Characterization and Identification of Roles in TCP Connection Networks

Larissa Spinelli, Daniel R. Figueiredo

System Engineering and Computer Science Department, Federal University of Rio de Janeiro (UFRJ), Brazil
{larissa,daniel}@land.ufrj.br

Abstract: This paper presents a studies the identification of roles in TCP Connection Networks. Based on the characterization of the strucutral properties of these networks, we develop simple algorithms to identify roles of vertices when this information is not available (e.g., undirected connection graph).

Keywords: Connection Network, Anonymity

1. Introduction

A connection network is an abstraction to model the establishment of connections and exchange of information among a set of entities. In this model, entities are represented by vertices of a graph while connections and information exchange are represented by edges between vertices. Entities of a connection network can play different roles or have different attributes that are related to their functionality in the system. In the TCP connection network vertices (IP addresses) can be labeled according by their role as client (C) if only open TCP connections, server (S) if only receive TCP connections or both (C-S) if open and receive TCP connections. Such information can be obtained by inspecting network packet traces of backbone links, for example [2].

In this work, we are interested in identifying the role of vertices of a connection network using only simple structural information, namely whom *talks* to whom. In particular, we focus on the TCP connection network and the identification of the roles (C, S or C-S) using just an undirected graph without any edge weights. Our identification techniques always produce *consistent* labels and are based on the characterization of structural properties of the different roles. The proposed techniques are evaluated using real data traces and compared to assess their efficiency in identifying roles. Numerical results are very promising and indicate that it is possible to identify roles with a success rate of over 96%. Our work is related to [3], where the goal is to identify servers in anonymized networks.

2. Characterizing Roles in TCP Connection Graphs

We use publicly available real-traffic traces "*The CAIDA Anonymized 2009 Internet Traces*" [1] and analyze a consecutive one hour period considering only TCP traffic in port 80. We obtain the roles of each IP address by inspecting the TCP SYN packets.

Empirical Degree Distribution. Figure 1 shows the degree distribution of vertices according to their roles (in log-log plot): client, server and client-server. We first observe that all distributions are heavy tailed and appear to follow a power-law, with the server vertices having a heavier tail (smaller slope) than the client vertices. More importantly, there are significant differences in the tail of these distributions. In particular, 0.19% of the servers and 0.09% of client-servers have a degree that is larger than the largest client degree, which is 11411. Moreover, the largest server degree (132900) is three times larger than the largest client/server vertex and eleven times the largest client vertex.

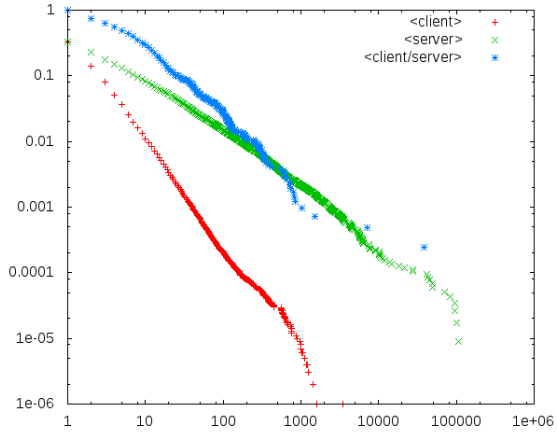

Figure 1: Empirical CCDF distribution by role.

Edges The edges of the network provide a good characterization of the adjacency relationship between the different roles. In particular, the TCP connection network considered has 2,520,009 C-S edges (94.24%), 124,717 CS-C edges (4.66%), 26,932 CS-S edges (1.01%) and 2,396 CS-CS edges (0.09%). Note that by far, most edges are between a client and a server, while far behind are edges between a client and a client-server. Another interesting statistic is the relative frequency of neighbor roles conditioned on the role. For example, given that a node is a server, 98.94% of its neighbors are clients while only 1.06% of its neighbors are client-servers.

Empirical Joint Degree Distribution. The joint degree distribution of edges conditioned on the roles provides more detailed information on the adjacency relationships. This distribution, omitted due space limitations, revels a tendency of low degree client nodes to be neighbors of large degree server nodes.

3. Identification of Roles

The observations concerning structural properties of roles in the TCP connection network led to development of three different techniques to identify the roles of vertices. In particular, these techniques use only the simple connection structure of whom "talks" to whom and provides a consistent identification for the nodes (i.e., all adjacency relationships are possible).

3.1 Decreasing Degree Classifier

This technique explores the differences between the tails of the empirical degree distribution of the different roles. According to our characterization, nodes with large degree are likely to be servers. This technique classifies the vertices in decreasing order of their degree. Whenever possible, a node is labeled server (for example, the highest degree will always be a server). If this is not possible, due to a consistency violation (e.g., neighbor of a node is already labeled server), the node is labeled client. Otherwise the node is labeled client-server.

3.2 BFS Classifier

This technique explores the fact that most edges are between a server and a client and the fact that the largest degree of the network is likely to be a server. The technique is based on a BFS (Breadth First Search) and propagates labels according to levels of nodes in the spanning tree induced by the BFS. In particular, the algorithm attempts to alternatively identify nodes as servers and clients, starting with the largest degree node labeled server. Note that every time an odd cycle is reached by the BFS, one node must be classified as client-server, as otherwise we would have a consistency violation with respect to possible adjacent relationships. The algorithm labels the node that closes the odd cycle as client-server.

3.3 Multiple-BFS Classifier

This technique is a generalization and improvement of the BFS classifier. In particular, an identification error in the BFS classifier can generate a cascade of identification errors down the BFS spanning tree rooted at the wrongly identified node. Despite the large percentage of C-S edges, a small identification error in the beginning of BFS spanning tree can generate large errors in the overall identification.

The multiple-BFS technique simply uses the k largest vertices of the network as starting points for the BFS in the attempt to classify them all as servers. The goal is to reduce the spread of an early identification error. Furthermore, this technique also allows for identification of C-S vertices out of odd cycles. Note that a single BFS is started, but the k largest vertices are added into the queue of nodes to be explored by the BFS. The identification process follows as usual, avoiding consistency violations.

4. Results

In this section we present preliminary numerical evaluation of the three proposed techniques to identify roles in the TCP connection network. We measure the precision, recall and F-measure of each of the techniques when applied to the same real-traffic trace as characterized in Section 2. Due to space limitations, we only report on the F-measure (the harmonic mean between precision and recall).

Figure 2 presents the F-measure for each of the different roles and for each of the techniques (larger is better). All techniques are better at identifying client roles, which is expected, given the considerably larger number of client roles in the TCP connection graph. We also note that BFS does a poor job in classifying client/server roles, due to the ambiguity of odd cycles. Finally, the multiple-BFS classifier, using $k = 10$ (that is, 10 largest degree nodes are considered for server role), shows the best result in all categories.

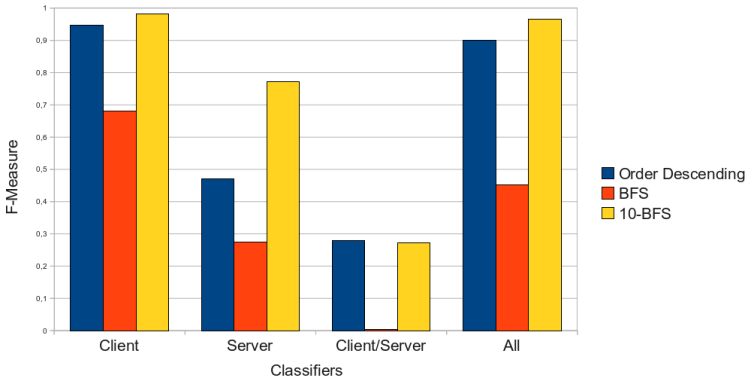

Figure 2: F-measure (efficiency) for different classifiers.

References

[1] E. Aben, kc claffy, D. Andersen, and C. Walsworth. The CAIDA anonymized 2009 internet traces.

[2] M. Iliofotou, P. Pappu, M. Faloutsos, M. Mitzenmacher, S. Singh, and G. Varghese. Network monitoring using traffic dispersion graphs (TDGs). In *ACM Internet Measurement Conference*, 2007.

[3] R. Pang, M. Allman, V. Paxson, and J. Lee. The devil and packet trace anonymization. *SIGCOMM Comp. Commun. Rev.*, 36(1), 2006.