# MBRA 2.3.0 Cascade Failure Simulation

September 7, 2010

The MBRA 2.3.0 Tool Window Simulation tab is used to perform analysis on the failure rate of a network. Its function is to estimate the cascade effect of a single failure on a network on the network as a whole. To save typing, the $n$ nodes and $m$ links are referred to together as nodes, where there are a total of $i = n + m$ of these nodes.

In order to simulate this effect, the algorithm calculates a CDF (cumulative distribution function) that allows it to pick a random node based on the Threat x Vulnerability values of every node. Once it has selected a random node $j$ to fail, the algorithm propagates that failure to adjacent nodes, repeating until there are no more adjacent nodes that are unvisited. Nodes may only be visited once, and if they do not fail, are safe (and do not cause their own adjacent nodes to be tested). Once the cascade has finished, the amount of "damage" that was done to the network is calculated by adding up the "consequence" value for each node that failed and dividing it by the sum of every "consequence" value of nodes in the network.

In order to end up with a useful amount of data, the algorithm collects the results of running the algorithm a large number of times in order to create a probability density function for the network. From the PDF, an Exceedence Probability for the network is calculated and displayed on the graph. The exceedence probability graph is the true goal of the whole algorithm. Its x-axis goes from [1, 100] and represents a percentage of the total number of nodes in the network. The graph's y-axis represents the probability that *at least* that percentage of nodes in the network will have failed. For example, if EP(60) = 30, then in 30% of all cascade failures, at least 60% of the network will fail.

**Algorithm 1** Network Cascade Failure

1. Repeat forever:
   (a) Construct a CDF of the network using Threat x Vulnerability values
   (b) Pick a random node from the CDF by selecting a random number $R \epsilon [0, 1]$. Iterate through the CDF, and when $CDF(x) \leq R < CDF(x + 1)$, pick node $x$ and add to queue Q.
   (c) A node or link $X$ fails if after picking a random $R \epsilon [0, 1)$, $R \leq (X_{vulnerability} * X_{threat})$
   (d) While there are nodes in Q:
      i. Select a node $i \epsilon Q$, and mark it as FAILED.
      ii. For all links $l$ adjacent to $i$ that have not been visited:
         A. If links failures are allowed, $l$ fails, the node $j$ it connects to $i$ fails, add $j$ to Q. Mark $l$ visited, but only mark $j$ visited if $l$ fails.
         B. If link failures are not allowed and node $j$ fails, add $j$ to Q and mark both $l$ and $j$ visited.
   (e) Find $k = \frac{\sum_{i \epsilon failed\ nodes} consequence_i}{\sum_{i \epsilon all\ nodes} consequence_i} * 100$
   (f) Increment the global PDF array: $pdf(k) + +$
   (g) Increment the global trial counter: $N + +$
   (h) Calculate the Exceedence Probability:
      i. $EP(100) = \frac{pdf(100)}{N}$
      ii. $\forall k \epsilon [1, 99]$, $EP(k) = \frac{pdf(k)}{N} + EP(k + 1)$