In Section 7.1, we described the intrinsic characteristics of video and voice, and then classified multimedia applications into three categories: (i) streaming stored audio/video, (ii) conversational voice/video-over-IP, and (iii) streaming live audio/video.

In Section 7.2, we studied streaming stored video in some depth. For streaming video applications, prerecorded videos are placed on servers, and users send requests to these servers to view the videos on demand. We saw that streaming video systems can be classified into three categories: UDP streaming, HTTP streaming, and adaptive HTTP streaming. Although all three types of systems are used in practice, the majority of today's systems employ HTTP streaming and adaptive HTTP streaming. We observed that the most important performance measure for streaming video is average throughput. In Section 7.2 we also investigated CDNs, which help distribute massive amounts of video data to users around the world. We also surveyed the technology behind three major Internet video-streaming companies: Netflix, YouTube, and Kankan.

In Section 7.3, we examined how conversational multimedia applications, such as VoIP, can be designed to run over a best-effort network. For conversational multimedia, timing considerations are important because conversational applications are highly delay-sensitive. On the other hand, conversational multimedia applications are loss-tolerant—occasional loss only causes occasional glitches in audio/video playback, and these losses can often be partially or fully concealed. We saw how a combination of client buffers, packet sequence numbers, and timestamps can greatly alleviate the effects of network-induced jitter. We also surveyed the technology behind Skype, one of the leading voice- and video-over-IP companies. In Section 7.4, we examined two of the most important standardized protocols for VoIP, namely, RTP and SIP.

In Section 7.5, we introduced how several network mechanisms (link-level scheduling disciplines and traffic policing) can be used to provide differentiated service among several classes of traffic.

Homework Problems and Questions

Chapter 7 Review Questions

SECTION 7.1

R1. Reconstruct Table 7.1 for when Victor Video is watching a 4 Mbps video, Facebook Frank is looking at a new 100 Kbyte image every 20 seconds, and Martha Music is listening to 200 kbps audio stream.

R2. There are two types of redundancy in video. Describe them, and discuss how they can be exploited for efficient compression.

R3. Suppose an analog audio signal is sampled 16,000 times per second, and each sample is quantized into one of 1024 levels. What would be the resulting bit rate of the PCM digital audio signal?

R4. Multimedia applications can be classified into three categories. Name and describe each category.

SECTION 7.2

R5. Streaming video systems can be classified into three categories. Name and briefly describe each of these categories.

R6. List three disadvantages of UDP streaming.

R7. With HTTP streaming, are the TCP receive buffer and the client's application buffer the same thing? If not, how do they interact?

R8. Consider the simple model for HTTP streaming. Suppose the server sends bits at a constant rate of 2 Mbps and playback begins when 8 million bits have been received. What is the initial buffering delay t_p?

R9. CDNs typically adopt one of two different server placement philosophies. Name and briefly describe these two philosophies.

R10. Several cluster selection strategies were described in Section 7.2.4. Which of these strategies finds a good cluster with respect to the client's LDNS? Which of these strategies finds a good cluster with respect to the client itself?

R11. Besides network-related considerations such as delay, loss, and bandwidth performance, there are many additional important factors that go into designing a cluster selection strategy. What are they?

SECTION 7.3

R12. What is the difference between end-to-end delay and packet jitter? What are the causes of packet jitter?

R13. Why is a packet that is received after its scheduled playout time considered lost?

R14. Section 7.3 describes two FEC schemes. Briefly summarize them. Both schemes increase the transmission rate of the stream by adding overhead. Does interleaving also increase the transmission rate?

SECTION 7.4

R15. How are different RTP streams in different sessions identified by a receiver? How are different streams from within the same session identified?

R16. What is the role of a SIP registrar? How is the role of an SIP registrar different from that of a home agent in Mobile IP?

SECTION 7.5

R17. In Section 7.5, we discussed non-preemptive priority queuing. What would be preemptive priority queuing? Does preemptive priority queuing make sense for computer networks?

R18. Give an example of a scheduling discipline that is *not* work conserving.

R19. Give an example from queues you experience in your everyday life of FIFO, priority, RR, and WFQ.

Problems

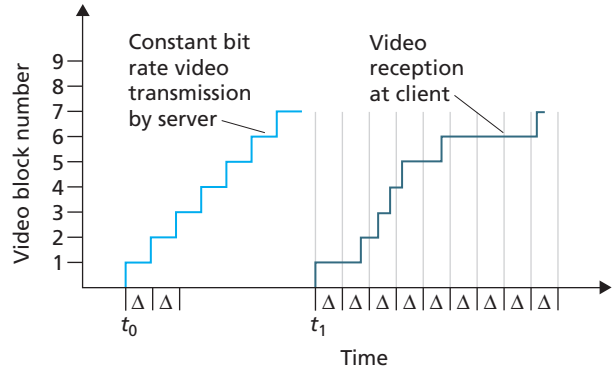
P1. Consider the figure below. Similar to our discussion of Figure 7.1, suppose that video is encoded at a fixed bit rate, and thus each video block contains video frames that are to be played out over the same fixed amount of time, $\triangle$. The server transmits the first video block at t_0, the second block at $t_0 + \triangle$, the third block at $t_0 + 2\triangle$, and so on. Once the client begins playout, each block should be played out $\triangle$ time units after the previous block.

a. Suppose that the client begins playout as soon as the first block arrives at t_1. In the figure below, how many blocks of video (including the first block) will have arrived at the client in time for their playout? Explain how you arrived at your answer.

b. Suppose that the client begins playout now at $t_1 + \triangle$. How many blocks of video (including the first block) will have arrived at the client in time for their playout? Explain how you arrived at your answer.

c. In the same scenario at (b) above, what is the largest number of blocks that is ever stored in the client buffer, awaiting playout? Explain how you arrived at your answer.

d. What is the smallest playout delay at the client, such that every video block has arrived in time for its playout? Explain how you arrived at your answer.



P2. Recall the simple model for HTTP streaming shown in Figure 7.3. Recall that B denotes the size of the client's application buffer, and Q denotes the number of bits that must be buffered before the client application begins playout. Also r denotes the video consumption rate. Assume that the server sends bits at a constant rate x whenever the client buffer is not full.

a. Suppose that $x < r$. As discussed in the text, in this case playout will alternate between periods of continuous playout and periods of freezing. Determine the length of each continuous playout and freezing period as a function of Q, r, and x.

b. Now suppose that $x > r$. At what time $t = t_f$ does the client application buffer become full?

P3. Recall the simple model for HTTP streaming shown in Figure 7.3. Suppose the buffer size is infinite but the server sends bits at variable rate $x(t)$. Specifically, suppose $x(t)$ has the following saw-tooth shape. The rate is initially zero at time $t = 0$ and linearly climbs to H at time $t = T$. It then repeats this pattern again and again, as shown in the figure below.

a. What is the server's average send rate?

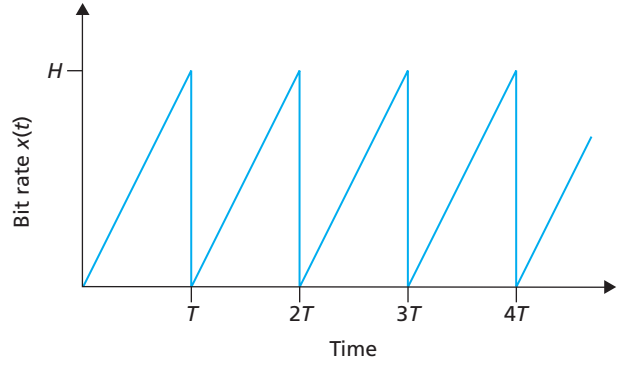
b. Suppose that $Q = 0$, so that the client starts playback as soon as it receives a video frame. What will happen?

c. Now suppose $Q > 0$. Determine as a function of Q, H, and T the time at which playback first begins.

d. Suppose $H > 2r$ and $Q = HT/2$. Prove there will be no freezing after the initial playout delay.

e. Suppose $H > 2r$. Find the smallest value of Q such that there will be no freezing after the initial playback delay.

f. Now suppose that the buffer size B is finite. Suppose $H > 2r$. As a function of Q, B, T, and H, determine the time $t = t_f$ when the client application buffer first becomes full.



P4. Recall the simple model for HTTP streaming shown in Figure 7.3. Suppose the client application buffer is infinite, the server sends at the constant rate x, and the video consumption rate is r with $r < x$. Also suppose playback begins immediately. Suppose that the user terminates the video early at time $t = E$. At the time of termination, the server stops sending bits (if it hasn't already sent all the bits in the video).

   a. Suppose the video is infinitely long. How many bits are wasted (that is, sent but not viewed)?

   b. Suppose the video is T seconds long with $T > E$. How many bits are wasted (that is, sent but not viewed)?

P5. Consider a DASH system for which there are N video versions (at N different rates and qualities) and N audio versions (at N different rates and versions). Suppose we want to allow the player to choose at any time any of the N video versions and any of the N audio versions.

   a. If we create files so that the audio is mixed in with the video, so server sends only one media stream at given time, how many files will the server need to store (each a different URL)?

   b. If the server instead sends the audio and video streams separately and has the client synchronize the streams, how many files will the server need to store?

P6. In the VoIP example in Section 7.3, let h be the total number of header bytes added to each chunk, including UDP and IP header.

   a. Assuming an IP datagram is emitted every 20 msecs, find the transmission rate in bits per second for the datagrams generated by one side of this application.

   b. What is a typical value of h when RTP is used?

P7. Consider the procedure described in Section 7.3 for estimating average delay d_i. Suppose that $u = 0.1$. Let $r_1 - t_1$ be the most recent sample delay, let $r_2 - t_2$ be the next most recent sample delay, and so on.

   a. For a given audio application suppose four packets have arrived at the receiver with sample delays $r_4 - t_4$, $r_3 - t_3$, $r_2 - t_2$, and $r_1 - t_1$. Express the estimate of delay d in terms of the four samples.

   b. Generalize your formula for n sample delays.

   c. For the formula in Part b, let n approach infinity and give the resulting formula. Comment on why this averaging procedure is called an exponential moving average.

P8. Repeat Parts a and b in Question P7 for the estimate of average delay deviation.

P9. For the VoIP example in Section 7.3, we introduced an online procedure (exponential moving average) for estimating delay. In this problem we will

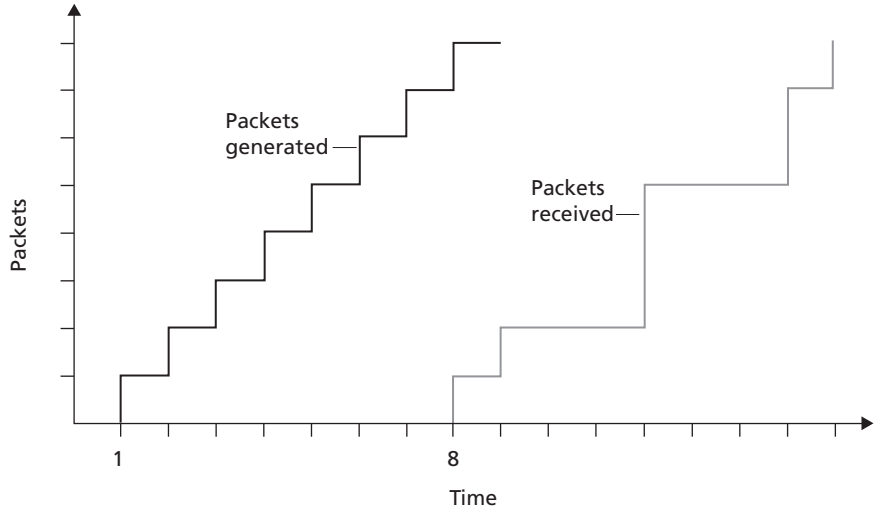
examine an alternative procedure. Let t_i be the timestamp of the ith packet received; let r_i be the time at which the ith packet is received. Let d_n be our estimate of average delay after receiving the nth packet. After the first packet is received, we set the delay estimate equal to $d_1 = r_1 - t_1$.

a. Suppose that we would like $d_n = (r_1 - t_1 + r_2 - t_2 + \ldots + r_n - t_n)/n$ for all n. Give a recursive formula for d_n in terms of d_{n-1}, r_n, and t_n.

b. Describe why for Internet telephony, the delay estimate described in Section 7.3 is more appropriate than the delay estimate outlined in Part a.

P10. Compare the procedure described in Section 7.3 for estimating average delay with the procedure in Section 3.5 for estimating round-trip time. What do the procedures have in common? How are they different?

P11. Consider the figure below (which is similar to Figure 7.7). 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 packets 2 through 8? 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 eight packets sent will *not* arrive in time for playout?

c. If audio playout begins at $t = 9$, which of the first eight 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 eight packets arriving in time for their playout?

P12. Consider again the figure in P11, showing packet audio transmission and reception times.

a. Compute the estimated delay for packets 2 through 8, using the formula for d_i from Section 7.3.2. 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 from Section 7.3.2. Use a value of $u = 0.1$.

P13. Recall the two FEC schemes for VoIP described in Section 7.3. Suppose the first scheme generates a redundant chunk for every four original chunks. Suppose the second scheme uses a low-bit rate encoding whose transmission rate is 25 percent of the transmission rate of the nominal stream.

a. How much additional bandwidth does each scheme require? How much playback delay does each scheme add?

b. How do the two schemes perform if the first packet is lost in every group of five packets? Which scheme will have better audio quality?

c. How do the two schemes perform if the first packet is lost in every group of two packets? Which scheme will have better audio quality?

P14. a. Consider an audio conference call in Skype with $N > 2$ participants. Suppose each participant generates a constant stream of rate r bps. How many bits per second will the call initiator need to send? How many bits per second will each of the other $N - 1$ participants need to send? What is the total send rate, aggregated over all participants?

b. Repeat part (a) for a Skype video conference call using a central server.

c. Repeat part (b), but now for when each peer sends a copy of its video stream to each of the $N - 1$ other peers.

P15. a. Suppose we send into the Internet two IP datagrams, each carrying a different UDP segment. The first datagram has source IP address A1, destination IP address B, source port P1, and destination port T. The second datagram has source IP address A2, destination IP address B, source port P2, and destination port T. Suppose that A1 is different from A2 and that P1 is different from P2. Assuming that both datagrams reach their final destination, will the two UDP datagrams be received by the same socket? Why or why not?

b. Suppose Alice, Bob, and Claire want to have an audio conference call using SIP and RTP. For Alice to send and receive RTP packets to and from Bob and Claire, is only one UDP socket sufficient (in addition to the socket needed for the SIP messages)? If yes, then how does Alice's SIP client distinguish between the RTP packets received from Bob and Claire?

P16. True or false:

a. If stored video is streamed directly from a Web server to a media player, then the application is using TCP as the underlying transport protocol.

b. When using RTP, it is possible for a sender to change encoding in the middle of a session.

c. All applications that use RTP must use port 87.

d. If an RTP session has a separate audio and video stream for each sender, then the audio and video streams use the same SSRC.

e. In differentiated services, while per-hop behavior defines differences in performance among classes, it does not mandate any particular mechanism for achieving these performances.

f. Suppose Alice wants to establish an SIP session with Bob. In her INVITE message she includes the line: m=audio 48753 RTP/AVP 3 (AVP 3 denotes GSM audio). Alice has therefore indicated in this message that she wishes to send GSM audio.

g. Referring to the preceding statement, Alice has indicated in her INVITE message that she will send audio to port 48753.

h. SIP messages are typically sent between SIP entities using a default SIP port number.

i. In order to maintain registration, SIP clients must periodically send REGISTER messages.

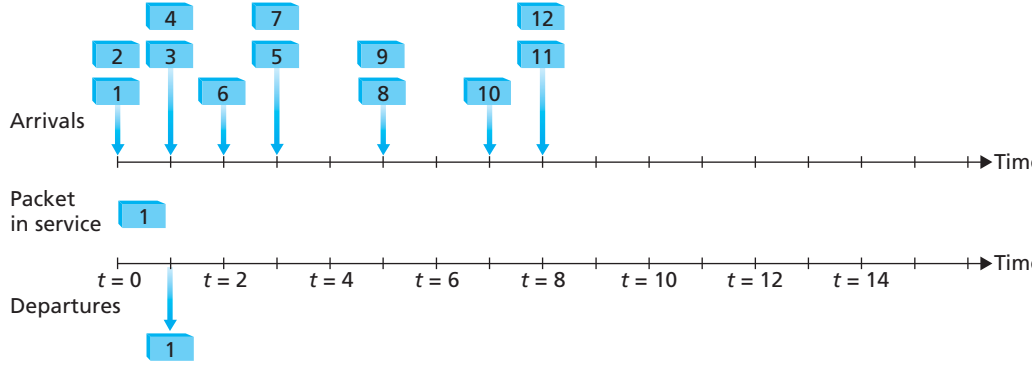
j. SIP mandates that all SIP clients support G.711 audio encoding.

P17. Suppose that the WFQ scheduling policy is applied to a buffer that supports three classes, and suppose the weights are 0.5, 0.25, and 0.25 for the three classes.

a. Suppose that each class has a large number of packets in the buffer. In what sequence might the three classes be served in order to achieve the WFQ weights? (For round robin scheduling, a natural sequence is 123123123 . . .).

b. Suppose that classes 1 and 2 have a large number of packets in the buffer, and there are no class 3 packets in the buffer. In what sequence might the three classes be served in to achieve the WFQ weights?

P18. Consider the figure below. Answer the following questions:



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 its departure? What is the average delay over all 12 packets?

d. Now assume weighted fair queueing (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 text, 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 its departure? What is the average delay over all 12 packets?

e. What do you notice about the average delay in all four cases (FIFO, RR, priority, and WFQ)?

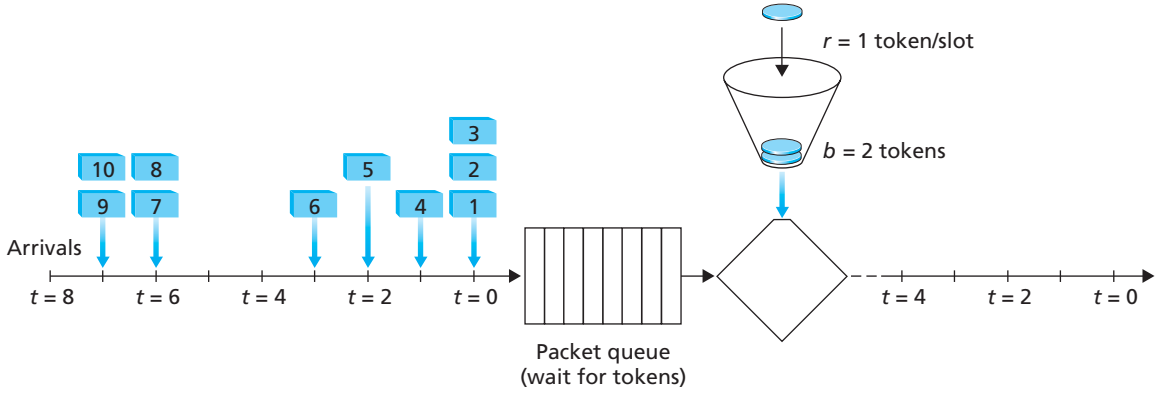
P19. Consider again the figure for P18.

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 than in the previous question). Indicate the slots in which packets 2 through 12 each leave the queue. See also the caveat in the question above regarding WFQ service.

P20. Consider the figure below, 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 one 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 go 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 figure, packets 1, 2, and 3 arrive in slot 0. If there are already packets in the queue, then the arriving packets join the end of the queue. Packets proceed towards the front of the queue in a FIFO manner.
2. After the arrivals have been added to the queue, if 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 go to the output link during that slot. Thus, packets 1 and 2 each remove a token from the buffer (since there are initially two tokens) and go 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.

Answer the following questions:

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 (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, 2 and 3 are in the queue, and there are two tokens in the buffer.