Problem 1 (20%). Generalizing Dijkstra: Suppose every edge has a speed in addition to having a weight (length). So, given an edge $l$, let $w_l > 0$ be its weight and $s_l > 0$ be its speed. Let the speed of a path be the speed of its slowest edge. Let also the length of a path be the sum of the weights of its edges. In this question we will study the problem of finding the fastest shortest path between every pair of nodes in a graph, i.e., of the multiple shortest paths between two nodes, we pick the fastest.

Define the (vector) “cost” of an edge $l$, as being $c_l = (w_l, s_l)$ and the sum of the costs of two links, $l$ and $m$ as being $c_l + c_m = (w_l + w_m, \min\{s_l, s_m\})$. This allows us to compute the cost of a path. Given a path $p$ let its cost be $c_p = (w_p, s_p)$.

(a). Suppose a graph had just two paths $p$ and $q$ with costs $c_p$ and $c_q$. What would be rules (in terms of $w_p, s_p, w_q, s_q$) for picking the fastest shortest path?

(b). Given any two paths $p$ and $q$, we will say that $c_p < c_q$ if path $p$ would be picked over path $q$ according to the rules in part (a). So, given a set of paths connecting the two nodes we can find the minimum cost path. Use the Dijkstra algorithm discussed in the lecture to find the fastest shortest paths from node 1 to all the nodes of the graph shown in Figure 1.

Figure 1: The graph for Problem 1.
(c). Show that we cannot generalize Dijkstra in a similar way to find the shortest fastest path, i.e., of the multiple fastest paths pick the shortest. Find criteria similar to (a) for this problem, but show that the algorithm picks the wrong path between nodes 1 and 4 of Figure 1.

Problem 2 (15%). Consider the network shown in Figure 2. The figure indicates the bandwidth of each link. The links are full-duplex. That is, they can transmit in both directions at the same time, with the indicated transmission rates.

Find the values of the rates \((x, y, z)\) that

a. maximize \(x + y + z\);

b. maximize the minimum value of \((x, y, z)\);

c. maximize \(2 \ln x + \ln y + \ln z\).

(In the last expression, \(\ln u\) designates the natural logarithm of \(u\) for \(u = x, y, z\).)

![Figure 2: The network in Problem 2.](image)

Problem 3 (15%). Two connections share a link with transmission rate \(C\) Mbps, as shown in Figure 3. Assume that the round-trip time for the connection with rate \(x\) (resp. \(y\)) is equal to \(S\) seconds (resp. \(T\) seconds). Use the diagram with which we explained the behavior of AIMD to predict the relative fractions of \(C\) that the two connections will obtain after convergence of AIMD.

![Figure 3: Connections sharing a router in Problem 3.](image)

For simplicity, we suggest that you assume that the two connections are synchronized when they increase or decrease their rate, even though the different values of RTT will introduce a lack of synchronization. That is, assume that shortly after \(x + y\) exceeds the value \(C\), both connections drop their rate by a factor 2.
Problem 4 (20%). Consider TCP with a given value of RTT and a timeout value that is 2RTT. Assume that the window size is constant and equal to W packets. Assume also that of the first W packets sent, exactly one is lost, and that any one of these W packets is equally likely to be lost. Assume also that no other packet will be lost thereafter. Calculate the average time it takes the source to receive the acknowledgments of the first 2W packets under each of the following two situations:

a. Fast retransmit with 3DA and Reno-like Go Back N with cumulative ACKs (for simplicity, assume that the window size is not reduced after the 3DA).

b. Selective ACKs (and retransmission of only the missing packet).

Problem 5 (15%). Chapter 4, Question 24

Problem 6 (15%). CIDR: Chapter 4, Question 39