Engineering Networks for Optimal Robustness

Data Communication and Networks I [4005-700]

Christopher Wood

April 10, 2012

Abstract

Due to the growing pervasiveness of civilian and military networks for the transmission of safety-critical and real-time data, it is critically important that they are resistant to selective and random network node deletions. Network robustness is a measure of the performance and throughput responsiveness of a network in response to such deletions. The nature of this metrics lends itself to the application of percolation theory, which can be used to describe the behavior of connected clusters in a random graph. This theory can be utilized to design and construct optimally robust networks in order to yield the best performance in the event of node deletions.

This paper presents some background information on network robustness and its importance in modern communication systems, presents some recent advances made in the topic, and concludes with avenues of future work that can be explored by researchers in the field.

1 Introduction

Military and civilian communications have seen two common trends in rencent years: an increase in network-oriented operations and an increase in high-risk threats to such networks [1]. These operational efforts place high reliance on the underlying network infrastructure for communication, so it is vital that this communication medium is protected against emerging attacks that focus on specific nodes in the network or communication lines that join nodes together. In this context the type of network attacks are irrelevant; the focus is more aligned with the optimal topology of networks and technological aids that can be utilized to help handle any changes in this topology.

This focus can be seen by a significant increase in research oriented around robust network design that provides high throughput and connectivitiy among all nodes in the network, especially when specific nodes are intentionally or unintentionally deleted from the network. Subsequently, the robustness of such networks can be viewed as a qualitative or quantitative measure of the network's resilience to such topology changes.

The problem of designing such networks has lent itself as a useful application of both graph theory and percolation theory. Graph theory has been applied to mathematically analyze the robustness of networks represented as undirected graphs based on their levels of vertex and edge connectivity. Similarly, percolation theory has been applied to study the behavior of connected clusters in undirected network graphs.

This paper will focus on recent research efforts centered around both of these branches of mathematical theory and their application to network design. It will also present practical methods of network engineering that have been employed to help networks deal with topology changes dynamically. Lastly, it will discuss avenues for future research and open problems that have been posed by researchers in the field.

2 Fundamentals

It is natural to model any communication network as an undirected graph G, which has a fixed set of vertices (nodes) $V(G)$ and edges (links) $E(G)$ that represent physical connections between such vertices. For conveniene, we let $N = |V(G)|$ and $M = |E(G)|$. The topology of a network can thus be visualized graphically using elements from these two sets. For the remainder of this paper, we use the term vertex as a synonom for node and edge as a synonom for communication link. As an example of a graph representing a network, consider a completely connected network with n nodes in which every node can directly communicate with every other node can be seen as K_n, the complete graph on n vertices (that is, $|V(G)| = n$). In such a network, every node v can communicate with exactly $n-1$ other nodes, which means that its degree $deg(v) = n-1$.

In order to discuss the connectivity of networks, it is necessary to define the connectivity of such graphs in terms of both the vertices and edges. We introduce the following terms to use throughout the remainder of this paper.

Definition 1. A graph G is said to be **connected** if and only if $\forall u, v \in V(G)$ there exists a path between u and v.

3 Network Functionality

Network engineers strive for high performance networks that display high throughput and low latency between any two nodes in the same network. Several mathematical metrics based on the corresponding graph that represents a network have been proposed to reflect this need. For example, metrics such as the average geodesic path length between any two nodes in a network and vertex and edge betweeness (which are essentially measures of centralities located within a graph).

The average geodesic length L can be defined as follows,

$$L(d(v,w)) = \frac{1}{N(N-1)} \sum_{v \in V(G)} \sum_{w \neq v \in V(G)} d(v,w),$$

where $d(v,w)$ is the distance of the shortest path between vertices v and w, and $N(N-1)$ is the total number of pairs of vertices, independent of whether or not each pair represents an edge in $E(G)$. The most immediate result from this measurement is that large values for L indicate that the average length between any two nodes in the network is long, and thus the latency between two nodes will be proportionally large as well.

Another important metric that measures the functionality of a network is the measure of vertex and edge centrality in the network. Although a high measure of centrality may indicate more traffic funnels through a vertex or an edge, it also implies that any attacks on this vertex or edge would most likely have an negative impact on the traffic in the network by increasing the load on neighboring nodes and increasing the average geodesic path length. Although there is not a single definition for this metric, Holme et al [**?**] propose the use of the following definitions for vertex, $C_B(v)$ and edge $C_B(e)$ centrality.

$$C_B(v) = \sum_{w \neq x \in V(G)} \frac{\sigma_{wx}(v)}{\sigma_{wx}},$$

where $\sigma_{wx}(v)$ is the number of paths between w and x that pass through v and σ_{wx} is total of paths from w to x (notice that $\sigma_{wx}(v) \leq \sigma_{wx}$).

$$C_B(v) = \sum_{w \neq x \in V(G)} \frac{\sigma_{wx}(e)}{\sigma_{wx}}$$

As in the centrality measure for vertices, $\sigma_{wx}(e)$ is the number of paths between w and x that contain e and σ_{wx} is total of paths from w to x (notice again that $\sigma_{wx}(e) \leq \sigma_{wx}$).

It is important to note that the centrality of a vertex and its measure of centrality are not the same metrics. In fact, as will be shown in section 4.1, network attacks can vary based on the measure an adversary is trying to reduce.

Finally, we introduce a metric that characterizes the number of nodes (or rather, a fraction of the total nodes) that need to be removed in order for the graph to become globally disconnected.

4 Network Attacks

4.1 Attack Strategies

Attacks on large scale networks are not usually ad-hoc; they are based on a logical and structured strategy for dereasing the connectivity of the network by taking as little action as possible. Clearly, if one was to delete all nodes from a network, then that would yield the maximum decrease in connectivity. However, such attacks are not practical, so these strategies must be considered at a smaller scale.

From a general perspective, practical attacks are theoretically focused on the objective of decreasing the number of total links in the network or the average geodesic length (or both). Consider, for example, the situations of cutting communication cables or performing a DDOS attack on a node or server with a high measure of centrality. Such attacks would decrease the number of edges in the network graph and increase the average geodesic path length, respectively.

From the definitions presented in section **??**, we can see that the number of edges in the network is directly related to the degree of each vertex (in fact, we know that $2|E(G)| = \sum_{v \in V(G)} deg(v)$). On the other hand, the measure of centrality of a vertex or edge is more related to the average geodesic path length in the network. As such, we consider practical attack patterns that focus on decreasing both of these measurements by targeting individual vertices and/or edges, as well as randomized attacks that have no specific targets.

In general, most focused attacks fall under one of the following four categories [?].

- **ID removal** - initial degree distribution vertex/edge removal
- **IB removal** - initial betweenness distribution vertex/edge removal
- **RD removal** - recalculated degree distribution vertex/edge removal
- **RB removal** - recalculated betweenness distribution vertex/edge removal

Clearly, RD and RB attacks on vertices would yield the optimal results because they take a greedy approach to decrease the target metric. However, the implication of these attacks is that there exists an efficient and tractable way to measure these metrics after every change, which isn't always the case (especially when the topology of the network is unknown). Therefore, ID and IB attacks are more realistic, but they also assume some prior knowledge of the network infrastructure before the attack begins. Attacks that do not rely on this knowledge are referred to as random attacks, and are discussed in section 4.4.

Furthermore, it should be noted that both the ID and RD attacks are computationally less taxing than IB and RB attacks [?]. In fact, the time complexity of a successful ID attack (and subsequently, an entire RD attack), runs in linear time with N (i.e. $O(N)$), whereas the time complexity of betweenness-based attacks has a time complexity of $O(NL)$. The implication of this is that the adversary must make tradeoffs based on their knowledge of the network infrastructure.

4.2 Vertex Attacks

Based on the attack objectives discussed in the previous section, the following distinct attack strategies on network nodes have been identified .

4.3 Edge Attacks

TODO

4.4 Random Attacks

TODO

5 Network Robustness

Many different measures for network robustness have been proposed in recent years. All of which tend to use the notion of densely connected components in the corresponding graph. In this section we present a two unique instances of such measurements, one of which is based entirely on node topology and the other that is based on both nodes and their respective edges, and summarize their accuracy when applied to real networks.

5.1 Node-based Measurement

A natural way to think of network robustness is from the perspective of individual nodes, since they are usually the primary targets in malicious or non-malicious network attacks. Using this idea, Herrmann et al defined a concise equation for calculating the robustness of a network based on the size of connected components in the corresponding graph. Mathematically, this can be defined as follows [2]:

$$R_n = \frac{1}{n} \sum_{q=\frac{1}{n}}^{1} S(q)$$

In laymen terms, this robustness measurement computes the fractions of nodes in the largest connected cluster $S(q)$ after removing q nodes. This is an intuitive calculation, since the goal of engineering robust networks is to ensure the highest measure of connectivity in the event of any node deletions. Furthermore, it has been mathematically verified to represent the exact amount of nodes that need to be deleted for the network to collapse when targeted by high-degree adaptive attacks, which are a specific class of attacks that attempt to remove highly connected nodes from the network.

In their study of optimal graph structures that yield the highest resilience to such attacks, Herrmann et al found that most networks will exhibit onion topolgies, meaning that there are distinct layers of nodes that are connected, and that each layer i has more connectivity than its parent layer $i + 1$ [2]. An example of such a graph is shown in Figure 5.1.

Herrmann et al have also conducted research on optimization algorithms that increase this robustness measure while at the same time maintaining the distribution of vertex degrees throughout the network. This is an important concept, because their algorithm simply seeks to re-arrange node edges and connections to improve the resilience of the host network to any kinds of attacks. This algorithm can be described as follows:

Algorithm 1: Monte-Carlo-based Robustness Optimization

```
Choose two random edges $(a, b)$ and $(c, d)$ from the graph $G$.
```

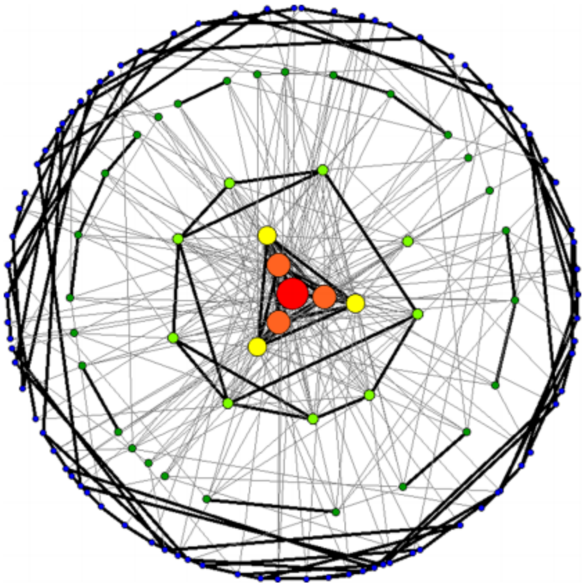

Figure 1: An example of a graph with the onion-like topology [2]

```
2: Replace these edges with $(a, c)$ and $(b, d)$.
3: If $R_{new} > R_{old}$, accept the swap and goto step 1. Otherwise, revert the swap and goto step 1.
```

Algorithm 1 is repeated until an ideal level of robustness has been obtained, albiet at the sake of sometimes massive computations (as is the case with Monte-Carlo methods).

5.2 Node- and Link-based Measurement

6 Network Traffic Load Balancing

6.1 Dynamic Load Balancing

References

[1] Anthony Dekker Bernard. Network robustness and graph topology.

[2] Hans J Herrmann, Christian M Schneider, Andr A Moreira, Jos S Andrade Jr, and Shlomo Havlin. Onion-like network topology enhances robustness against malicious attacks. *Journal of Statistical Mechanics: Theory and Experiment*, 2011(01):P01027, 2011.

[3] Petter Holme, Beom Jun Kim, Chang No Yoon, and Seung Kee Han. Attack vulnerability of complex networks. *Phys. Rev. E*, 65:056109, May 2002.