Homework 2 (Solutions)

CS414, Multimedia Systems (Instructor: Klara Nahrstedt)

Posted: April 18, 2008

Due: April 25, 2008 at 11:59pm CST

Important Instructions

This homework assignment should be done individually. Penalties for cheating as described in the grading policy on the course website apply. Solutions should be done with a document preparation system, such as LaTeX or Microsoft Word (figures may be drawn by hand). In the homework solutions, you should show all of your work that you used to arrive at each answer to the problems. If possible, a hard copy of the assignment should be turned into the instructor at the beginning of class on Friday, April 25. Otherwise, a hard copy of the assignment should be slid under the door to the instructor's office (3104 SC) by the specified deadline.

1. Problem – Buffering (20 Points)

- **(5 Points)** Consider the Felini multimedia file system. In this system, data for client requests are retrieved in cycles P (P refers to the common time period of the whole system). If a real time client '*i*' has the consumption rate f_i (in bits per second) and there are *k* bits in the buffer at the start of a cycle, then the admission controller checks at the start of each cycle if $k \geq P \times f_i$. With this admission test, what does the admission controller ensures? Explain.

Solution: With this admission equation we can ensure that at the beginning of each cycle we are at least one frame ahead, if f_i rate means consumption rate of one frame from client '*i*'. It means that if there is congestion on the network, the client will have at least one frame to display at the beginning of the cycle to ensure that no blocking occurs. However, this condition does not prevent overflow of the buffer.

- **(5 Points)** Consider the **Maxbuf** buffering strategy with the maximal buffer size of B_{max}. Let us consider that the multimedia frame size at time t is *M(t)* , and the number of bits received at the receiver side at time t is *C(t).* What are the two buffer states that a receiver needs to check for and try to avoid? What are the exact conditions that the receiver could check for both buffer states?

Solution: The two states that any buffer management scheme must avoid are: (a) **starvation** when the buffer does not have any data to display due to congestion or other problems in the network and/or system. The system blocks in this case since there is no multimedia frame to display. (b) **overflow** when the amount of incoming data exceeds B_{max} and the buffer overflows. In this case the sender/network delivered packets faster than the receiver could display.

The exact conditions are: to avoid starvation $C(t) \geq M(t)$: with this condition we ensure that we will have always at least amount of bits corresponding to frame size to display. To avoid overflow, the condition is $C(t) \leq B_{max} + M(t)$: it means that one frame will be in display frame buffer and the rest of the data received will be in buffer which should not exceed the max buffer size.

- **(5 Points)** Explain the two buffer management schemes used for multimedia protocol processing, the *Offset buffer management* and *Scatter/Gather buffer management*, and compare them with respect to CPU bandwidth and memory bandwidth requirements and reason why.

Solution*:* *Offset Management buffer management* takes into account at the highest level (application level) what are the sizes of protocol control headers of individual protocol data units (PDUs) in the layers below. It means that if the application has under it three other protocol layers (e.g., RTP, UDP, IP) below it, it needs to assign buffer that encompasses (a) the application header, (b) the header of RTP, (c) header for UDP, (d) header for IP, and (e) application PDU.

Scatter/Gather buffer management establishes at the sender side a joint data structure where individual protocol layer includes the pointer where its data is located in the memory. It means that the application protocol layer stores in the scatter/gather data structure the points to its application header, and application PDU, the RTP layer will store into the common structure the pointer to the RTP header information, the UDP layer stores into the common structure the pointer to its UDP header and IP stores into the common structure the pointer to its IP header. The last sending software module (e.g., MAC layer, or an extended IP function) that writes the whole data into the network device will then gather all the memory segments (headers and application PDU) using the scatter/gather joint pointer structure and writes it into the network device.

Comparison: Both offset and scatter/gather schemes have low memory and CPU bandwidth. The reasons are: (a) both schemes are not copying across multiple layers, hence low memory bandwidth usage. In offset case each layer writes into the joint memory space of application where there is a placeholder for each header, and in case of scatter/gather, the memory bandwidth is only needed to write into the joint pointer data structure, before all data is copied to network device. (b) both schemes have only low CPU usage since only small amount of data is moved, written across multiple layers.

- **(5 Points)** Let us assume Video-on-Demand (VOD) service with one client and one server. Let us assume that the server sends the Motion JPEG video at 20 frames per second and the client receives 20 frames per second. Let us assume that the end-to-end delay between client and server is 50 ms (in both directions) including the computational overheads on client and server side. Let us assume that the movie in fast forward model uses step-skipping method and step is equal to 5 (e.g., , if FF mode triggered from the beginning, only frames will be played: 1^{st}, 6^{th}, 11^{th}, 16^{th}, etc in FF mode). Under the above assumption, consider the following scenario:

- **The client receives streaming video and plays it on the screen. Suddenly somewhere in the middle of the movie, the client switches to fast forward operation.**

What is the minimal buffer size (in number of frames) at the client side for this scenario, so that the movie in fast forward mode will continue to play the FF frames at the rate of 20 frames per second?

Solution: First, the client must buffer at least one frame to avoid starvation. Second, in order to avoid delay for FF, the buffer should have all the frames up to the skip step frame, it means if the system has buffered frame 6, it should also have frames 7,8,9,10, 11, so that if the user triggers FF mode at the frame 6, the system can immediately jump to frame 11. Third, since the end-to-end delay is 50ms, it will take 100 ms for the system to get feedback to the server about changing from PLAY to FF and get the next FF frame 16 (which would violate the required rate for FF mode of 20 frames per second), so we need to buffer at least up to the frame 16, so that we have in buffer frames 11 and 16 to play if request for FF is hit at frame 6. Inbetween the system can deliver the next frame 21, 26, etc in a prefetch mode, i.e., send and fill up buffer with FF frames.

(Note: We do not have any requirements to come from FF to PLAY mode, only from PLAY to FF model).

So the minimal buffer size is frames 6, 7,8,9,10,11,12,13,14,15,16 (11 frames) to achieve 20 frames per second of playback when switching from PLAY to FF mode and the switch happens at the frame 6. This solution assumes that immediately when FF mode is triggered at frame 6, a feedback is sent to server to send frame 21 and more.

2. Problem – Networking

Assume Token Ring network with negligible ring latency (you can assume to be zero) and maximal token holding time of 10 ms.

- **(5 Points)** If we assume only one station transmitting video stream as the high-priority traffic and other stations transmitting best effort traffic, then can we guarantee that this video will play at 60 frames per second? Note that there is no buffering at the receiver side of the video stream and assume that one video frame fits into one Token Ring packet, and can be processed within the maximum token holding time. Explain your answer in detail (not just yes or no).

Solution: In case of one high priority stream, the access time $\tau \le 2*\tau_{max}$, where max token holding time $\tau_{max} = 10ms$, so the highest priority station gets the token every 20ms. To play video at 60 frames per second, the period it 1000/60 = 16.6 ms < 20ms, hence with these parameters of the token ring, the high priority station will not be able to play at 60 fps.

- **(5 Points)** Assume four stations (A,B,C,D) transmitting video streams as high-priority traffic. Assume the order A,B,C,D in terms of movement of the token, i.e., A gets the token first, then B, C, and D. Can we guarantee that video sent from station A can play at **30 frames per second? Can video sent from station D be played at 10 frames per** second? Explain your answers. Note that there is no buffering at receiving stations and assume that one video frame fits into one Token Ring packet. Explain your answer in detail.

Solution: Since we have four stations with high priorities, the time to access for any of the stations will be $t \le (M\text{-}1)\tau + 2\tau_{max}$, where $M=4$, so the time to access is 3*10+20 = 50ms for any of the **station** since they go in round robin manner passing the token . So under this condition the node A still cannot play since it needs token at least every 30ms to play at 30 frames per second and it will get it only every 50 ms. The station D can play at 10 frames per second, since it needs to get frames every 100 ms.

- **(5 Points)** Explain the Time Token Rotation Protocol in Token Ring, how it provides support for multimedia traffic. Specify conditions and constraints under which the support works.

Solution: The time token rotation protocol (TTRP) has two major functions: (a) passing a token among stations that represents the synchronization mechanism of access to the shared medium. A station who has token can access the medium. (b) sending/passing data among the stations from a station who has token. TTRP constraints how long each station can hold a token, i.e., it has so called token holding time (THT) parameter that provides fairness so that one station cannot starve others. Once THT expires, the station must release the token and move it to the next station. If TTRP supports priorities, stations can differentiate how urgently they need the token and access the medium.

- **(5 Points)** Consider transmitting MPEG-2 video over ATM networks. Which class of service would you consider to use for MPEG-2 video and why? Explain clearly.

Solution: ATM should use the VBR- variable bit rate traffic for MP2 video because MPEG2 video characteristics changes over time. It means that the video frames are not of equal size (I,P,B) , hence we will see variable bit rate coming out of the MPEG2 application.

3. Problem – Scheduling (20 Points)

Let us assume retrieval of three MPEG-2 videos with the following Group of Pictures (GOP) IPBBPI… Note that each movie is stored with the same GOP pattern at the media server. The processing time '*e*' of the individual frames fluctuates as follows: e(I) = 10ms +/- 2 ms, e(P) = 5 ms +/- 1ms, e(B) = 2ms +/- 2ms (the same execution time of I, P, B frames for each video). Let us assume that 'video 1' has the recorded frame rate 25 frames per second, 'video 2' has the recorded frame rate of 20 frames per second and 'video 3' has the recorded frame rate of 10 frames per second. Design the CPU soft-real-time scheduling framework for this workload to guarantee that the streams are schedulable at the media server. Specify admission control, reservation, scheduling policy, schedule how the tasks are scheduled and possibly adaptation policy in your scheduling framework if needed.

Solution: There are multiple approaches to the design of multimedia CPU scheduling framework at the media server.

One possibility would be that each video process retrieves the GOP (IPBBP) for each video. The video1 then needs to retrieve 5 GOP per second (recording rate of 25 fps) to send the 25 frames per second. The video2 needs to retrieve 4 GOP per second, and video3 needs to retrieve 2 GOPs per second. Each GOP will take in **worst case** scenario into account, i.e., the following amount of execution time e(IPBBP)=12+6+4+4+6 = 32ms. 5 GOP per second means period of 200 ms for video1, 4 GOP per second means period of 250 ms for video2 and 2 GOP per second means 500 ms as period for video3.

The admission control is then 32/200 + 32/250 + 32/500 = (160 + 128 + 64)/1000 = (352/1000) = 0.352 < ln2. Hence, this load can be scheduled via Rate-monotonic scheduling easily. So the soft-real-time framework will have admission control corresponding to the RMS, the reservation will reserve for each video process the worst case execution time of the GOP and the GOP rates for each video. The scheduler will use the RMS scheduling policy. Because we have been using the worst case processing time in admission control calculation, the anticipation is that there won't be any overruns hence no adaptation is needed.

Schedule is straight forward in this solution.

4. Problem – Synchronization (20 Points)

Assume multimedia presentation where a speaker uses pointer, audio and 5 visual clips (image, graphics, video, animation). The presentation takes 2 minutes long. The viewer sees the visual clips one after another. The media are used in the presentation as follows:

- The first clip – graphics of a map – occurs and the speaker starts to talk (audio) in sync with the graphics clip. The pointer moves to the upper left corner of the graphics clip. Audio is ahead of pointer by 100 ms. The speaker talks for 1 second. After this audio segment, the speaker starts to talk about the object in the lower right corner of the graphics clip and the pointer moves to the lower right corner with skew delay of 100 ms behind audio.
- 3 seconds after the start of the presentation, the speaker changes the graphics clip (clip 1) to video (clip 2) of length 20 seconds. With skew of 80 ms behind the video, the speaker starts talking. After 5 seconds of the video run time, the speaker starts to use the pointer. At this point video should be ahead of audio with 80 ms and audio should be ahead of pointer by 180ms.
- At 23 second from the start of the presentation, the speaker changes clip 2 to clip 3 which is an image. The speaker starts to talk in sync with the action of displaying the image. The speaker talks for 20 seconds.

- After that, clip4 – animation- is shown for 17 seconds. Audio and pointer are used by the speaker to explain the animation. Audio and pointer are in sync with the animation.
- The 5th visual clip – video – runs at the end for 1 minute with comments from the speaker. Audio and video are in sync.

1. **(10 Points)** Specify synchronization relations using time-axis specification.

   (see solution presented in class)

2. **(7 Points)** Specify synchronization relations using interval-based specification.

Audio1 while(0,0) Clip11;

Audio1 while(100ms,0) P1;

Audio1 before(0) Audio2;

Clip11 before(0) Clip12;

Audio2 while(100ms,0) P2;

Clip12 before(0) Video21;

Video21 while(80ms,0) Audio3;

Video 22 while(80ms,0) Audio4;

Audio4 while(160ms,0) P3;

Video 22 before(0) Clip3;

Audio4 before(0) Audio5;

Clip3 while(0,0) Audio5;

Clip3 before(0) clip4;

Clip4 while(0,0) Audio 6;

Clip4 while(0,0) P4;

Clip4 before(0) Clip5;

Clip5 while(0,0) Audio7;

3. **(3 Points)** Which synchronization specification out of the above is optimal for description of the multimedia presentation? Why is the other specification not suitable? What kind of information gets lost by the less optimal synchronization?

Solution: the time-axis sync specification is more accurate and hence is optimal for description of the multimedia presentation. The other spec, the interval-based spec does not allow expressing the time intervals, e. g;, that Audio1 lasts 1 second, or that Clip12 needs to finish after 3 seconds expires, etc). The interval-based spec looses the duration information, the start and end global time of each presentation segment.

5. Problem – Internet Protocols (20 Points)

- **(5 Points)** Specify two techniques that would be useful to speed up the current TCP or IP protocols and explain why they would speed up the process of current protocols. Provide one example for each technique in either the TCP or the IP protocol.

Solution:

(1) use caches of frequently used information. For example, in the routers, IP packets of the same session will go to the same destination, so one can cache the route since a lot of packets following first packet will follow the same route. One can also cache the TCP header fields of packets, e.g., source and destination information that can be the same for the same streaming session, for example.

(2) better lookup algorithms in the routing tables or in the connection tables for TCP. In this particular case one could use hashing using open chaining, which means that the head of each hashed link list keeps a cache of last accessed connection control blocks.

- **(10 Points)** Consider the following set of protocols (SIP, RTSP, RSVP, RTCP, RTP on top of UDP. If you want to design a protocol stack (control and data plane) for Video-On-Demand (VOD) service between client and server, (a) which protocols would you use and why, and (b) in which order would you apply your selected protocols? Explain how the protocol stack of selected protocols would be used.

Solution: To design the VOD service,

(a) *one would use the RTP protocol for transmission of the video* . RTP/UDP enables real-time transmission. The accompanying control protocol for RTP is the RTCP that would allow the receiver to provide feedback to the sender if some parameters need to be adjusted during the streaming session. Since VOD would use commands such as play, stop, pause, fast forward, rewind, one should also use the RTSP protocol, since it specifies the signaling to accomplish these control functions between client and server. If one also wants to have underlying reservation capabilities of the VOD traffic at the IP level, it is important to enable the RSVP protocol to start the integrated guaranteed services using the RSVP reservation protocol.

(b) *the order of the protocols should be*: Before RTP starts, RSVP should be invoked to reserve bandwidth for the VOD session. Once the resources are reserved, VOD traffic should be transmitted via RTP which is on top of UDP/IP. Concurrently to RTP RTCP and RTSP should run to (1) provide control feedback to the RTP (RTCP) about traffic and state of receiver, and (2) allow client to control the streaming via signaling (RTSP) to stop, play, pause, and otherwise control the VOD playback.

- **(5 Points**) What are the differences between SIP and RTSP protocols at the session layer of a multimedia protocol stack? Give 5 differences.

Solutions: (1) SIP is a session initiation protocol, RTSP is a control protocol to control streaming, (2) SIP protocol is suitable to initiate interactive session such as voice call, RTSP works for on-demand applications such as Video-on-demand, (3) SIP protocol supports mobile receivers, i.e., through location service it finds mobile user, RTSP does not have any mobility function built in, (4) SIP does not have a concept of cache control to cache media, RTSP has the concept of cache control and indicates if proxies are allowed to cache media; (5) SIP addressing is unique since it chooses email-like identifier, RTSP does not have that kind of requirement for SID.