**Problem Collection**

**ECE 466: Computer Networks II**

This copy is distributed to aid with exam preparation for students in ECE 466 in Winter 2020, with the understanding that the copy will not be posted on a public site, e.g., http://courses.skule.ca/.

Note: The prepared solutions are generally more detailed than what is required for a correct solution.

- **Min-Plus Algebra.**

1. Assume that $F$ and $G$ are arbitrary non-decreasing one-sided functions. Show that the following statements are false for every value of $t$:
   
   (a) $F \ominus G(t) < F(t)$,
   (b) $F \otimes G(t) > F(t)$.

2. Let $F$ be defined as
   
   $F(t) = \begin{cases} 
   0, & t \leq 0 \\
   t^2, & t > 0 
   \end{cases}$.

   (a) Provide an expression for $F \ominus F(t)$ for $t > 0$.
   (b) Provide an expression for $F \otimes F(t)$ for $t > 0$.

**Solution:**

1. Contradictions are quickly obtained by inserting $s = 0$ in the deconvolution and convolution. Since $G(0) = 0$, we get:

   i. $F \ominus G(t) = \sup_{s \geq 0} \{F(t + s) - G(s)\} \geq F(t + 0) - G(0) = F(t)$
   
   ii. $F \otimes G(t) = \inf_{0 \leq s \geq t} \{F(t - s) + G(s)\} \leq F(t - 0) + G(0) = F(t)$.

2. (a)

   
   \[
   f \ominus f(t) = \sup_{s \geq 0} \{(t + s)^2 I_{t+s>0} - s^2 I_{s>0}\} \\
   = \sup_{s>0} \{(t + s)^2 - s^2\} \\
   = \sup_{s>0} \{t^2 + 2st\} \\
   = \infty \\
   = \delta(t) .
   \]

   In the second line, we used that the supremum is not achieved for $s = 0$. In the third line, we see that the term $2st$ is increasing in $s$. Hence, the supremum is achieved for $s \to \infty$. 


(b) \[
f \otimes f(t) = \inf_{0 \leq s \leq t} \{(t - s)^2I_{t-s>0} + s^2I_{s>0}\}
= \min \left\{ \frac{t^2}{s=0}, \inf_{0<s<t} \{(t - s)^2 + s^2\}, \frac{t^2}{s=t} \right\}.
\]

In the range \(0 < s < t\), the function \(g(s) = (t - s)^2 + s^2\) takes the minimum at \(s = t/2\) (This can be seen by taking the derivative \(g'(s) = 4s - 2t\), setting to zero, and solving for \(s\)), which evaluates to \(g(t/2) = \frac{t^2}{2}\). So we get

\[
f \otimes f(t) = \min \left\{ \frac{t^2}{s=0}, \frac{t^2}{2}, \frac{t^2}{s=t} \right\} = \frac{t^2}{2}.
\]

• Min-Plus Algebra.

1. Assume that \(F\) and \(G\) are non-decreasing one-sided functions. Show that the following holds:

\[
F \otimes G \geq F.
\]

2. Let \(A\) and \(F\) be non-decreasing one-sided functions. Let \(\mathcal{E}_A\) denote the minimal envelope of \(A\). Show that the following three statements are equivalent:

- \(F\) is a deterministic traffic envelope for \(A\).
- \(\mathcal{E}_A \leq F\).
- \(A \otimes F = A\).

Solution:

1. We have

\[
F \otimes G = \sup_{s>0} \{F(t+s) - G(s)\}
\geq F(t+0) - G(0)
= F(t),
\]

where the second line uses \(s = 0\), and the third line uses that \(G\) is one-sided.

2. \((1) \iff (2)\): The definition of a traffic envelope is that for all \(t \geq 0\) and all \(s \geq 0\)

\[
F(t) \geq A(t+s) - A(s).
\]

This can be re-written as for all \(s\)

\[
F(s) \geq \sup_{s>0} \{A(t+s) - A(t)\} \quad \text{or} \quad F(s) \geq A \otimes A(s).
\]

Since \(\mathcal{E}_A = A \otimes A\), the definition of the envelope is the same as \(\mathcal{E}_A \leq F\).
(2) ⇔ (3): Note that - from (a) - we have \( A \otimes F \geq A \). Hence, we only need to be concerned with \( A \otimes F \leq A \). We will show that
\[
A \otimes F \leq A \iff A \otimes A \leq F
\]

Writing out \( A \otimes F \leq A \), we get
\[
\forall s \geq 0, \forall t \geq 0 : A(t + s) - F(s) \leq A(t) ,
\]
which is re-written as
\[
\forall s \geq 0, \forall t \geq 0 : A(t + s) - A(t) \leq F(s) .
\]

This is the equivalent to
\[
\forall s \geq 0 : \sup_{s \geq 0} \{ A(t + s) - A(t) \} \leq F(s) .
\]

The last line is \( A \otimes A \leq F \).

- Service curves.

Provide concise expressions for the service curves of the following network elements.

1. Provide a service curve of a network element which guarantees that the available service in any time interval of length \( \tau > 1 \) is at least \( \log(\tau) \).
2. Provide a service curve of a network element that delays all arriving traffic by 2 seconds.
3. Provide a network service curve for a sequence of two elements with service curves \( S_1 \) and \( S_2 \), described by:
   \[
   S_1(t) = \begin{cases} 
   0, & \text{if } t \leq 2 \\
   5, & \text{if } t > 2 
   \end{cases}, \quad
   S_2(t) = \begin{cases} 
   0, & \text{if } t \leq 3 \\
   t, & \text{if } t > 3 
   \end{cases} .
   \]

Solution:

1. \[
   S(t) = \begin{cases} 
   \log t, & \text{if } t > 1 \\
   0, & \text{if } t \leq 1 
   \end{cases} .
   \]

2. \[
   S(t) = \delta(t) = \begin{cases} 
   \infty, & \text{if } t > 2 \text{ seconds} \\
   0, & \text{if } t \leq 2 \text{ seconds} 
   \end{cases} .
   \]

3. The service curve is \( S_1 \otimes S_2 \), which must be computed. Here is a plot of \( S_1 \) and \( S_2 \).
We see that both service curves can be expressed as a convolution with a delay function as follows:

\[ S_1(t) = \delta_2 \otimes S'_1(t), \]
\[ S_2(t) = \delta_3 \otimes S'_2(t), \]

where

\[ S'_1(t) = \begin{cases} 0, & \text{if } t \leq 0 \\ 5, & \text{if } t > 0 \end{cases}, \quad S'_2(t) = \begin{cases} 0, & \text{if } t \leq 0 \\ 3 + t, & \text{if } t > 0 \end{cases}. \]

So far, we have

\[ S_1 \otimes S_2(t) = \delta_2 \otimes S'_1 \otimes \delta_3 \otimes S'_2(t) = S'_1 \otimes S'_2 \otimes \delta_5(t), \]

What remains to be done is to convolve \( S'_1 \otimes S'_2 \). We can use that both \( S'_1 \) and \( S'_2 \), yielding

\[ S'_1 \otimes S'_2(t) = \min\{S'_1(t), S_2(t)\} = \min\{t + 3, 5\}. \]

Since \( f \otimes \delta_5(t) = f(t - 5) \), we obtain

\[ S'_1 \otimes S'_2 \otimes \delta_5(t) = \min\{(t - 5) + 3, 5\} \cdot I_{t>5} = \min\{t - 2, 5\} \cdot I_{t>5}. \]

Alternatively, one can compute the convolution:

\[ S'_1 \otimes S'_2(t) = \inf_{0 \leq s \leq t} \{(t + 3 - s)I_{t-s>0} + 5I_{s>0}\} \]
\[ = \min \left\{ t + 3, \inf_{0 \leq s \leq t} \{(t + 3 - s) + 5\} \right\} \]
\[ = \min\{t + 3, 5\}. \]

The last line holds since

\[ \inf_{0 \leq s \leq t} \{(t + 3 - s) + 5\} = 8 < 5. \]

Here is a plot:
• Minimum envelope.
Consider the arrival of a sequence of video frames that arrive at time units \( n = 1, 2, \ldots \). The cumulative arrival function \( A(n) \) gives the size of the first \( n \) video frames in kilobytes (kB), with the following values:

\[
\begin{array}{cccccccc}
    n = & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
A(n) = & 20 & 40 & 90 & 100 & 130 & 180 & 210 \\
\end{array}
\]

1. Determine the values of the minimum envelope \( E_A(n) \) for \( n = 0, 1, 2, \ldots, 7 \).
2. Assume that the arrival function \( A(n) \) is regulated by a token bucket which enforces the envelope function \( E \), given by

\[
E(n) = \begin{cases} 
0 & \text{if } n = 0 \\
b + rn & \text{if } n = 1, 2, \ldots 
\end{cases}
\]

Determine the value of \( r > 0 \) and \( b > 0 \) subject to the following constraints:
- \( E(7) = E_A(7) \).
- Traffic is never delayed at the token bucket.
- The bucket size \( b \) is as small as possible.

Solutions:
1. Let \( a_n \) be the size of the \( n \)-th frame. Since \( a_n = A(n) - A(n - 1) \) (assuming \( A(0) = 0 \)), we obtain

\[
\begin{array}{cccccccc}
    n = & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
A(n) = & 20 & 40 & 90 & 100 & 130 & 180 & 210 \\
a_n = & 20 & 20 & 50 & 10 & 30 & 50 & 30 \\
\end{array}
\]

We use the discrete time version of the minimal envelope, given by

\[
E_A(n) = \max_{i=0,1,\ldots,7-n} \{A(n+i) - A(i)\}.
\]

Thus, \( E_A(0) = 0 \), \( E_A(1) \) is equal to the largest frame, \( E_A(2) \) is equal to the largest sum of a sequence of two consecutive frames, and so on. We obtain

\[
E_A(1) = \max \{a_1, a_2, a_3, a_4, a_5, a_6, a_7\} = a_3 = 50,
\]

\[
E_A(2) = \max \{a_1 + a_2, a_2 + a_3, \ldots, a_6 + a_7\} = a_5 + a_6 = a_6 + a_7 = 80,
\]

\[
E_A(3) = \max \{a_1 + a_2 + a_3, a_2 + a_3 + a_4, \ldots, a_5 + a_6 + a_7\} = a_5 + a_6 + a_7 = 110,
\]

\[
E_A(4) = \max \{a_1 + a_2 + a_3 + a_4, \ldots, a_4 + a_5 + a_6 + a_7\} = a_3 + a_4 + a_5 + a_6 = 140,
\]

\[
E_A(5) = \max \{a_1 + a_2 + a_3 + a_4 + a_5, \ldots, a_3 + a_4 + a_5 + a_6 + a_7\} = a_3 + a_4 + a_5 + a_6 + a_7 = 170,
\]

\[
E_A(6) = \max \{a_1 + a_2 + a_3 + a_4 + a_5 + a_6, a_2 + a_3 + a_4 + a_5 + a_6 + a_7\} = a_2 + a_3 + a_4 + a_5 + a_6 + a_7 = 190,
\]

\[
E_A(7) = a_1 + a_2 + a_3 + a_4 + a_5 + a_6 + a_7 = 210.
\]
Here is the result in a table:

<table>
<thead>
<tr>
<th>n</th>
<th>( E_A(n) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>50</td>
</tr>
<tr>
<td>2</td>
<td>80</td>
</tr>
<tr>
<td>3</td>
<td>110</td>
</tr>
<tr>
<td>4</td>
<td>140</td>
</tr>
<tr>
<td>5</td>
<td>170</td>
</tr>
<tr>
<td>6</td>
<td>190</td>
</tr>
<tr>
<td>7</td>
<td>210</td>
</tr>
</tbody>
</table>

2. The condition that traffic is never delayed in the buffer of the token bucket corresponds to \( E(n) \geq E_A(n) \) for all \( n \). The figure shows a graphical depiction of the solution with \( A(n) \) and \( E_A \). Since \( E(0) = 0 \) by definition and \( E(7) = 210 \) by requirement, the problem is to place a point \( b \) to the smallest point on the y-axis, so that \( E(n) \geq E_A(n) \) holds for all \( n \). The value of \( r \) is then implied. We obtain

\[
b = 70 , r = 20 .
\]

- **Arrivals and Envelopes** Consider a flow with fluid-flow traffic where arrivals follow a periodic On-Off pattern with time period \( T \). For a time period \( T_1 \) (\( T_1 < T \)), the flow is transmitting at rate \( p \), followed by a time period of length \( T - T_1 \), where the flow is not sending. Thus, the rate of the traffic flow at time \( t \geq 0 \), denoted by \( r(t) \), is:

\[
r(t) = \begin{cases} 
p , & \text{if } 0 \leq t - \lfloor \frac{t}{T} \rfloor \leq T_1 \\ 0 , & \text{if } t - \lfloor \frac{t}{T} \rfloor > T_1 \end{cases} .
\]

1. Draw a sketch of the arrival function \( A(t) \) for \( t \in [0, 3T] \). Indicate relevant values on the x-axis and y-axis.

2. Provide an expression for the minimal envelope \( E_A(t) \) in terms of \( T, T_1, \) and \( p \).
3. Provide the parameters of a token bucket envelope \( E(t) = (B + Rt)I_{t>0} \), such that (1) \( E \) is an envelope for \( A \), and (2) the value of \( R \) is as small as possible.

**Solution:**

1. (See figure)

2. The arrival function \( A \) is its own minimal envelope, that is, \( E_A = A \), with

\[
E_A(t) = \begin{cases} 
(t - \lfloor \frac{t}{T} \rfloor) p + \lfloor \frac{t}{T} \rfloor T_1 p , & 0 \leq t - \lfloor \frac{t}{T} \rfloor \leq T_1 \\
\lfloor \frac{t}{T} \rfloor T_1 p , & t - \lfloor \frac{t}{T} \rfloor \geq T_1 .
\end{cases}
\]

3. See the red line in the figure. The rate must be at least \( R = \frac{pT_1}{T} \), the average rate of the flow. The value of \( B \) must be chosen so that \( B + \frac{pT_1}{T} \cdot t \geq A(t) \). This can be done by ensuring that the envelope is equal to the arrival curve at time \( T_1 \):

\[
B + \frac{pT_1}{T} \cdot T_1 = pT_1 .
\]

This gives:

\[
B = pT_1(1 - T_1/T) .
\]

**Performance Bounds**

Consider a flow with arrival function \( A \). The arrivals comply with a traffic envelope \( E \) \((A \sim E)\), where \( E \) is given by

\[
E(t) = \min \{3 + 4t, 6 + 3t\} \quad \forall t > 0 .
\]

The arrivals of the flow enter a network element with service curve

\[
S(t) = \min \{Pt, B + Rt\} \quad \forall t > 0 .
\]

with \( P \geq R \). The network element has a fixed buffer size equal to \( X \geq 3 \).

1. Assume that \( B = +\infty \). Find the smallest value of \( P \) which ensures that there is no buffer overflow.

2. Do not assume that \( B = +\infty \). Instead, assume that \( P \) has the value computed in part (a). Find the values of \( B \) and \( R \) such that there is no buffer overflow and such that the sum \( B + R \) is minimized.
Solution:

1. Since \( B = +\infty \), we have \( S(t) = Pt \). The condition that the maximum backlog does not exceed is
\[
\sup_{t \geq 0} \{ E(t) - S(t) \} = \sup_{t \geq 0} \{ \min \{ 3 + 4t, 6 + 3t \} - Pt \} \leq X .
\]
The figure shows the solution space.

The maximum backlog is reached when \( 3 + 4t = 6 + 3t \), i.e., when \( t = 3 \). Then, the backlog is:
\[
E(t) - S(t) = 3 + 4 \cdot 3 - P \cdot 3 \leq X \\
5 - X/3 \leq P .
\]
So, the minimum rate is \( P = 5 - X/3 \).

2. We can construct the values of \( B + Rt \) as follows, indicated by the purple dash-dotted line in the figure:
   
   - The rate \( R \) should satisfy \( R \geq 3 \), otherwise the backlog grows infinite for \( t \to \infty \).
   
   - We can get a solution, by selecting \( R = 3 \), and let \( Pt = B + 3t \) at \( t = 3 \), then the backlog is equal to \( X \) for \( t \geq 3 \). This gives:
     \[
     (5 - X/3)3 = B + 3 \cdot 3 \\
     B = 6 - X .
     \]
   
   - Now we vary the rate to \( R = 3 + \epsilon \) and compute the resulting value of \( B \) (Again, the backlog is \( X \) at time \( t = 3 \)).
     \[
     (5 - X/3)3 = B + 3 \cdot (3 + \epsilon) \\
     B = 6 - X - 3\epsilon .
     \]

We see that by increasing \( R \) by \( \epsilon \), the sum \( B + R \) decreases by \( 2\epsilon \). Thus, we should set \( B \) to the smallest value \( (B = 0) \). Solving \( B = 6 - X - 3\epsilon \) gives \( \epsilon = 2 - X/3 \).
So the smallest sum $B + R$ is obtained by setting $R = 3 + \varepsilon = 5 - X/3$ and $B = 0$.

**Performance Bound** Consider a flow with arrival function $A$ with traffic envelope $E$ given by

$$E(t) = \min \{3 + 4t, 6 + 3t\} \quad \forall t > 0 ,$$

and a greedy traffic shaper with service curve

$$S(t) = \min \{Pt, B + Rt\} \quad \forall t > 0 .$$

with $P \geq R$. The traffic shaper has a fixed buffer size equal to $X \geq 3$.

1. Assume that $B = +\infty$. Find the smallest value of $P$ which ensures that there is no buffer overflow.

2. We do not assume that $B = +\infty$, but we assume that $P$ has the value computed in part (a). Find the values of $B$ and $R$ such that there is no buffer overflow and such that the sum $B + R$ is minimized.

**Solution:**

1. Since $B = +\infty$, we have $S(t) = Pt$. The condition that the maximum backlog does not exceed is

$$\sup_{t \geq 0} \{E(t) - S(t)\} = \sup_{t \geq 0} \{\min \{3 + 4t, 6 + 3t\} - Pt\} \leq X .$$

The figure shows the solution space.

![Solution Diagram](image)

The maximum backlog is reached when $3 + 4t = 6 + 3t$, i.e., when $t = 3$. Then, the backlog is:

$$E(t) - S(t) = 3 + 4 \cdot 3 - P \cdot 3 \leq X$$

$$5 - X/3 \leq P .$$

So, the minimum rate is $P = 5 - X/3$.

2. We can construct the values of $B + Rt$ as follows, indicated by the purple dash-dotted line in the figure:
The rate $R$ should satisfy $R \geq 3$, otherwise the backlog grows infinite for $t \to \infty$.

We can get a solution, by selecting $R = 3$, and let $Pt = B + 3t$ at $t = 3$, then the backlog is equal to $X$ for $t \geq 3$. This gives:

$$(5 - X/3)3 = B + 3 \cdot 3$$

$$B = 6 - X .$$

Now we vary the rate to $R = 3 + \varepsilon$ and compute the resulting value of $B$ (Again, the backlog is $X$ at time $t = 3$).

$$(5 - X/3)3 = B + 3 \cdot (3 + \varepsilon)$$

$$B = 6 - X - 3\varepsilon .$$

We see that by increasing $R$ by $\varepsilon$, the sum $B + R$ decreases by $2\varepsilon$. Thus, we should set $B$ to the smallest value ($B = 0$). Solving $B = 6 - X - 3\varepsilon$ gives $\varepsilon = 2 - X/3$.

So the smallest sum $B + R$ is obtained by setting $R = 3 + \varepsilon = 5 - X/3$ and $B = 0$.

• Scheduling

Consider a variation of the WFQ scheduler which operates without system virtual time. In this scheduler, the virtual finishing time of the $n$th packet from flow $j$ is computed as follows:

$$F^n_j = \max \left( a^n_j, F^{n-1}_j \right) + \frac{L^n_j}{\phi_j} ,$$

with $F^0_j = 0$, and with

$a^n_j$ Arrival time of the $n$th packet from flow $j$ in milliseconds.

$L^n_j$ Size of the $n$th packet from flow $j$ in bits.

$\phi_j$ Rate guarantee of flow $j$ in Mbps.

Consider the following arrival scenario of two flows 1 and 2:

Each arrow indicates the arrival of a packet. Arrival times are in milliseconds. All packets have the same size of 1000 bits. The rate of the link is 1 Mbps. The rate guarantees are $\phi_1 = \phi_2 = 1/2$ Mbps.

(a) Given the scheduler described above, determine the virtual finishing times assigned to packets in the arrival scenario. Determine the order of packet transmission.

(b) For the same arrival scenario, construct the transmission order for the (idealized) fluid-flow WFQ scheduler.
(c) Compare the transmission schedules in (a) and (b) with respect to the goal of achieving a ‘fair allocation’ of the data rate.

Solutions:

<table>
<thead>
<tr>
<th>(flow #, packet #)</th>
<th>$a^n_j$</th>
<th>$F^n_j$</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1, 1)</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>(1, 2)</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>(1, 3)</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>(1, 4)</td>
<td>3</td>
<td>8</td>
</tr>
<tr>
<td>(2, 1)</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>(2, 2)</td>
<td>4</td>
<td>7</td>
</tr>
<tr>
<td>(2, 3)</td>
<td>5</td>
<td>9</td>
</tr>
</tbody>
</table>

(a) The feasible schedule is:

<table>
<thead>
<tr>
<th>feasible schedule</th>
<th>time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1,1</td>
<td>0</td>
</tr>
<tr>
<td>1,2</td>
<td>1</td>
</tr>
<tr>
<td>1,3</td>
<td>2</td>
</tr>
<tr>
<td>2,1</td>
<td>3</td>
</tr>
<tr>
<td>2,2</td>
<td>4</td>
</tr>
<tr>
<td>2,3</td>
<td>5</td>
</tr>
<tr>
<td>1,4</td>
<td>6</td>
</tr>
<tr>
<td>1,4</td>
<td>7</td>
</tr>
</tbody>
</table>

(b) Fluid-flow WFQ schedule is:

<table>
<thead>
<tr>
<th>fluid-flow WFQ schedule</th>
<th>time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1,1</td>
<td>0</td>
</tr>
<tr>
<td>1,2</td>
<td>1</td>
</tr>
<tr>
<td>1,3</td>
<td>2</td>
</tr>
<tr>
<td>2,1</td>
<td>3</td>
</tr>
<tr>
<td>1,4</td>
<td>4</td>
</tr>
<tr>
<td>2,2</td>
<td>5</td>
</tr>
<tr>
<td>2,3</td>
<td>6</td>
</tr>
<tr>
<td>2,3</td>
<td>7</td>
</tr>
</tbody>
</table>

The figures shows the feasible transmission order.

(c) In (a), packet (1, 4) is transmitted after (2, 2). This means that flow 1 is ‘punished’ for using bandwidth between [0, 3], even though there was no demand from other flows.

• Scheduling and Performance Bounds

Consider a buffered link with two priorities as shown in the figure:

```
A_H \sim E_H
```

```
A_L \sim E_L
```

High priority arrivals are denoted by $A_H$, and low priority arrivals are denoted by $A_L$. The link has rate $C$. The link transmits low priority traffic only if there is no high priority traffic.

The high priority arrivals conform to traffic envelope $E_H$ (i.e, $A_H \sim E_H$), and low priority arrivals conform to traffic envelope $E_L$ (i.e, $A_L \sim E_L$), which are given by:

\[
E_H(t) = \begin{cases} 
0 & \text{if } t \leq 0 \\
\min(P_{ht}, B_h + R_{ht}) & \text{if } t > 0
\end{cases}
\]

and
\[
E_L(t) = \begin{cases} 
0 & \text{if } t \leq 0 \\
B_l + R_l t & \text{if } t > 0
\end{cases}
\]
1. For each priority level, provide a service curve expression that characterizes the service available to the traffic from that priority. The expression must be in terms of $C, P_h, B_h, R_h, B_l$ and $R_l$.

2. Assume that $P_h > C > R_h > 0$ and $B_h > 0$. Express the service curve that characterizes the service available to the low priority traffic by a latency-rate service curve.

   A latency rate service curve has the form $S(t) = X(t-Y)I_{t>Y}$, where $X > 0$ and $Y \geq 0$ are constants. Your answer must provide the constants $X$ and $Y$ of the latency-rate service curve, in terms of $C, P_h, R_h,$ and $B_h$.

3. Let $D_L(t)$ denote the departures of the low priority traffic from the buffered link in the time interval $[0, t)$. Determine a traffic envelope for $D_L$, in terms of the parameters $C, P_h, B_h, R_h, B_l$. Show the steps of your solution.

**Solution:**

1. A lower service curve for the high priority traffic is

   \[ S_H(t) = Ct^+ . \]

   A lower service curve for the low priority traffic is

   \[ S_L(t) = [Ct - E_H(t)]^+ = [Ct - \min(P_h t, B_h + R_h t)]^+ . \]

   An alternative expression is $S_L(t) = \max\{(C - P_h)t, (C - R_h)t - B_h, 0\}$.

2. With the information $P_h > C > R_h > 0$, we can simplify further:

   \[
   S_L(t) = \max\{(C - P_h)t, (C - R_h)t - B_h, 0\}
   = \max\{\{(C - R_h)t - B_h, 0\} \quad \text{since}\ C - P_h < 0
   \]

   \[
   = (C - R_h)(t - \frac{B_h}{C - R_h})I_{t > \frac{B_h}{C - R_h}}
   \]

   The figure shows the solution:

   ![Figure showing the solution](image)

3. An output envelope can be obtained by deconvolving the envelope and the service curve for the low priority, i.e., $E_L \odot S_L$. Since the envelope is an affine function and the service curve is a latency-rate service curve, the deconvolution is known from derivations in the lecture:
We will compute the deconvolution, but use $T := \frac{B_h}{C - R_h}$, for simpler notation:

$$E_L \odot S_L(t) = \max \left( \sup_{s \leq T} \{B_l + R_l(t + s)\}, \sup_{s > T} \{(B_l + R_l(t + s) - (C - R_h) \cdot (s - T)\}\right)$$

$$= \begin{cases} B_l + R_l(t + T), & C - R_h \geq R_l \\ \infty, & C - R_h < R_l \end{cases}.$$  

Assuming that the system is not overloaded, i.e., $R_l + R_h \leq C$, and inserting the value of $T$, we get

$$E_L \odot S_L(t) = B_l + R_l \frac{B_h}{C - R_h} + R_l t.$$

**Statistical Multiplexing**

Consider $N$ traffic flows. The arrivals from each flow are governed by a random variable $X$ with a geometric distribution. Specifically, in any time slot, the probability that the number of arrivals from a single flow are equal to $k$ kilobytes (kB) is given by

$$P(X = k) = \begin{cases} 0, & k = 0 \\ \frac{1}{2^k}, & k = 1, 2, \ldots \end{cases}.$$  

Assume that the $N$ flows are i.i.d. (independent and identically distributed).

In every time slot, the arrivals are entered into a buffer with a given size. If arrivals to the buffer exceed the buffer size, the excess traffic is lost. The buffered data is removed from the buffer before the start of the next time slot, so that the buffer is empty at the beginning of each time slot.

The objective is to design the size of the buffer such that the probability that a loss occurs does not exceed a given threshold value.

**Hints:**

- The mean and variance of $X$ are given by $E[X] = 2$ and $\text{Var}[X] = 2$.
- The formula for a finite geometric series is given by $\sum_{k=a}^{b} p^k = \frac{p^a - p^{b+1}}{1 - p}$.
- Let $\Phi(\cdot)$ denote the cumulative distribution function of the standard normal distribution. To get $1 - \Phi(x) \approx \varepsilon$, you may use the approximation $z = \sqrt{\ln(2\pi\varepsilon)}$. 
1. Consider that there is one buffer for each each flow, i.e., we have $N$ buffers. Express the total size of all $N$ buffers needed so that the loss probability of each flow does not exceed $\varepsilon$.

2. Consider that there is a single buffer that is shared by all $N$ flows. Use the Central Limit Theorem to derive the buffer size needed so that the loss probability does not exceed $\varepsilon$.

3. Evaluate the solutions of (a) and (b), using $\varepsilon = 10^{-3}$ and $N = 1000$. Interpret the results.

Solution:

A random variable with geometric distribution is given by:

\[
\begin{array}{|c|c|}
\hline
0 < p < 1 & p = 1/2 \\
\hline
P(X = k) = p(1-p)^{k-1} & P(X = k) = \frac{1}{2^k} \\
E[X] = \frac{1}{p} & E[X] = 2 \\
\text{Var}[X] = \frac{1-p}{p^2} & \text{Var}[X] = 2 \\
M_X(s) = \frac{pe^s}{1 - (1-p)e^s} & M_X(s) = \frac{e^s/2}{1 - e^s/2} \\
\hline
\end{array}
\]

1. Without statistical multiplexing: For a single flow, the loss probability with buffer size $B$ is

\[
P_{\text{loss}} = P(X > B) = 1 - \sum_{k=1}^{B} P(X = k) \\
= 1 - \frac{1/2 - (1/2)^{B+1}}{1 - 1/2} \\
= 1 - 2(1/2 - (1/2)^{B+1}) \\
= \left(\frac{1}{2}\right)^B.
\]

Given $\varepsilon$, the loss probability of a single flow must satisfy

\[
\left(\frac{1}{2}\right)^B = \varepsilon.
\]

Solving for $B$, we obtain

\[
B \ln \frac{1}{2} = \ln \varepsilon \iff B = \frac{\ln \varepsilon}{\ln \frac{1}{2}},
\]

and for $N$ flow the total buffer space $B_{\text{total}}$ is

\[
B_{\text{total}} = NB = N \frac{\ln \varepsilon}{\ln \frac{1}{2}}.
\]
2. With statistical multiplexing:

(Variation of the problem: Given MGF, derive $E[X]$ and $\text{Var}[X]$. Even better to work with general $p$.)

The Central Limit Theorem gives

\[ P_{\text{loss}} \approx 1 - \Phi \left( \frac{B_{\text{total}} - N E[X]}{\sqrt{N \text{Var}[X]} \sqrt{N}} \right) = \varepsilon . \]

Using the (bad) approximation $x = \sqrt{\ln(2\pi\varepsilon)}$, we get

\[ \frac{B_{\text{total}} - 2N}{\sqrt{2N}} = \sqrt{\ln(2\pi\varepsilon)} , \]

and

\[ B_{\text{total}} = 2N + \sqrt{2N \ln(2\pi\varepsilon)} , \]

3. Without statistical multiplexing:

\[ B_{\text{total}} = N \ln \frac{\varepsilon}{\ln \frac{1}{2}} \approx 9965 \text{ kB} . \]

With statistical multiplexing:

\[ B_{\text{total}} = 2N + \sqrt{2N \ln(2\pi\varepsilon)} = 500 + \sqrt{2000 \sqrt{5.07} \approx 2100 \text{ kB}} . \]

The comparison shows the impact of statistical multiplexing. With statistical multiplexing, the total capacity is close to the average traffic rate from all sources.

- **Bufferless Multiplexer.**

Consider $N$ (fluid flow) traffic flows where the arrival rate $X$ of each flow at an arbitrary time instant is given by the distribution function:

\[ P(X \leq a) = 1 - e^{-\lambda a} , \]

where $\lambda > 0$ is a constant. Here, $P(X \leq a)$ is the probability that the rate of the flow does not exceed $a$.

Assume that we have $N$ flows, which are i.i.d. (independent and identically distributed).

The flows send the traffic to a bufferless multiplexer. The objective is to design the rate of the multiplexer such that the loss probability does not exceed a given value.

(The loss probability at a bufferless multiplexer is the probability that the total arrival rate at the multiplexer exceeds its capacity.)

1. Consider that we have one bufferless multiplexer for each flow, i.e., we have $N$ multiplexers. Express the total capacity (summed over all multiplexers) needed so that the loss probability of each flow does not exceed $\varepsilon$. The capacity is expressed in terms of $N, \lambda$, and $\varepsilon$.

2. Consider that we have one bufferless multiplexer for all $N$ flows. Use the Central Limit Theorem, to derive the capacity needed so that the loss probability does not exceed $\varepsilon$. The capacity is expressed in terms of $N, \lambda$, and $\varepsilon$.

To compute the argument $z$ so that $1 - \Phi(z) \approx \varepsilon$, use the approximation $z = \sqrt{\ln(2\pi\varepsilon)}$. (where $\Phi(z)$ is the probability distribution function of the standard Gaussian distribution).
3. Evaluate the solutions of (a) and (b), using $\lambda = 1$ and $\varepsilon = 10^{-3}$ for $N = 100$. Interpret the results.

**Solution:**

1. **Without statistical multiplexing:** For a single flow, the loss probability with capacity $c$ is
\[ P_{\text{loss}} = P(X > c) = e^{-\lambda c} = \varepsilon. \]

Given $\varepsilon$, the capacity for a single flow is therefore
\[ c = -\frac{\ln \varepsilon}{\lambda}, \]

and for $N$ flow the capacity $C_{\text{total}}$ is
\[ C_{\text{total}} = \frac{N}{\lambda} \ln(1/\varepsilon). \]

2. **With statistical multiplexing:** We need the expected value and the variance of $X$, given by
\[ E[X] = \frac{1}{\lambda}, \quad \text{Var}[X] = \frac{1}{\lambda^2}. \]

The Central Limit Theorem gives
\[ P_{\text{loss}} \approx 1 - \Phi \left( \frac{C_{\text{total}} - NE[X]}{\sqrt{N \text{Var}[X]}} \right) = \varepsilon. \]

Using the (bad) approximation $z = \sqrt{\ln(2\pi\varepsilon)}$, we get
\[ \frac{C_{\text{total}} - N/\lambda}{\sqrt{N/\lambda^2}} = \sqrt{\ln(2\pi\varepsilon)}, \]

and
\[ C_{\text{total}} = N/\lambda + \sqrt{N/\lambda^2} \sqrt{\ln(2\pi\varepsilon)}. \]

3. **Without statistical multiplexing:**
\[ C_{\text{total}} = \frac{N}{\lambda} \ln(1/\varepsilon) = 100 \ln(1000) = 690.8 \]

**With statistical multiplexing:**
\[ C_{\text{total}} = N/\lambda + \sqrt{N/\lambda^2} \sqrt{\ln(2\pi\varepsilon)} = 100 + \sqrt{100} \sqrt{5.07} = 122.5. \]

The comparison shows the impact of statistical multiplexing. With statistical multiplexing, the total capacity is close to the average traffic rate from all sources.