

# ECE158a Assignment 3

Due date: November 10, 11:00pm

1. Consider a wireless link between a source node and a destination node. Let the time indexed by discrete time slots  $t \in \{0, 1, \dots\}$ , with one time slot corresponding to the transmission time of a packet. Due to the channel condition, the wireless link could be either in the ON state or the OFF state at each time slot  $t$ . If the link is in the ON state, then a packet could be successfully transmitted through the link; on the other hand, if the link is in the OFF state and the source node sends a packet, then that packet would be dropped by the link. Assume the link state to be independent over time and at each time slot, the link is in ON state with probability  $p$ , and in OFF state with probability  $1 - p$ . If a packet is received at the destination, the destination immediately transmits a acknowledgment to the source node (with a negligible transmission time). In contrast, if a packet is dropped in the wireless link, then the source node could only know that when the wait for acknowledgment is timed out. We assume the time out period is two time slots, in other words, if a packet is sent at slot 1 and is dropped, then the source node would only be able to know until the end of slot 2 (and could then attempt retransmission starting from slot 3).

The packets are enqueued in a first-in-first-out (FIFO) queue at the source node before being transmitted to the destination. The packet arrival in this queue is independent across time, and at each time slot, with probability  $\lambda$  a packet arrives, and with probability  $1 - \lambda$  no packet arrives.

- (a) Assume the source node is able to perceive the link state before transmission, and it only sends packet when the link state is ON. Let  $T$  be the time required for packet at the head-of-line of the queue to be served (transmitted with success). Determine the distribution of  $T$  assuming  $p$  is known. Based on the result, what is the condition required for  $\lambda$  such that the queue is stable (queue length does not grow unboundedly)?
  - (b) Following (a), and assume the queue is stable. Determine the mean waiting time in the queue for these packets.
  - (c) Now consider the source node that is not able to perceive the link state before transmission. At each time slot, if the source node is not waiting for an acknowledgment, then it simply attempts to transmit the head-of-line packet (if the queue is not empty). On the other hand, if the source node is waiting for an acknowledgment, then it does not transmit any packet. Again, determine the distribution of  $T$  and derive the condition on  $\lambda$  such that the queue is stable, and determine the mean waiting time in the queue for these packets.
2. Due to the complexity of finding good scheduling algorithms, some researchers have proposed practical sub-optimal algorithms such as the following algorithm: At each time, the algorithm selects  $d$  permutation matrices chosen uniformly at random from amongst the possible  $N!$  matchings; let the schedule at time  $t$  be the one that has the largest weight of the  $d$  selected matchings, where the weight of a matching is defined as the sum of the lengths of the queues selected by the matching. For example, if  $\mathbf{X} = [X_{ij}]$  is the matrix of queue lengths, and  $\mathbf{S} = [S_{ij}]$  is the matching, then the weight of the matching is defined by  $W = \sum_{ij} X_{ij} S_{ij}$ .
    - (a) Given the schedule length matrix, we define the size of a matching as the number of non-empty queues selected by the matching, and a maximum size matching is a matching that has the maximum size (which may not be unique). Construct an

example to show that the algorithm described above can result in a matching which is not of maximum size.

- (b) Consider an  $N \times N$  switch. Assuming  $\text{VOQ}_{ij}$  has a non-zero backlog at a given time; calculate a lower bound on the probability that this  $\text{VOQ}_{ij}$  remains unserved.

**Hint:** Consider the cases that  $\text{VOQ}_{ij}$  is not served in all randomly selected schedules.

- (c) Discuss what happens as  $N \rightarrow \infty$ , while keeping  $d \leq cN$ , where  $c > 0$  is a constant. In particular, find a condition on  $\lambda_{ij}$  for which, the queue becomes unstable (gets larger and larger).

**Hint:** Use the condition  $d \leq cN$  and the result from (b) to obtain an upper bound bound on the probability that the  $\text{VOQ}_{ij}$  is served.

3. Represent a directed graph having  $N$  vertices with the vertex set  $\mathcal{V} = \{1, 2, \dots, N\}$  and a  $N \times N$  weight matrix  $W$ , with  $w_{ij} = 0$  if there exists no edge from node  $i$  to  $j$ , and  $w_{ij} > 0$  otherwise.

- (a) Write a function in MATLAB (`func1.m`) or Python (`func1.py`) which implements Dijkstra's algorithm for finding the minimum weight path to some node  $T$  from all the other nodes. This function should take as input the weight matrix  $W$ , and the target node  $T$ , and output an  $N \times 1$  vector  $D$  containing the weights of the minimum weight paths from all the nodes to  $T$ .

- (b) Write another function in MATLAB (`func2.m`) or Python (`func2.py`) which finds the minimum weight path to some node  $T$  by the listing out all the paths and then sorting them. This function should also output an  $N \times 1$  vector containing the weights of the minimum weight paths from all the nodes to  $T$ .

- (c) Write a MATLAB function (`func3.m`) or Python (`func3.py`) which takes in as inputs  $(W, D, T, i)$  where  $i \in \mathcal{V} \setminus \{T\}$ ; and outputs the minimum weight path from  $i$  to  $T$ .

- (d) Finally, use these MATLAB (or Python) functions to do the following:

- Create a graph with  $N = 8$ , by placing edges between any two nodes randomly with probability  $p = 0.7$ , and assign weights randomly from the range  $[0, 5]$  to the edges created. Generate the vector  $D$  for  $T = 8$ , using the two functions `func1.m` and `func2.m` and calculate the ratio of the running time of `func2.m` to the running time of `func1.m`
- Create a graph with  $N = 20$  as before, and obtain the vector  $D$  for  $T = 20$  by using `func1(.m or .py)`. Then use `func3` to find the path from node  $i = 1$  to  $T$ .
- Create graphs for different values of  $N$  in the range  $[10, 100]$  as before. For every value of  $N$ , use `func1.m` to obtain  $D$  corresponding to  $T = 1$  and compute the average running time over 10 trials. Then plot the average running time versus  $N$ . Does this plot match the theoretical results?

## 1 Report Submission

Please prepare your report in PDF format. There will be a specifics part in Gradescope to upload your codes. Create a zip file that contains your report and the Matlab codes (`func1.m`, `func2.m`, `func3.m`), and then upload it through Gradescope.