CSCE 619-600: Computer Networks
Homework 3 due March 7, 2017

1 Purpose

This homework applies renewal-process techniques to distributed P2P systems.

2 Description

Many decentralized peer-to-peer networks are built by dynamically constructing a routable graph using other peers as intermediate nodes. The goal of this problem is to understand the likelihood of node isolation, or *local fault resilience*, of such networks.

Assume that each joining peer v spends $L \sim F(x)$ time units in the system, where L is determined by the peer's attention span and/or browsing habits. For the questions below, suppose that $E[L] < \infty$ holds. Upon join, each user obtains $k \geq 1$ uniformly random neighbors from the nodes already present in the network. If these neighbors all fail (i.e., depart the system) before v decides to leave, it becomes disconnected from the network and must re-join in a potentially different part of the graph. Many concurrent isolations may also cause the network to split into disjoint components, which is a highly undesirable event.

Note that for this problem we assume that failed neighbors are not replaced and that edges arriving to v from other joining users are not considered.

1. Define W_i to be the remaining lifetime of neighbor i. Obtain the distribution $G(x)$ of W_i and verify that its tail matches simulations using Pareto $F(x)$ with $\alpha = 3$. Using n at least 100K, simulate on/off processes $Z_i(t)$ shown in class. The length of off durations Y is not essential, e.g., you can make them uniform in $[0, 24]$ hours. Choose the join time t of user v to be large enough to have a fully randomized system, e.g., $t = 100(E[L]+E[Y])$.

2. Derive the expected time to isolation $E[T]$ and the probability p of this happening before v leaves the system. In the first two theorems, obtain respectively $E[T]$ and p using generic distributions of lifetime $F(x)$. In the next two theorems, simplify these expressions to a closed-form result (i.e., without any remaining integrals) for exponential and Pareto lifetimes. See the last section for helpful hints and use Wolfram Alpha as necessary.

3. How does Pareto β and exponential λ affect probability of isolation p?

4. Use simulations to verify your models of $E[T]$ and p. Since you confirmed the distribution of W_i, each iteration of the simulation now consists of a) drawing a random lifetime $L \sim F(x)$; b) drawing variables $W_1, \ldots, W_k$ from $G(x)$; c) computation of T based on $(W_1, \ldots, W_k)$, and d) comparison of L against T to determine whether isolation occurred or not. Repeating 10M times and averaging the result should yield $E[T]$ and $P(L > T)$. Armed with these results, show the following plots of models vs simulation:

   a) Exponential $F(x)$: vary $k \in [1, 10]$ while keeping $\lambda = 2$ fixed;

   b) Pareto $F(x)$: vary $\alpha \in [1.5, 10]$ while keeping $\beta = 1, k = 10$ fixed;

   c) Pareto $F(x)$: vary $k \in [1, 10]$ while keeping $\alpha = 3, \beta = 1$.

   This should give you six figures total (i.e., three for $E[T]$ and three for p).

5. Use the model to compute a numerical value of p under Pareto lifetimes with $\alpha = 1.05$ and $k = 30$. Assuming that 1 billion users join the system every day, obtain the average number of years between isolation events in this network. Hint: use the mean of the geometric distribution.

6. Explain how Pareto parameter α affects the result and why Pareto L always yields more resilient graphs than exponential L. Hint: show that $R(t)$ under Pareto is stochastically larger than L.

7. Analyze the model and determine what happens to Pareto $E[T]$ as $\alpha \to 2$. Similarly, uncover what happens to p as $\alpha \to 1$. You can use Matlab to plot both functions as α tends to the corresponding limit from above.

3 Integrals

You may find the following useful:

$$\frac{\alpha}{\beta} \int_0^\infty (1 + \tfrac{z}{\beta})^{-\alpha-1} \left(1 - (1 + \tfrac{z}{\beta})^{1-\alpha}\right)^k dz = \left({}_2F_1(\tfrac{\alpha}{\alpha-1}, -k; \tfrac{2\alpha-1}{\alpha-1}; z) z^{\alpha/(\alpha-1)}\right)\Big|_0^1,$$

where ${}_2F_1(a, b; c; z)$ is the Gauss hypergeometric function. Its value for $z = 0$ is 1 and for $z = 1$ is:

$$ {}_2F_1(a, b; c; 1) = \frac{\Gamma(c)\Gamma(c - b - a)}{\Gamma(c - a)\Gamma(c - b)}, \tag{1}$$

where $\Gamma(.)$ is the gamma function (available in Matlab under the same name).