is the percentage of what is left over from queue 2 that is allocated to queue 1
- p0 is the percentage of what is left over from queue 1 that is allocated to queue 0

For the default weights, these calculations yield the following guaranteed minimum percentages:

- p0 = 1.7%
- p1 = 3.3%
- p2 = 15%
- p3 = 80%

Example Queue Cycle

You can compare the queue cycle to the operation of a gasoline pump with multiple dials. Some dials turn only once during a cycle and are triggered by the smaller dials, which turn multiple times during a cycle. For example, the dollars dial moves only slightly after many revolutions of the cents dials. Note that this comparison does not exactly match because the slower dials on the gas pump are the more important dials. (For example, the dollars dial has more effect on your pocket-book than the cents dials). Figure C.1 shows an example of the queues represented as dials.

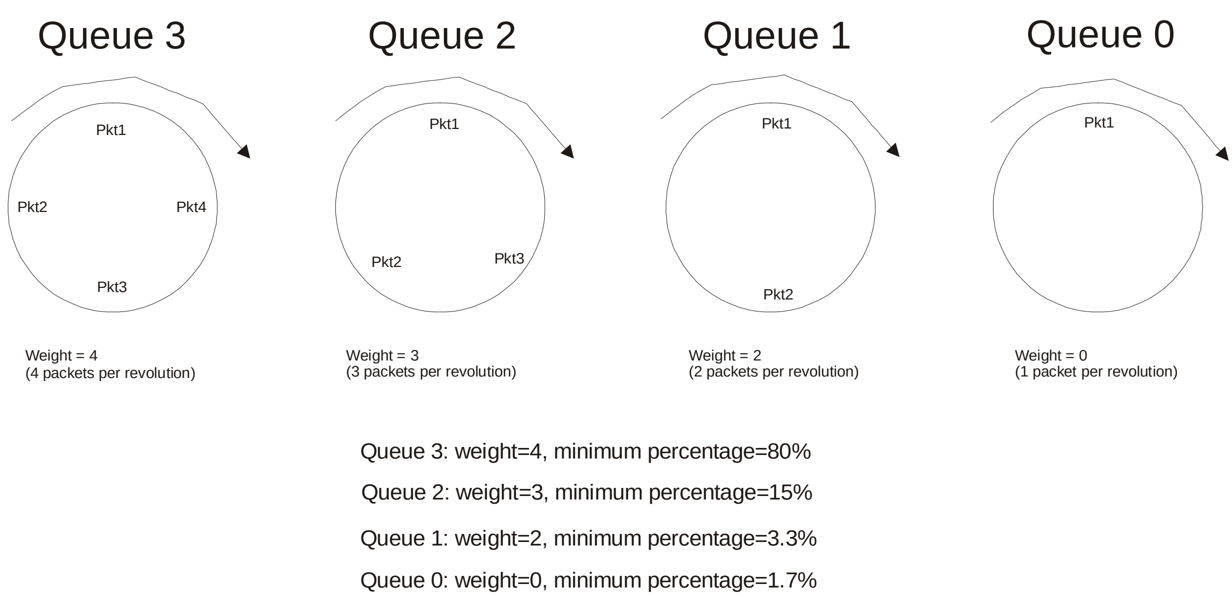


Figure C.1 Queues represented as dials on a meter

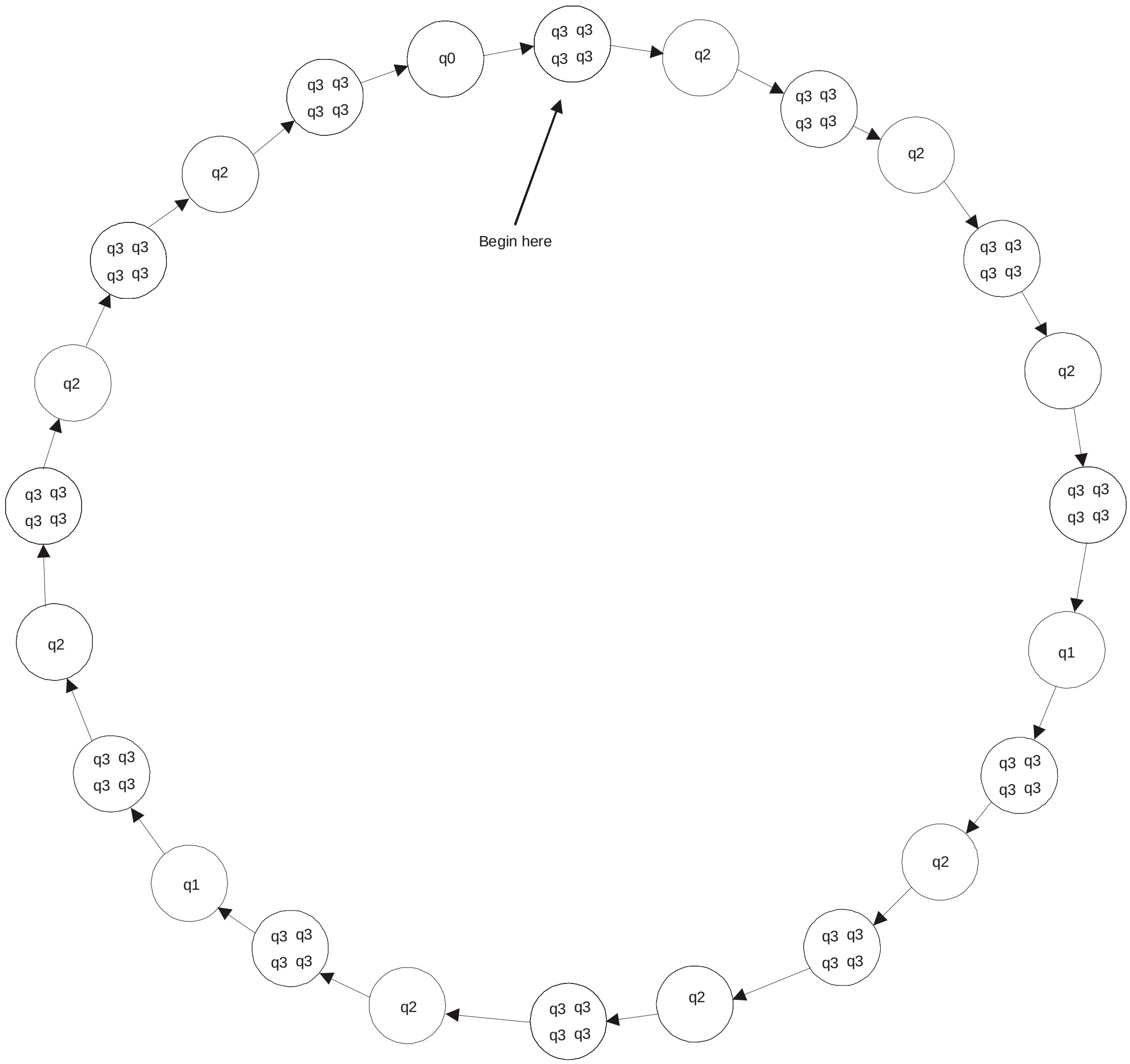
Queue 3 receives the most servicing during a cycle. The device interleaves service for the lower queues in between servicing queue 3. QoS queue 0 receives the least servicing during a QoS cycle. Notice that queue 3 always makes a complete revolution before a lower queue advances. When a queue completes a revolution, the device has serviced the number of packets equal to the weight of the queue. Assuming that queue 3 is using the default weight 4, four packets are serviced in queue 3 before the system advances to the next queue to process a packet.

Table C.1 shows the progress of packets through the queue at each stage in the rotation of the dials shown in Figure C.1.

Table C.1: Queue time-line using default weights

Queue 3 (weight 4)		**Queue 2 (weight 3)**		**Queue 1 (weight 2)**		**Queue 0 (weight 0)**	
Total Revolutions	**Total Packets**	**Total Revolutions**	**Total Packets**	**Total Revolutions**	**Total Packets**	**Total Revolutions**	**Total Packets**
1	4		1				
2	8		2				
3	12	1	3				
4	16				1		
5	20		4				
6	24		5				
7	28	2	6				
8	32			1	2		
9	36		7				
10	40		8				
11	44	3	9				
12	48					1	1

Figure C.2 shows one cycle of the QoS algorithm.



Queue 3: weight=4, minimum percentage=80%

Queue 2: weight=3, minimum percentage=15%

Queue 1: weight=2, minimum percentage=3.3%

Queue 0: weight=0, minimum percentage=1.7%

Figure C.2 One full QoS cycle

As shown in this example, queue 3 is always processed first in a queue cycle. Queue 3's weight determines how many packets from the queue are processed before the algorithm chains to the next queue level.

The weights of the lower queues determine how many times the next higher queue will chain to them before moving to an even lower queue. Because queue 2's weight is 3, the algorithm chains from queue 3 to queue 2 three times before chaining farther down to queue 1. Because queue 1's weight is 2, the algorithm chains to queue 1 twice during the cycle, once after each iteration of servicing queue 2. Queue 0 always has a weight of 0 and always is serviced only once during the cycle, at the very end.