

1. (40 points) Consider a network of capacity-constrained links in Fig. 1. The number next to each link represents the capacity of the link. Define the capacity of a path as the *minimum* capacity over all links on the path. Given a source node S , a destination node T , and a capacity requirement b , a capacity-constrained minimum-hop routing problem aims to find the path from S to T such that the number of hops is the smallest among all paths whose capacity is greater than or equal to b .

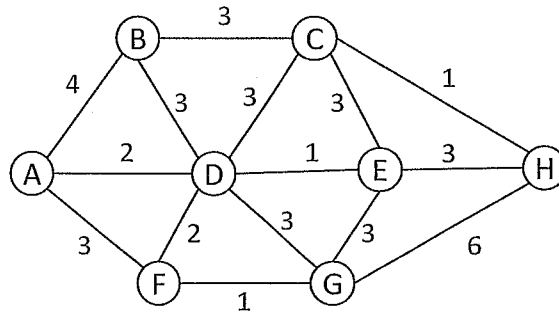


Figure 1: A network with capacity-constrained links.

Let $c(v, w)$ denote the capacity of the link connecting node v and node w , and let $c(v, w) = 0$ if the node v is not connected to node w .

- (a) (20 points) Carefully describe a version of the Bellman-Ford algorithm that can be used to find the capacity-constrained minimum-hop path from S to T .
- (b) (20 points) Using your algorithm from step (a), find the capacity-constrained minimum-hop path from node A to node H in Fig. 1 with the capacity requirement of $b = 2$. When you execute the Bellman-Ford algorithm, please update according the following node sequence: $G - F - E - D - C - B - A$. Clearly show intermediate steps.
2. (60 points) Today's data centers (e.g., those run by Google and Facebook) consume a huge amount of electricity. One way to reduce the electricity consumption is to slow

down the CPU clock of the computers when the workload is low. Suppose that the maximum speed of the CPU clock is 1. If the speed of the CPU clock is reduced to $\alpha < 1$, then the same job will take $1/\alpha$ times as much time to complete, compared to the case of running the CPU at the maximum speed. Let $e(\alpha)$, $0 < \alpha \leq 1$, be the power consumption of the computer when running at the reduced clock α . $e(\alpha)$ is in unit of Watts. In this problem, you will study the delay and energy tradeoff of a simple system with one computer.

Suppose that jobs arrive to a computer according to a Poisson process with rate λ . If the CPU clock of the computer is operating at the maximum speed of 1, then each job takes an exponentially-distributed amount of time to finish, with the mean execution time of $1/\mu$. In order to save electricity, the computer uses the following speed-control policy, parameterized by an integer K . If there are no jobs in the system, the speed of the computer's CPU clock is set to $1/K$. If there are $1 \leq n < K$ number of jobs in the system, the speed of the CPU clock is set to $n/K < 1$. If there are K or more jobs in the system, the speed of the computer's CPU clock is set to 1 (i.e., the maximum speed). All jobs are served in a first-come first-serve manner. When the CPU clock speed changes in the middle of a job, the job will simply continue with the remaining part at the new speed.

- (a) (20 points) Let P_n be the probability that there are n jobs in the system. Draw the state-transition diagram that can be used to find P_n , and write down the balance equations that can be used to find P_n . (*Hint: You do NOT need to solve P_n yet.*)
- (b) (20 points) Consider the case when $K = 2$. Let W be the random variable that denotes the delay from the time that a job arrives, to the time that the job is completed. Find $\mathbf{E}[W]$. (*Hint: you only need to consider two speed levels. Further, Table 1 on the next page contains some useful formula for summations.*)
- (c) (10 points) Consider the case when $K = 2$. Find the average power assumption (in Watts) of the computer. Your answer can depend on the function $e(\cdot)$.
- (d) (10 points) For the general case with arbitrary $K > 1$, under what range of λ will the system be stable (i.e., the number of jobs in the system remains finite)?

$$\sum_{n=0}^{K-1} p^n = \frac{1 - p^K}{1 - p}$$
$$\sum_{n=0}^{K-1} np^n = \frac{p[1 + (K - 1)p^K - Kp^{K-1}]}{(1 - p)^2}$$

Table 1: Formulas for some frequently-used summations

Write in Exam Book Only
