Algorithmic aspects of communication networks	**Assignment 1**
OR 739 Spring 1997	**January 29, 1997**

Due date: Friday, February 7.

1. Consider the on-line admission control problem discussed in lecture (maximizing the number of path accepted subject to routing at most k paths on any edge). Assume that we have a parameter L such that paths using more than L edges may not be accepted. Assume that $k = \Theta(\log L)$. You may also assume that k is at least small multiple of $\log_2 L$.

   (a) Give a variant of the on-line admission control algorithm that is $O(\log L)$ competitive. **Hints:**

   - Use the cost function $c(e) = 2^{f(e)} - 1$, where $f(e)$ is the number of paths routed on edge e.
   - Recall that given two nodes s and t, the minimum cost path from s to t using at most L edges can be computed in $O(mL)$ time.

   (b) Extend the solution also to the case when k is allowed to be larger than $\Theta(\log L)$. **Hint:** No need here for awkward calculus. Notice that m, the number of edges, did not play any role in your solution to the first part.

2. Extend the solution to the on-line admission control problem discussed in lecture (or your solution to the previous problem) to the case when a connection j if accepted brings a profit π_j. In this version we aim to maximize the sum of the profits of accepted connections, rather than the number of accepted connections. Assume that the profits are at least 1 and at most P. The solution should be $O(\log nP)$ competitive. You may make any assumption about what k is, that help you. **Hint:** Use the same cost function as before, and make decisions comparing the cost of the min-cost path to a scaled version of the profit that the connection brings.

3. Extend the solution of the first problem to the case when each connections lasts for a while. For this problem assume that when a connection is requested between two nodes s and t, you are immediately told also how long this connection will last. Assume that connection requests arrive at integral times, and last an integral amount. Your algorithm should be $O(\log LT)$-competitive, where T is the longest any connection lasts, and L is as in Problem 1. You may make any assumption about what k is, that help you. **Hint:** Consider multiple copies of your graph, one for each integral time.