EE549: Problem Set #5 Solutions

I. RINGS AND PRIORITIES

We first present a preliminary lemma:

Lemma 1: Suppose two arrival processes $X_1(t)$ and $X_2(t)$ with bit rates r_1 and r_2, respectively, enter a single-server queue with backlog $U(t) = U_1(t) + U_2(t)$, where $U_1(t)$ and $U_2(t)$ are the current bits of type 1 and 2, respectively. If $U(t)$ is rate stable, then the fluid-based departure processes $Y_1(t)$ and $Y_2(t)$ have rates r_1 and r_2.

Proof: If $U(t)$ is rate stable, then $\lim_{t\to\infty} U(t)/t = 0$ with probability 1. However, for $i \in \{1, 2\}$ we have:

$$0 \leq U_i(t) \leq U(t) \quad \text{for all } t \text{ and for } i \in \{1, 2\}$$

Dividing the above inequality by t and taking limits yields (with prob. 1):

$$\lim_{t\to\infty} \frac{U_1(t)}{t} = 0 \ \ , \ \ \lim_{t\to\infty} \frac{U_2(t)}{t} = 0$$

Further, for $i \in \{1, 2\}$, we have:

$$Y_i(t) = X_i(t) - U_i(t)$$

Thus we have (with prob. 1):

$$\lim_{t\to\infty} \frac{Y_i(t)}{t} = \lim_{t\to\infty} \frac{X_i(t)}{t} - \lim_{t\to\infty} \frac{U_i(t)}{t} = r_1 - 0 = r_1$$

Therefore, $Y_1(t)$ has rate r_1, and $Y_2(t)$ has rate r_2. □

Note that Lemma 1 also implies that if a single input of rate r enters a queue that is rate stable, then the output has rate r. Now consider node A of the ring. Since $X_1(t)$ data has preemptive priority, it does not see the other data entering the queue. Because $X_1(t)$ has rate r_1, and $r_1 \leq \mu_A$, it follows that the queue backlog $U_1^{(A)}(t)$ (data in queue A due to $X_1(t)$) is rate stable (by the rate-stability theorem). Hence, the output process $Y_1(t)$ (data from stream $X_1(t)$ that departs queue A) has rate r_1 (by Lemma 1).

Now consider node B of the ring. The input processes are $Y_1(t)$ and $X_2(t)$, having rates r_1 and r_2, respectively. Because $r_1 + r_2 \leq \mu_B$, by the rate-stability theorem we know node B is rate stable. It follows by Lemma 1 that the output process $Y_2(t)$ (bits from stream $X_2(t)$ departing node B) has rate r_2.

Now consider node C. The input processes are $Y_2(t)$ and $X_3(t)$, having rates r_2 and r_3, respectively. Because $r_2 + r_3 \leq \mu_C$, we know node C is rate stable. It follows that $Y_3(t)$ (bits from stream $X_3(t)$ that depart node C) has rate r_3.

Now consider node A. The input processes are $Y_3(t)$ and $X_1(t)$, having rates r_3 and r_1. Because $r_1 + r_3 \leq \mu_A$, node A is rate stable.

Thus, we have shown that all nodes (nodes A, B, C) are rate stable. Note that our proof only required the priority service assumption in node A to "get started," and the priority assumption was not needed in the other nodes. Indeed, rate stability would hold if these other nodes had any work conserving strategy (possibly FIFO).

b) The proof above (in part (a)) relied on first proving that one of the nodes in the ring was rate stable. This allowed us to conclude that the departure process from that node had a well defined rate, and enabled us to then sequentially show that all nodes of the ring are rate stable. However, establishing that one of the nodes was rate stable relied heavily on the priority service assumption, which allowed us to "start" the analysis by considering the highest priority data in the first queue it enters. This data is not influenced by the additional input stream into that node, and hence we can prove the high priority data output has a well defined rate. This would not be possible without the priority service, because then the $X_1(t)$ data in node A would interact with the $Y_3(t)$ data, and we do not yet know if $Y_3(t)$ has a well defined rate. Indeed, without priority service, just starting a "proof" by saying that $Y_3(t)$ has a well defined rate would be incorrect "circular" logic.

c) Because $X_1(t)$ data does not interact with any other data (as it has preemptive priority), we know by the I-O invariance property that:

$$Y_1^{(A)} \sim (r_1, \sigma_1)$$

This creates an "effective" time-varying server $\mu_B(t)$ for $X_2(t)$ data in node B, where $\mu_B(t) \sim (\mu_B - r_1, \sigma_1)$. Note that $r_2 \leq \mu_B - r_1$. It follows by the I-O invariance property for time-varying servers that:

$$Y_2^{(B)}(t) \sim (r_2, \sigma_1 + \sigma_2)$$

The $Y_2^{(B)}(t)$ data stream creates an effective time-varying server rate $\mu_C(t)$ for the $X_3(t)$ data in node C, where $\mu_C(t) \sim (\mu_C - r_2, \sigma_1 + \sigma_2)$. Because $r_3 \leq \mu_C - r_2$, we have by the I-O invariance property for time-varying servers:

$$Y_3^{(C)}(t) \sim (r_3, \sigma_1 + \sigma_2 + \sigma_3)$$

d) Worst case end-to-end delay bounds are:

$$\text{Delay of } X_1 \text{ data} \leq \sigma_1/\mu_A + \sigma_1/\mu_B$$

$$\text{Delay of } X_2 \text{ data} \leq \frac{\sigma_1 + \sigma_2}{\mu_B - r_1} + (\sigma_1 + \sigma_2)/\mu_C$$

$$\text{Delay of } X_3 \text{ data} \leq \frac{\sigma_1 + \sigma_2 + \sigma_3}{\mu_C - r_2} + \frac{2\sigma_1 + \sigma_2 + \sigma_3}{\mu_A - r_1}$$

II. Real Time Video

Let D_{min} and D_{max} represent the minimum possible end-to-end network delay, and the maximum possible end-to-end network delay, respectively. Parts (a)-(d) all ask to compute the receive buffer size K required for particular D_{min} and D_{max} properties in the network. Thus, it shall be useful to first prove the following result.

Theorem 1: Suppose the network has finite D_{min} and D_{max} values, representing the minimum and maximum possible network delay, respectively. Suppose the source sends packets periodically every T seconds. Suppose the destination starts playing the packets when the Kth packet is received in its buffer, and must play packets periodically every T seconds thereafter. Then there are no stalls or outages provided that the value K satisfies:

$$K \geq 1 + \frac{(D_{max} - D_{min})}{T}$$

Define K^* as the smallest integer that satisfies the above constraint. Thus, we have:

$$K^* = 1 + \left\lceil \frac{(D_{max} - D_{min})}{T} \right\rceil \tag{1}$$

Note that the value K^* does not depend on the absolute size of D_{max}, rather, it depends only on the difference $(D_{max} - D_{min})$. The value $(D_{max} - D_{min})$ can be viewed as the *worst case delay jitter*.

Proof: (Theorem 1) Let $\{t_i\}_{i=1}^{\infty}$ and $\{d_i\}_{i=1}^{\infty}$ represent the packet transmission times at the source and the packet arrival times at the destination, respectively. Specifically:

- t_i = time packet i is transmitted by the source (for $i \in \{1, 2, 3, \ldots\}$)
- d_i = time packet i is received at the destination (for $i \in \{1, 2, 3, \ldots, \}$)

Because packets are transmitted periodically every T seconds at the source, we have:

$$t_i = t_1 + (i-1)T \quad \text{for all } i \in \{1, 2, 3, \ldots\} \tag{2}$$

Because D_{min} and D_{max} are minimum and maximum network delays, we have:

$$t_i + D_{min} \leq d_i \leq t_i + D_{max} \quad \text{for all } i \in \{1, 2, 3, \ldots\} \tag{3}$$

Now define p_1 as the time when the first packet is played by the destination, and define p_i as the time required for packet i to be played at the destination if there are no stalls or outages. Because the first packet is played when packet K arrives at the destination, and packets must be played periodically thereafter, we have:

$$\begin{aligned} p_1 &= d_K \\ p_i &= d_K + (i-1)T \quad \text{for } i \in \{1, 2, 3, \ldots\} \end{aligned} \tag{4}$$

To ensure that there are no stalls or outages, we must ensure that all packets arrive before their playout deadlines. Specifically, it suffices to ensure that $d_i \leq p_i$ for all i. Because the video packets are received in FIFO order, we have $d_1 \leq d_2 \leq \ldots \leq d_K$. Therefore, the first K packets are in the receive buffer when the receiver first starts playing, and hence we trivially have $d_i \leq p_i$ for $i \in \{1, \ldots, K\}$. It suffices to prove that $d_{K+j} \leq p_{K+j}$ for $j \in \{1, 2, 3, \ldots\}$. To this end, note that for any $j \in \{1, 2, 3, \ldots\}$ we have:

$$d_{K+j} \leq t_{K+j} + D_{max} \tag{5}$$
$$\leq t_K + jT + D_{max} \tag{6}$$

where (5) follows from (3) and (6) follows from (2). Similarly, we have:

$$\begin{aligned} p_{K+j} &= p_1 + (K + j - 1)T \\ &= d_K + (K + j - 1)T \\ &\geq t_K + D_{min} + (K + j - 1)T \end{aligned} \tag{7}$$

where (7) follows from (3). Therfore, using (7) and (6), we have:

$$p_{K+j} - d_{K+j} \geq (K - 1)T - (D_{max} - D_{min})$$

It follows that, for all $j \in \{1, 2, 3, \ldots\}$, $(p_{K+J} - d_{K+j}) \geq 0$ whenever $(K - 1)T - (D_{max} - D_{min}) \geq 0$. Thus, if K satisfies $K \geq 1 + (D_{max} - D_{min})/T$, we have no stalls or outages, proving the result. □

We can now easily give answers for (a), (b), (c), (d):

a) We have $D_{min} = B/\mu_2 + B/\mu_3$. We now compute D_{max}. Note that $X_1(t) \sim (B/T, B)$. Thus, $Delay_1 \leq (B + \gamma)/\overline{\mu}_1$. The packet-based output process of node 1 is $Y_1(t) \sim (B/T, 2B + \gamma)$ (where the extra B is added due to the packet-based departures). Since there are no competing streams at node 2, we have:

$$Delay_2 \leq (2B + \gamma)/\mu_2 \ , \ Y_2(t) \sim (B/T, 3B + \gamma)$$

Note that a better bound is $Y_2(t) \sim (B/T, 2B + \gamma)$, where we notice that because node 2 has a constant server rate, $X_2(t) = 0$, and packets are fixed in size, we have that the I-O invariance holds without adding another B. However, we state the answer in terms of the looser bound $Y_2(t) \sim (B/T, 3B + \gamma)$. Because the $Y_2(t)$ packets are served in FIFO order with the competing $X_3(t)$ packets in node 3, we have:

$$Delay_3 \leq 3B + \gamma + \sigma$$

Therefore:

$$D_{max} = \frac{(B + \gamma)}{\overline{\mu}_1} + \frac{2B + \gamma}{\mu_2} + \frac{3B + \gamma + \sigma}{\mu_3}$$

Using these values for D_{min} and D_{max} in (1) yields:

$$K^* = 1 + \left\lceil \frac{1}{T} \left[\frac{B + \gamma}{\overline{\mu}_1} + \frac{B + \gamma}{\mu_2} + \frac{2B + \gamma + \sigma}{\mu_3} \right] \right\rceil$$

b) The value of $(D_{max} - D_{min})$ does not change when a propagation delay is introduced, and hence the answer for K^* is the same as part (a).

c) D_{min} is the same as before. However, if $X_2(t)$ is non-zero, we may be delayed by an additional packet service time of B/μ_2 in node 2. This would affect things by changing the effective service rate at node 2 to a time-varying service rate $\mu_2(t)$, and may also change the leaky bucket parameters for $Y_2(t)$. Overall, D_{max} would increase, and hence $D_{max} - D_{min}$ would increase, so that K^* might possibly increase.

d) The value of worst case delay at node 3 in this case would not change, as the worst case delay in this FIFO queue with leaky bucket inputs is the same value provided that the total input rate is less than or equal to μ_3.

III. Dynamic Power Allocation with Two Queues

a) $\phi_1(t) = 2$ for the first two seconds, providing 4 kbits of work. $\phi_2(t) = 4$ for the next two seconds (while $\phi_1(t) = 0$ during this time), providing 8 kbits of work. Thus, at time 4 we have:

$$U_1(4) = 7 \text{ kbits} \quad , \quad U_2(4) = 3 \text{ kbits}$$

After time 4, we have:

$$\phi_1(t) = \log_2(2.5) = 1.32193 \text{ kbits/sec} \quad , \phi_2(t) = 2\log_2(2.5) = 2.64386 \text{ kbits/sec}$$

Thus, the first queue empties at time $t = 9.29529$. The second queue empties at time $t = 5.1347$. Thus, $T_{empty} = 9.29529$.

We also have:

$$\overline{P}_1 = \overline{P}_2 = \frac{(3)(2) + (1.5)(5.29529)}{9.29529} = 1.5$$

Therefore, $\overline{P}_1 + \overline{P}_2 \leq 3$.

b) Now let us use $P_1(t) = P_2(t) = 1.5$ for all t. Thus:

$$\phi_1(t) = 1.32193 \text{ for all } t, \phi_2(t) = 2.64386 \text{ for all } t$$

Hence, queue 1 empties at time $U_1(0)/(1.32193) = 8.3212$, and queue 2 empties at time $U_2(0)/(2.64286) = 4.16216$. Hence, $T_{empty} = 8.3212$, which is indeed better than part (a).

c) The optimal powers are constant p_1, p_2, and empty both queues at the same time. Thus, we need (where $p_1 + p_2 = 3$):

$$\frac{11}{\phi_1(p_1)} = \frac{11}{\phi_2(p_2)}$$

Thus, we need:

$$\log_2(1 + p_1) = 2\log_2(1 + (3 - p_1))$$

Hence:

$$(1 + p_1) = (4 - p_1)^2$$

Or:

$$p_1^2 - 9p_1 + 15 = 0$$

Solving the quadratic equation yields:

$$p_1 \in \{2.2087122, 6.79128\}$$

The answer must satisfy $0 \leq p_1 \leq 3$, and hence we have:

$$p_1^* = 2.2087122 \quad , p_2^* = 0.7912878$$

Under these power allocations, we have $\phi_1(p_1^*) = \phi_2(p_2^*) = 1.68199$. The minimum emptying time is thus $t = 11/1.68199 = 6.5399$, which indeed is better than the emptying times in parts (a) or (b).

IV. Minimum Clearance Time

a) The maximum transmission rate is 1 kbit/sec. Thus, it will take at least $U_3/1 = U_3$ units of time to empty queue 3. Thus: $T_{min}(\vec{U}) \geq U_3$.

Further, the sum transmission rate (summed over both channels) is at most 2 kbit/sec. Thus, it will take at least $(U_1 + U_2 + U_3)/2$ units of time to empty all bits in the system. Thus: $T_{min}(\vec{U}) \geq (U_1 + U_2 + U_3)/2$. Therefore:

$$T_{min}(\vec{U}) \geq \max[(U_1 + U_2 + U_3)/2, U_3]$$

b) Here we show that we can empty the system in *exactly* $\max[(U_1+U_2+U_3)/2, U_3]$ amount of time, and hence this time must be optimal (from part (a)).

Note that we assume $U_1 \leq U_2 \leq U_3$ for convenience.

- Case 1: Suppose $U_1 + U_2 \leq U_3$. In this case, consider the following power allocation strategy:

$$\begin{aligned} p_1(t) &= 1 \text{ for all } t \\ p_2(t) &= U_1/(U_1+U_2) \text{ for all } t \\ p_3(t) &= U_2/(U_1+U_2) \text{ for all } t \end{aligned}$$

Notice that this is a feasible power allocation strategy, as

$$p_1(t) + p_2(t) + p_3(t) \leq 2 \text{ for all } t$$

The queue 3 clearly empties in time U_3 under this strategy. The queue 1 empties in time $U_1 + U_2$. The queue 2 also empties in time U_1+U_2. However, $U_1+U_2 \leq U_3$, Thus, the total time required for the system to empty under this strategy is exactly U_3. Hence, in this case, we have:

$$T_{empty} = U_3 = \max[(U_1+U_2+U_3)/2, U_3]$$

and so $T_{min}(\vec{U}) = \max[(U_1+U_2+U_3)/2, U_3]$ in this case.

- Case 2: Suppose $U_1 + U_2 > U_3$. There are many power allocation policies that will empty the system in time $(U_1+U_2)/2$ (some of them are time-varying). Consider the following power allocation policy that has constant powers:

$$\begin{aligned} p_1^* &= \frac{2U_1}{U_1+U_2+U_3} \\ p_2^* &= \frac{2U_2}{U_1+U_2+U_3} \\ p_3^* &= \frac{2U_3}{U_1+U_2+U_3} \end{aligned}$$

Note that these are feasible, as $p_1^* + p_2^* + p_3^* = 2$. Also note that $0 \leq p_i^* \leq 1$ for $i \in \{1,2,3\}$. That is because (note that $U_1 \leq U_2 \leq U_3$):

$$\begin{aligned} p_1^* &= \frac{2U_1}{U_1+U_2+U_3} \leq \frac{U_1+U_2}{U_1+U_2+U_3} \leq 1 \\ p_2^* &= \frac{2U_2}{U_1+U_2+U_3} \leq \frac{U_2+U_3}{U_1+U_2+U_3} \leq 1 \\ p_3^* &= \frac{2U_3}{U_1+U_2+U_3} \leq \frac{U_1+U_2+U_3}{U_1+U_2+U_3} = 1 \end{aligned}$$

Therefore, we have constant rates $\mu_1^* = p_1^*, \mu_2^* = p_2^*, \mu_3^* = p_3^*$. The emptying time for each queue is thus:

$$\begin{aligned} T_{empty}[1] &= U_1/\mu_1^* = \frac{U_1+U_2+U_3}{2} \\ T_{empty}[2] &= U_2/\mu_2^* = \frac{U_1+U_2+U_3}{2} \\ T_{empty}[3] &= U_3/\mu_3^* = \frac{U_1+U_2+U_3}{2} \end{aligned}$$

Therefore, the entire system empties in time $(U_1+U_2+U_3)/2$. Note in this case that $(U_1+U_2+U_3) = max[(U_1+U_2+U_3)/2, U_3]$. Hence this must be optimal. It follows that $T_{min}(\vec{U}) = \max[(U_1+U_2+U_3)/2, U_3]$.

V. IMET WITH A PARTICULAR $T_{min}(\vec{U})$ FUNCTION

Let T be a frame size (to be chosen later). Let $\vec{U}_i$ be the backlog that arrives in frame i. We want to ensure that $T_{min}(\vec{U}_i) \leq T$. We have:

$$\begin{aligned} T_{min}(\vec{U}) &\leq T_{min}(\vec{r}T + \vec{\sigma}) \\ &= \max[5, r_1T + \sigma_1 + (2r_2T + 2\sigma_2), 3r_1T + 3\sigma_1 + r_2T + \sigma_2] \\ &= \max[5, (r_1+2r_2)T + (\sigma_1 + 2\sigma_2), (3r_1 + r_2)T + (3\sigma_1 + \sigma_2)] \\ &\leq \max[5, (1-\epsilon)T + (\sigma_1+2\sigma_2), (1-\epsilon)T + (3\sigma_1+\sigma_2)] \\ &\leq \max[5, (1-\epsilon)T + \max[(\sigma_1+2\sigma_2), (3\sigma_1+\sigma_2)]] \end{aligned} \tag{8}$$

Suppose now that $T \geq 5$ and $T \geq \max[(\sigma_1 + 2\sigma_2), (3\sigma_1 + \sigma_2)]/\epsilon$. Then from (8) we have:

$$T_{min}(\vec{U}) \leq \max[5, (1-\epsilon)T + \epsilon T] \leq \max[5, T] \leq T$$

Thus, we have no outages or packet drops if the frame size T satisfies:

$$T \geq \max\left[5, \frac{\max[(\sigma_1 + 2\sigma_2), (3\sigma_1 + \sigma_2)]}{\epsilon}\right]$$

VI. Variable Frame IMET

a) First note that, by definition of T, we have:

$$z \leq T \ , \ \frac{T_{min}(\vec{\sigma})}{(1-\rho)} \leq T$$

and hence:

$$T_{min}(\vec{\sigma}) \leq (1-\rho)T \tag{9}$$

Now suppose that $T_m \leq T$ (this is true for the first frame $m = 1$, because $T_1 = z \leq T$). We show that this implies $T_{m+1} \leq T$. By induction, the result thus holds for all frames $m \in \{1, 2, 3, \ldots\}$.

Let $\vec{U}_m$ denote the vector of data that arrived during frame m. There are two cases:

- Case 1: $\vec{U}_m = \vec{0}$. In this case, $T_{m+1} = z \leq T$, and so we are done.
- Case 2: $\vec{U}_m \neq \vec{0}$. In this case, we have $T_{m+1} = T_{min}(\vec{U}_m)$. However, because frame m had duration less than or equal to T, and arrivals are leaky bucket constrained, we know that:

$$\vec{U}_m \leq \vec{r}T + \vec{\sigma}$$

Therefore:

$$\begin{aligned} T_{m+1} = T_{min}(\vec{U}_m) &\leq T_{min}(\vec{r}T + \vec{\sigma}) \\ &\leq T_{min}(\vec{r}T) + T_{min}(\vec{\sigma}) \\ &= T_{min}\left((\vec{r}/\rho)\rho T\right) + T_{min}(\vec{\sigma}) \\ &\leq \rho T + T_{min}(\vec{\sigma}) \qquad (10) \\ &\leq \rho T + (1-\rho)T \qquad (11) \\ &= T \end{aligned}$$

where inequality (10) follows because $\vec{r}/\rho \in \Lambda$, and inequality (11) follows from (9). Thus, $T_{m+1} \leq T$, and we are done.

b) Advantages/Disadvantages of variable frame IMET in comparison to fixed-frame IMET

Advantages:

- Not as much idle time in comparison to fixed-frame IMET. This can significantly reduce delay.
- The variable-frame IMET adapts the frame size to the inputs as needed. There is no need to know, estimate, or bound the $\vec{r}$, $\vec{\sigma}$, or ρ parameters.
- The delay bound for variable-frame IMET is in terms of the actual $\vec{r}, \vec{\sigma}, \rho$ parameters, and hence is much better than a delay bound based on bounds for these parameters. For example, consider a fixed-frame IMET designed for $\rho = 0.9$, but with actual inputs that satisfy $\rho = 0.4$. The frame size and delay would be unnecessarily large in the fixed-frame IMET, but would adapt to the $\rho = 0.4$ parameter in variable-frame IMET.

Disadvantages:

- The fixed-frame size IMET algorithm ensures the worst case delay of non-dropped data is at most $2T_{fix}$ (for arbitrary inputs), for a value T_{fix} that is specified in advance. However, for arbitrary or unknown inputs, there is no a-priori known bound on the frame size for variable-frame IMET.
- Because there is no bound on frame size for variable-frame IMET, buffering can be large, particularly when inputs are not leaky bucket or are outside the capacity region.

- If the input rate is outside of the capacity region, the frame sizes in variable-frame IMET will get larger and larger, growing to infinity. The fixed-frame IMET ensures more predictability, and will start dropping data in this case.

An improved algorithm would likely be one that uses variable frames, but with some maximum frame size T_{max} that is used to overcome the disadvantages listed above. If the value T_{max} is set to the frame size T_{fix} of the fixed-frame IMET, then the algorithm would provide all the predictability advantages of fixed-frame IMET while also providing adaptive frame sizing to improve delay in cases when inputs are small.

VII. Little's Theorem

a)

$$\overline{W} = \frac{2.2 + 4.6 + 12.0}{\lambda_1 + \lambda_2}$$

b)

$$\overline{W}_1 = \frac{2.0 + 2.4 + 6.0}{\lambda_1}$$

Clearly we have:

$$\begin{aligned}
\overline{L}_1^{(2)} &= \overline{L}_1 - \overline{L}_1^{(1)} = 0.2 \\
\overline{L}_2^{(2)} &= \overline{L}_2 - \overline{L}_2^{(1)} = 2.2 \\
\overline{L}_3^{(2)} &= \overline{L}_3 - \overline{L}_3^{(1)} = 6.0
\end{aligned}$$

Therefore:

$$\overline{W}^{(2)} = \frac{0.2 + 2.2 + 6.0}{\lambda_2}$$