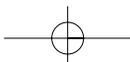
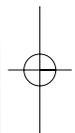
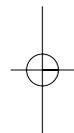


CHAPTER

# 7

## Multimedia Networking





## Most Important Ideas and Concepts from Chapter 7

- ◆ **Differences (and similarities) among the requirements of streaming stored multimedia, streaming live multimedia, and real-time interactive multimedia.** In Section 7.1, we identified three classes of multimedia applications: *streaming stored multimedia*, *streaming live multimedia*, and *real-time interactive multimedia*. (We also identified but ignored, a fourth case in which a stored multimedia file is downloaded in its entirety and then played out, as this case is simply a file transfer application.) In all three cases, multimedia data has both content (for example, the bytes that make up an audio sample or a video frame) and timing attributes. The timing attribute of a video frame might be its temporal location during a particular 1/30 of a second interval of time in the video. Similarly, a packet audio stream might consist of chunks of audio data gathered every 20 msec; the timing attribute of an audio chunk might then be its temporal location within a sequential stream of audio chunks.

Stored applications have the flexibility to transmit data as fast as the network path will allow, since all of the multimedia is stored and always available for transmission. Live applications do not have this flexibility. Interactive human-to-human communication (for example, a teleconference or an audio call) requires low end-end latencies, typically less than 400 msec in order for such interaction communication to feel “natural” for the participants.

- ◆ **Playback delay for jitter removal.** When multimedia data is transferred over a network to a receiver for playback, the receiver must playback the data according to the data’s original timing attributes (see “Differences (and similarities) among the requirements of streaming stored multimedia, streaming live multimedia, and real-time interactive multimedia” above). For example, audio and video data might need to be played out periodically at the receiver, for example, at a rate of one video frame every 1/30 sec, or one audio sample every 20 msec. Once playback begins, the remaining pieces of data each have a playback time that depends on its timing attribute. Data not received before their playback time are considered lost. Another challenge in networked multimedia is that network transmission results in variable delays—jitter—in the received data. For example, data transmitted periodically typically will not arrive periodically. This network-induced jitter must not be apparent in the multimedia playback at the receiver.

One technique to decrease the amount of late-arriving data *and* to accommodate jitter is to delay the beginning of playback, essentially pushing the playback deadlines further into the future. In this case, pieces of arriving data are placed in a playback buffer. After some initial playback delay, the playback process begins and pieces of data are removed from the buffer as dictated by their original timing attributes. The playback buffer not only decreases late-arrival loss, but also masks the jitter. For example, if there are ten packets of data in the playback buffer, it is irrelevant whether those ten packets arrived smoothly over time (with no jitter) or arrived with wildly different delays (high jitter).

- ◆ **Forward error correction (FEC).** In Chapter 3, we studied a number of reliable data transfer protocols that retransmitted lost or damaged packets. An alternative approach toward achieving reliability is to use *forward error correction* techniques. With FEC, enough redundant information is added to the original data so that even if some of the transmitted data (original data plus redundant data) is lost, the receiver can still recover the original data. The simple two-dimensional parity technique that we studied in Chapter 5 for detection and correction of single bit errors is a simple example of FEC. FEC techniques can be particularly valuable when an application cannot wait for a round-trip time to recover lost data via a timeout-and-transmit mechanism.
- ◆ **The Real-Time Transport Protocol (RTP).** RTP is an Internet-standard protocol for the transport of real-time data such as multimedia. It can be used for streaming stored multimedia, streaming live multimedia, and real-time interactive multimedia. RTP does not itself provide for resource reservation, call admission, or quality of service control; these tasks are left to RSVP and other protocols (see “Intserv, Diffserv, and RSVP” below). Instead, RTP provides information (carried both in RTP packet headers, as well as via a separate control protocol known as RTCP) to help the senders and receivers of RTP data perform tasks such as timing reconstruction (see “Playout delay for jitter removal” above), loss detection, content identification, and synchronization among multiple multimedia streams.
- ◆ **The Session Initiation Protocol (SIP).** In telephone networks, so-called signaling protocols have been used for decades to control the manner in which telephone calls are initiated, end-points (for example, the phone associated with an individual subscriber number or the service point for an 800 call) are located, endpoints are contacted, and the circuit through the network connecting the endpoint is set up. SIP is an Internet-standards-track signaling protocol for Internet telephony, teleconferencing, instant messaging, and more. Key elements of the SIP architecture include SIP proxies (which help locate remote endpoints and direct calls to these endpoints), and SIP registrars (which keep track of the locations of registered users). As a more recently-developed protocol, SIP’s design reflects many of the best aspects of earlier protocols, such as HTTP, DNS, and Mobile IP.
- ◆ **High-quality multimedia applications are possible over today’s best-effort network.** During the 1990s, a considerable amount of networking research was devoted toward developing a new network architecture (see “Beyond best effort: packet classification, isolation among traffic flows, and resource reservation” below) that would provide quality of service (QoS) guarantees to multimedia applications. However, the astounding success of multimedia applications such as Skype demonstrates that it *is* possible to run multimedia applications over today’s best-effort public Internet, a network architecture that provides no explicit QoS support. Certainly, as long as resources (for example, bandwidth) are plentiful, multimedia applications can indeed operate effectively over today’s Internet. Application-layer techniques such as adaptive playout buffering, FEC, loss masking, and adaptive coding rates that

match the coding rate to the available bandwidth can improve application-layer performance when the network becomes congested. However, these techniques can compensate for scarce bandwidth only up to a certain point; beyond that, the quality of the multimedia applications will inevitably degrade as the network becomes more congested. So, the question remains—are new network mechanisms and new network architectures required to support multimedia applications? In the end, the answer to this question is likely to be determined more by economics than by technology. If bandwidth remains relatively plentiful and multimedia users are willing to put up with the (hopefully occasional) poor performance when the network is congested, then multimedia over a best-effort Internet may well be the direction in which future networked multimedia activity grows.

- ◆ **Beyond best effort: packet classification, isolation among traffic flows, and resource reservation.** An alternative to continuing to run multimedia applications over today's best-effort Internet (see “High quality multimedia applications are possible over today's best-effort network” above) is to develop a new network architecture that provides explicit QoS support for multimedia applications. In such a network, once a multimedia call is admitted to the network, it receives a guarantee that it will receive a given quality of service (for example, a bounded end-end delay and packet loss rate) throughout the duration of the call. This service model is similar to that of the telephone network—either a call is admitted to the network with a guaranteed QoS or the call is rejected (that is, the user receives a “busy signal” from the network) and the user must try the call again, when the network is hopefully less congested. In Section 7.6, we identified several key architectural components of a future QoS-enabled network, including packet classification, isolation among traffic flows, and resource reservation. Indeed, these concepts are already embodied in a number of Internet RFCs and protocols, including Intserv, Diffserv, and RSVP, as discussed in Sections 7.8 and 7.9, and “Intserv, Diffserv, and RSVP” below.
- ◆ **Scheduling disciplines: FIFO, Round Robin, Priority, and WFQ.** Buried deep within every Internet router and host, in the guts of the link layer, is a very important construct—the queue (buffer) of frames waiting to be forwarded across the link to the device at the other end of the link. The manner in which queued packets are selected for transmission across this link—the link scheduling discipline—has a tremendous impact on application performance. We studied four packet scheduling disciplines in this chapter: FIFO (in which packets are transmitted in their order of arrival), priority service (in which packets are divided into classes, with packets from a higher priority class being transmitted before queued packets from a lower priority class), Round Robin (where packets are again divided into classes, with each class receiving a turn to transmit a packet from that class), and Weighted Fair Queuing (a generalization of Round Robin, with different classes of traffic being given a different number of turns to transmit a packet).

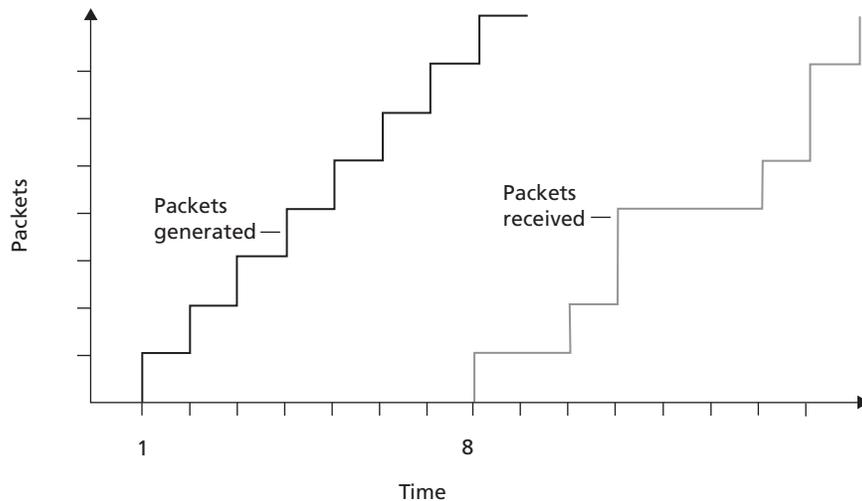
- ◆ **Policing: the leaky bucket mechanism.** In today's best-effort Internet, there are no constraints (other than the physical link speed) on how fast a user (say using the UDP transport protocol) can send packets into the network. For example, if  $N-1$  users are each sending packets to their first hop router at rate  $R$ , an  $N$ th user is free to send packets at rate  $5R$ , or indeed any rate. A leaky bucket mechanism limits both the long term rate at which packets can be sent into the network (given by the leaky bucket's token rate,  $r$ , shown in Figure 7.29 on page 626 of the textbook) and the so-called burstiness of packet transmission (given by the size of the token bucket,  $b$ , in Figure 7.29, which limits the maximum number of packets that can be sent into the network in a short period of time to a maximum of  $b$  packets). A policing mechanism such as the leaky bucket is important for providing QoS guarantees because it limits the amount of traffic that an individual user can send into the network, thereby providing a degree of isolation among users.
- ◆ **Intserv, Diffserv, and RSVP.** The integrated service (Intserv) and differentiated services (Diffserv) architectures are the two network architectures developed within the Internet community to provide QoS guarantees to network applications. Intserv provides the framework for providing hard guarantees (for example, a maximum guaranteed end-end delay) to a session via resource reservation and call admission/blocking. The call admission decision is based on the network's ability to meet the session's requested QoS without violating QoS guarantees made to existing sessions that have already been admitted to the network. Diffserv provides performance guarantees among classes of traffic, rather than to individual sessions. Both Intserv and Diffserv need a signaling protocol to convey information about the traffic demands and performance requirements of individual sessions (in the case of Intserv) or classes of traffic (in the case of Diffserv). This is one of the roles of the Resource Reservation Protocol (RSVP). We would be remiss if we did not mention that Asynchronous Transfer Mode (ATM) networks were also designed to provide QoS guarantees. For example, we saw in Table 4.1 on page 306 of the textbook, that ATM provides a class of service with even stronger guarantees than the Internet Intserv model—ATM's constant bit rate (CBR) service class not only provides a bandwidth guarantee, but also promises to maintain the inter-packet timing of packets flowing through a CBR connection!



## Review Questions

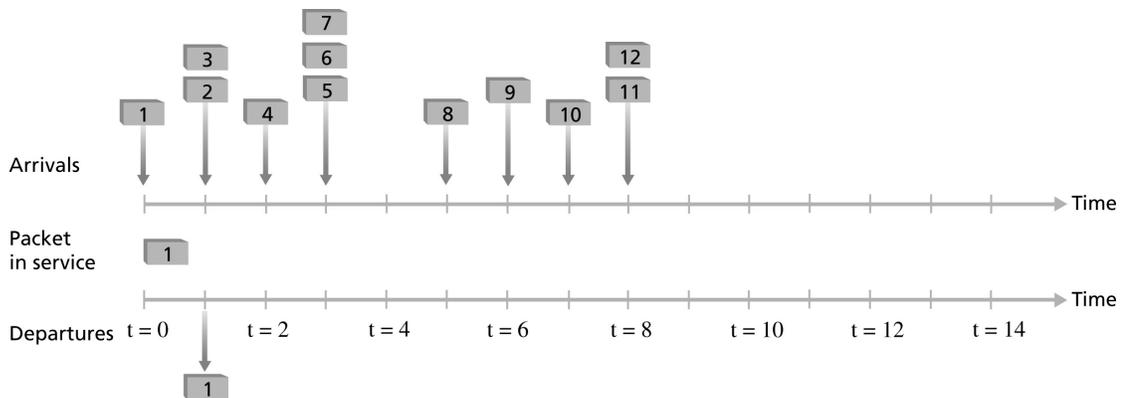
This section provides additional study questions. Answers to each question are provided in the next section.

1. **End-end delay versus delay jitter.** What is the difference between end-end delay and delay jitter? Which of these (delay or delay jitter) is ameliorated with the use of a playout buffer? Suppose that packet audio is transmitted periodically. If the end-end delay is very large, but the jitter is zero, would a large or small playout buffer be needed?
2. **Packet audio playout.** Consider the figure below (which is similar to Figure 7.6 on page 588 of the textbook). A sender begins sending packetized audio periodically at  $t = 1$ . The first packet arrives at the receiver at  $t = 8$ .



- a. What are the delays (from sender to receiver, ignoring any playout delays) of the second, third, fourth, and fifth packets sent? Note that each vertical and horizontal line segment in the figure has a length of 1, 2, or 3 time units.
- b. If audio playout begins as soon as the first packet arrives at the receiver at  $t = 8$ , which of the first 8 packets sent will *not* arrive in time for playout?
- c. If audio playout begins at  $t = 9$ , which of the first 8 packets sent will *not* arrive in time for playout?
- d. What is the minimum playout delay at the receiver that results in all of the first 8 packets arriving in time for playout?

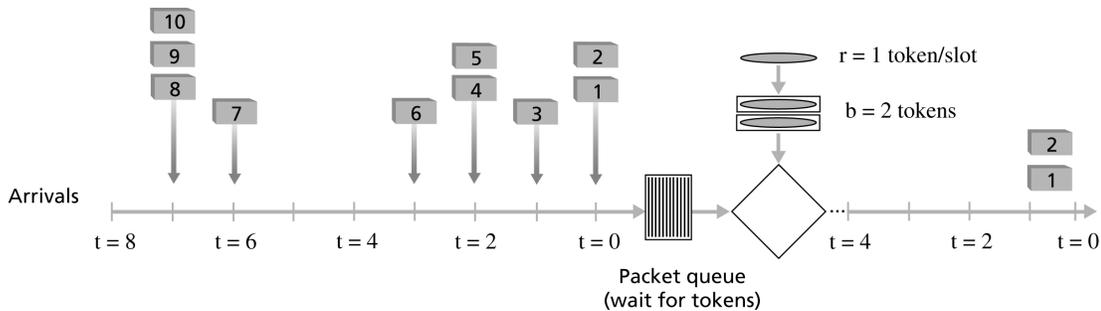
3. **Estimating delay and delay deviation.** Consider the figure from Review Question 2 showing packet audio transmission and reception times.
  - a. Compute the estimated delay for packets 2 through 8, using the formula for  $d_i$  on page 589 of the textbook. Use a value of  $u = 0.1$ .
  - b. Compute the estimated deviation of the delay from the estimated average for packets 2 through 8, using the formula for  $v_i$  on page 589 of the textbook. Use a value of  $u = 0.1$ .
4. **ACKs versus FEC.** Consider a sender and receiver that are separated by a long-distance, high-bandwidth link that can occasionally lose or damage packets. The link is running at low utilization, and it is important to keep the application-to-application delivery delay as small as possible. Would you recommend using an acknowledgement-based mechanism or an FEC-based mechanism for reliable data transfer? Why?
5. **RTP.** Why do you think RTP has both a timestamp field and a sequence number field? For example, in order to recover from loss, if the receiver knows that the packetization interval is 20 msec, and receives packets with timestamps of 0, 20, 40, 60, and 100 msec, isn't that sufficient to know that the sample taken at 80 msec has been lost?
6. **Similarities and differences between SIP and Mobile IP.** Comment on the similarities and differences between how SIP and Mobile IP support communication between two (potentially mobile) devices.
7. **Packet scheduling.** Consider the following figure, which is similar to Figures 7.24–7.27 on pages 622–624 of the textbook.



- a. Assuming FIFO service, indicate the time at which packets 2 through 12 each leave the queue. For each packet, what is the delay between its arrival and the beginning of the slot in which it is transmitted? What is the average of this delay over all 12 packets?

- b. Now assume a Priority Service, and assume that odd-numbered packets are high priority, and even-numbered packets are low priority. Indicate the time at which packets 2 through 12 each leave the queue. For each packet, what is the delay between its arrival and the beginning of the slot in which it is transmitted? What is the average of this delay over all 12 packets?
      - c. Now assume Round Robin service. Assume that packets 1, 2, 3, 6, 11, and 12 are from class 1, and packets 4, 5, 7, 8, 9, and 10 are from class 2. Indicate the time at which packets 2 through 12 each leave the queue. For each packet, what is the delay between its arrival and the beginning of the time slot in which it is transmitted? What is the average delay over all 12 packets?
      - d. Now assume Weighted Fair Queuing (WFQ) service. Assume that odd-numbered packets are from class 1, and even-numbered packets are from class 2. Class 1 has a WFQ weight of 2, while class 2 has a WFQ weight of 1. Note that it may not be possible to achieve an idealized WFQ schedule as described in the textbook, so indicate why you have chosen the particular packet to go into service at each time slot. For each packet, what is the delay between its arrival and the beginning of the time slot in which it is transmitted? What is the average delay over all 12 packets?
      - e. What do you notice about the average delay in all four cases (FCFS, RR, Priority, and WFQ)?
8. **Packet scheduling (more).** Consider again the figure from Review Question 7.
  - a. Assume a priority service, with packets 1, 4, 5, 6, and 11 being high priority packets. The remaining packets are low priority. Indicate the slots in which packets 2 through 12 each leave the queue.
  - b. Now suppose that round robin-service is used, with packets 1, 4, 5, 6, and 11 belonging to one class of traffic, and the remaining packets belonging to the second class of traffic. Indicate the slots in which packets 2 through 12 each leave the queue.
  - c. Now suppose that WFQ service is used, with packets 1, 4, 5, 6, and 11 belonging to one class of traffic, and the remaining packets belonging to the second class of traffic. Class 1 has a WFQ weight of 1, while class 2 has a WFQ weight of 2 (note that these weights are different from those in Review Question 7). Indicate the slots in which packets 2 through 12 each leave the queue. See also the caveat in the question 7d above regarding WFQ service.
9. **Leaky bucket.** Consider the following figure, which shows a leaky bucket policer being fed by a stream of packets. The token buffer can hold at most two tokens, and is initially full at  $t = 0$ . New tokens arrive at a rate of 1 token per slot. The output link speed is such that if two packets obtain tokens at the beginning of a time slot, they can both pass to the output link in the same slot. The timing details of the system are as follows:

1. Packets (if any) arrive at the beginning of the slot. Thus, in the example below, packets 1 and 2 arrive in slot 0. If there are already packets in the queue, then the arriving packets join the end of the queue. Packets proceed toward the front of the queue in a FIFO manner.
2. If, after the arrivals (if any) have been added to the queue, there are any queued packets, one or two of those packets (depending on the number of available tokens) will each remove a token from the token buffer and pass to the output link during that slot. Thus, as shown in the example below, packets 0 and 1 each remove a token from the buffer (since there are initially two tokens) and pass to the output link during slot 0.
3. A new token is added to the token buffer if it is not full, since the token generation rate is  $r = 1$  token/slot.
4. Time then advances to the next time slot, and these steps repeat.



- a. For each time slot, identify the packets that are in the queue and the number of tokens in the bucket, immediately after the arrivals have been processed (see step 1 above) but before any of the packets have passed through the queue and removed a token. Thus, for the  $t = 0$  time slot in the example above, packets 1 and 2 are in the queue, and there are two tokens in the buffer.
  - b. For each time slot, indicate which packets appear on the output after the token(s) have been removed from the queue. Thus, for the  $t = 0$  time slot in the example above, packets 1 and 2 appear on the output link from the leaky bucket during slot 0.
10. **Leaky bucket (more).** Repeat Review Question 9, but assume that  $r = 2$ . Assume again that the bucket is initially full.
  11. **Leaky bucket (even more).** Consider Review Question 10 and suppose that  $r = 3$ , and that  $b = 2$  as before. Will your answer to the question above change?



## Answers to Review Questions

1. The end-end delay of a packet is the total accumulated delay from when the packet is sent by the sender to when it is received at the final destination, including propagation, queuing, and forwarding delays at the intervening routers on the end-end path. Delay jitter refers to the difference in end-end delay between two adjacent packets. The playout buffer is used to remove the jitter from the received audio packets, so that packets can be played out according to their original timing attributes. If there was no delay jitter, periodically-transmitted packets would arrive periodically with their inter-packet timing preserved, and hence there would be no need for a playout buffer (regardless of the end-end delay).
2.
  - a. The delay of packet 2 is 8 slots. The delay of packet 3 is 8 slots. The delay of packet 4 is 7 slots. The delay of packet 5 is 9 slots.
  - b. Packets 2, 3, 5, 6, 7, and 8 will not be received in time for their playout if playout begins at  $t = 8$ .
  - c. Packets 5 and 6 will not be received in time for their playout if playout begins at  $t = 9$ .
  - d. No packets will arrive after their playout time if playout begins at  $t = 10$ .
3. The answers to parts a and b are in the table below.

Packet Number	$r_i - t_i$	$d_i$	$v_i$
1	7	7	0
2	8	7.10	0.09
3	8	7.19	0.162
4	7	7.17	0.163
5	9	7.35	0.311
6	9	7.52	0.428
7	8	7.57	0.429
8	8	7.61	0.425

4. An FEC-based mechanism would be a good idea here, since the utilization is low, FEC bits can be used to correct an error (and even a lost packet) at the receiver without having to wait a round-trip time to timeout and re-transmit the data in error.
5. If there is no multimedia data generated during the interval beginning at 80 msec (for example, the audio source is in a silent period), then (depending on the audio application) it is possible that no packet would be sent. With sequence

numbers, the fact that there was no audio sample at 80 msec would be clear, since the audio sample at 100 msec would have a sequence number that is only one larger than the sample taken at 60 msec (indicating that there was no sample at 80 msec). Without the sequence number, the receiver would not know whether there was simply no data generated during the 80 msec interval (for example, silence) or whether data was generated but the packet containing the data was lost.

6. Similarities:

- The idea of a registrar—a place that one can go to get information about the location of a user—is similar to the home agent in Mobile IP. However, a user may be registered with multiple SIP registrars (as shown in Figure 7.14 on page 608 of the textbook), and so a SIP proxy may need to contact several registrars before locating a user. In this respect, SIP location resolution is somewhat closer to DNS name resolution than Mobile IP user location. In mobile IP there is only a single home agent.
- In both Mobile IP and in SIP, a mobile user will register with its home agent or SP registrar so that the agent/registrar knows its location.

Differences:

- In Mobile IP, all communication to the mobile host goes through the home agent. In SIP, once the SIP clients have each other's address, communication in both directions is directly between the clients, without additionally involving the SIP proxy or registrar.

7. a.

<b>FCFS Service</b>			
Packet	Arrival slot	Transmission slot	Delay
1	0	0	0
2	1	1	0
3	1	2	1
4	2	3	1
5	3	4	1
6	3	5	2
7	3	6	3
8	5	7	2
9	6	8	2
10	7	9	2
11	8	10	2
12	8	11	3
Average delay:			1.583

**120** STUDY COMPANION FOR COMPUTER NETWORKING, THIRD EDITION

b.

<b>Priority Service</b>				
Packet	Class	Arrival slot	Transmission slot	Delay
1	H	0	0	0
2	L	1	2	1
3	H	1	1	0
4	L	2	5	3
5	H	3	3	0
6	L	3	7	4
7	H	3	4	1
8	L	5	9	4
9	H	6	6	0
10	L	7	10	3
11	H	8	8	0
12	L	8	11	3
Average delay:				1.583

c.

<b>Round Robin Service</b>				
Packet	Class	Arrival slot	Transmission slot	Delay
1	C1	0	0	0
2	C1	1	1	0
3	C1	1	3	2
4	C2	2	2	0
5	C2	3	4	1
6	C1	3	5	2
7	C2	3	6	3
8	C2	5	7	2
9	C2	6	9	3
10	C2	7	11	4
11	C1	8	8	0
12	C1	8	10	2
Average delay:				1.583

- d. In this solution, we implement WFQ by dividing time into sets of three arrivals slots (0–2, 3–5, 6–8, 9–11). For each set of arrival slots, we consider the packets that are available for transmission during those three slots. We try to transmit two packets from class 1 and one packet from class 2 during these three slots. By convention, class 1 packets go before class 2 packets within this group of three, when possible.

In slots 0–2, it is possible to transmit two class 1 packets (packets 1, 3) and one class 2 packet (packet 2). During slots 3–5, it is possible to send two class 1 packets (packets 5, 7) and one class 2 packet (packet 4). During slots 6–8, it is possible to send only one class 1 packet (9), so we send two class 2 packets (6, 8) after sending packet 9 (assuming that once we begin sending class 2 packets during a set of three slots, we only send class 2 packets from then on). In slots 9–11, there is only one more class 1 packet to send (11), so we send it, followed by the last two class 2 packets (10, 12).

WFQ Service				
Packet	Class	Arrival slot	Transmission slot	Delay
1	C1	0	0	0
2	C2	1	2	1
3	C1	1	1	0
4	C2	2	5	3
5	C1	3	3	0
6	C2	3	7	4
7	C1	3	4	1
8	C2	5	8	3
9	C1	6	6	0
10	C2	7	10	3
11	C1	8	9	1
12	C2	8	11	3
Average delay:				1.583

- e. The average delay of a packet is the same in all cases! This illustrates an important conservation law of queuing systems: as long as the queue is kept busy whenever there is a packet queued, the average packet delay will be the same, regardless of the scheduling discipline! Of course, spe-

**122** STUDY COMPANION FOR COMPUTER NETWORKING, THIRD EDITION

cific packets will suffer higher or lower delays under different scheduling disciplines, but the average will always be the same.

8. The answers to parts a–c are in the table below.

Packet	Class	Arrival slot	Transmission slot under Priority	Transmission slot under RR	Transmission slot under WFQ
1	C1	0	0	0	0
2	C2	1	1	1	1
3	C2	1	5	3	2
4	C1	2	2	2	3
5	C1	3	3	4	6
6	C1	3	4	6	9
7	C2	3	6	5	4
8	C2	5	7	7	5
9	C2	6	9	9	7
10	C2	7	10	10	8
11	C1	8	8	8	10
12	C2	8	11	11	11

9. The answers to parts a and b can be found in the table below.

Slot	Packets in queue	Number of tokens before output	Packets on output
0	1, 2	2	1, 2
1	3	1	3
2	4, 5	1	4
3	5, 6	1	5
4	6	1	6
5	empty	1	
6	7	2	7
7	8, 9, 10	2	8, 9
8	10	1	10

10.

Slot	Packets in queue	Number of tokens before output	Packets on output
0	1, 2	2	1, 2
1	3	2	3
2	4, 5	2	4, 5
3	6	2	6
4	empty	2	
5	empty	2	
6	7	2	7
7	8, 9, 10	2	8, 9
8	10	3	10

11. No. Since the bucket can only hold two tokens, one of the arriving three tokens will overflow and be lost. Thus, at most, two tokens can actually enter the bucket per slot, which is the same condition as in Review Question 10.

