FAIR QUEUEING

Fair Queueing: Fair queueing attempts to provide equitable access to transmission bandwidth. Each user flow has its own logical queue. In an ideal system the transmission bandwidth, say, C bits/second, is divided equally among the queues that have packets to transmit.The contents of each queue can then be viewed as a fluid that is drained continuously. Fair queueing prevents the phenomenon of hogging, which occurs when an information flow receives an unfair share of the bit

Fair queueing is ``fair'' in the following sense. In the ideal fluid flow situation, the transmission bandwidth is divided equally among all nonempty queues. Thus if the total number of flows in the system is n and the transmission capacity is C, then each flow is guaranteed at least Cfln bits/second. In general, the actual transmission rate experienced may be higher because queues will be empty from time to time, so a share larger than Cfln bps is received at those times.

In practice, dividing the transmission capacity exactly equally is not possible. As shown in Figure 1.45 one approach could be to service each nonempty queue one bit at a time in round-robin fashion. However, decomposing the resulting bit stream into the component packets would require the introduction of framing information and extensive processing at the demultiplexer. In the case of ATM, fair queueing can be approximated in a relatively simple way. Because in ATM all packets are the same length, the multiplexer need only service the nonempty queues one packet at a time in round-robin fashion. User flows are then guar- anteed equal access to the transmission bandwidth.

Figure1.46 illustrates the differences between ideal or ``fluid flow'' and packet-by-packet fair queueing. The figure assumes that queue 1 and queue 2 each has a single L-bit packet to transmit at t = 0 and that no subsequent

packets arrive. Assuming a capacity of C = L bits/second fl 1 packet/second,
the fluid-flow system transmits each packet at a rate of 1/2 and therefore.

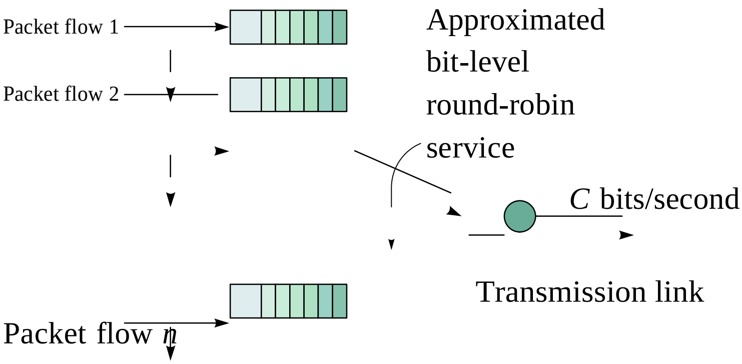

FIGURE 1.45 Fair queueing

completes the transmission of both packets exactly at time t = 2 seconds. The bit-by-bit system (not shown in the figure) would begin by transmitting one bit from queue 1, followed by one bit from queue 2, and so on. On the other hand, the packet-by-packet fair-queueing system transmits the packet from queue 1 first and then transmits the packet from queue 2, so the packet completion times are 1 and 2 seconds. In this case the first packet is 1 second too early relative to the completion time in the fluid system.

Approximating fluid-flow fair queueing is not as straightforward when pack- ets have variable lengths. If the different user queues are serviced one packet at a time in round-robin fashion, we do not necessarily obtain a fair allocation of transmission bandwidth. For example, if the packets of one flow are twice the size of packets in another flow, then in the long run the first flow will obtain twice the bandwidth of the second flow. A better approach is to transmit packets from the user queues so that the packet completion times approximate those of a fluid- flow fair queueing system. Each time a packet arrives at a user queue, the completion time of the packet is derived from a fluid-flow fair-queueing system. This number is used as a finish tag for the packet. Each time the transmission of a packet is completed, the next packet to be transmitted is the one with the smallest finish tag among all of the user queues. We refer to this system as a packet-by-packet fair-queueing.

Assume that there are n flows, each with its own queue. Suppose for now that each queue is served one bit at a time. Let a round consist of a cycle in which all n queues are offered service as shown in Figure 7.47. The actual duration of a given round is the actual number of queues n_{active}…t† that have information to transmit.
When the number of active queues is large, the duration of a round is large; when the number of active queues is small, the rounds are short in duration.

Now suppose that the queues are served as in a fluid-flow system. Also suppose that the system is started at t ˆ 0. Let R…t† be the number of the rounds at time t, that is, the number of cycles of service to all n queues. However, we let R…t† be a continuous function that increases at a rate that is inversely proportional to the number of active queues; that is:

dR…t†fldt = Cfln_{active}…t†

where C is the transmission capacity. Note that R…t† is a piecewise linear function that changes in slope each time the number of active queues changes. Each time R…t† reaches a new integer value marks an instant at which all the queues have been given an equal number of opportunities to transmit a bit.

k Let us see how we can calculate the finish tags to approximate fluid-flow fair queueing. Suppose the kth packet from flow i arrives at an empty queue at time t^i and suppose that the packet has length P…i; k†. This packet will complete its transmission when P…i; k† rounds have elapsed, one round for each bit in the packet. Therefore, the packet completion time will be the value of time t when the R…t† reaches the value:

F…i; k† ˆ R…$_k$t^i † +P…i; k†

We will use F…i; k† as the finish tag of the packet. On the other hand, if the kth packet from the ith flow arrives at a nonempty queue, then the packet will have a finish tag F…i; k† equal to the finish tag of the previous packet in its queue F…i; k 1† plus its own packet length P…i; k†; that is:

F…i; k† ˆ F…i; k 1† + P…i; k†

The two preceding equations can be combined into the following compact equation:

F…i; k† = maxfF…i; k 1†;$_k$R…t^i †g + P…i; k† for fair queueing:

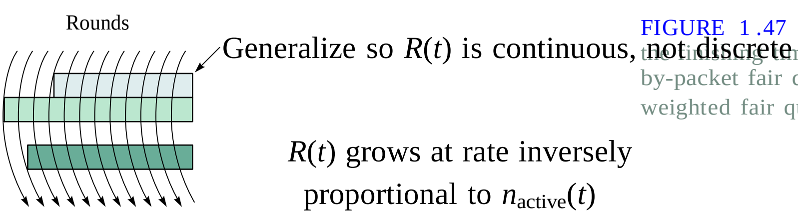

FIGURE 1 .47 Computing the finishing time in packet-by-packet fair queueing and weighted fair queueing

We reiterate: The actual packet completion time for the kth packet in flow I in a fluid-flow fair-queueing system is the time t when R…t† reaches the value F…i; k†. The relation between the actual completion time and the finish tag is not straightforward because the time required to transmit each bit varies according to the number of active queues.

As an example, suppose that at time t 0 queue 1 has one packet of length one unit and queue 2 has one packet of length two units. A fluid-flow system services each queue at rate 1/2 as long as both queues remain nonempty. As shown in Figure 1.48, queue 1 empties at time t = 2. Thereafter queue 2 is served at rate 1 until it empties at time t = 3. In the packet-by-packet fair-queueing system, the finish tag of the packet of queue 1 is F…1; 1† = R…0† + 1 = 1. The finish tag of the packet from queue 2 is F…2; 1† = R…0† + 2 = 2. Since the finish tag of the packet of queue 1 is smaller than the finish tag of queue 2, the system will service queue 1 first. Thus the packet of queue 1 completes its transmissions at time t = 1 and the packet of queue 2 completes its transmissions at t = 3.