ELECTRONICS AND ELECTRICAL ENGINEERING
ISSN 1392 – 1215 *2010. No. 5(101)*
ELEKTRONIKA IR ELEKTROTECHNIKA

T 180 *TELECOMMUNICATIONS ENGINEERING*
TELEKOMUNIKACIJŲ INŽINERIJA

Graph Mining for Traffic Source Similarity Search

M. Ekmanis
Faculty of Electronics and Telecommunication, Riga Technical University,
Azenes str. 12-137, LV–1048, Riga, Latvia, phone: +371 29119856, e–mail: martins.ekmanis@rtu.lv

Introduction

Data network consists of a huge amount of different traffic sources that make difficulties to find abnormal and unwanted traffic sources between them. Available signature based detectors are able to identify only well known threads and to apply predefined actions to prevent them. Advanced experience based systems, for instance case-based reasoning, depends on similar case retrieval. Practical experience has demonstrated that a simple feature vectors are not adequate to represent the complexity of cases encountered in practice.

The objective of this paper is to review graph based similarity method for describing individual host role in the network, based on their relationships with other hosts. The method relays only on the fact that there is communication between pair of hosts, without taking into account unreliable information about payload, port numbers, protocols, etc. We will analyze single and multi sensor traffic flow capture aspects.

This work is focused on the TCP/IP network protocols.

Connections graph

There exist three categories of case representation [1]: feature value representations, sequences and strings, and structural representations. The first and the second category were observed in my previous publications [2], [3].

The structural representation itself can be divided in three categories: hierarchical structure, network structure, and flow structure. The most straightforward structure representing network communications is the network structure.

In the field of similarity findings for networks and graphs there exist several well known methods, like structure mapping engine (SME) [4], graph edit distance, largest common subgraph. All these methods are computationally expensive, thus they are not applicable in a huge structures like communication graphs in real networks. Still graph theory is successfully applied to solve different tasks in telecommunication networks. Dense bipartite graph are used for spam filtering [5]. Virus propagation and its epidemical threshold also are obtained from the graph theory [6]. Well known graph is global routing topology of the Internet.

In general the graph $G=(V,E)$ consists of a set of vertices V and a set of edges E. An edge e_{ij} represents the connection between vertex i and j (unique IP addresses).

The network is dynamic environment, where we need to fix some time windows to measure similarity. The time window can be fixed or sliding. There are two general methods how to construct graph from network traffic [7] - the first is Edge on First Packet (EFP) filter that characterizes protocol independency. The second is Edge on First SYN Packet (EFPS) that is more accurate, but is applicable only for TCP flows, therefore not usable in more general cases like this.

The source of data in my research is netflow data [8] obtained from routers. As it follows from TCP/IP model, the routers operate at internetwork level and the communication inside broadcast domain does not traverse them (Fig. 1) – thus any local communication F5, F6 are not captured.

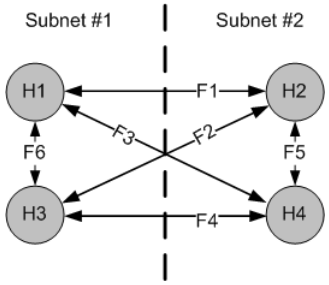

Fig. 1. Subnet boundary and graph limitations

This limitation reflects in the local clustering coefficient C_i (2). In the case of one router, $C_i = 0$ for any vertex v_j in the graph, because C_j is given as proportion of links between the vertices within its neighborhood divided by the number of links that could possibly exist between them. The neighborhood is in different subnet, thus we don't see any connections

between them. The neighborhood $N(v)$ for vertex v is defined as its immediately connected neighbors (1). The degree $k(v)$ of vertex is defined as the number of vertices in its neighborhood

$$N(v) = \{v : e_{ij} \in E \wedge e_{ji} \in E\}, \quad (1)$$

$$C_i = \frac{|\{e_{jk}\}|}{k_i(k_i - 1)} : v_j, v_k \in N_i, e_{jk} \in E. \quad (2)$$

The increase of sensor number changes situation slightly as it is possible to capture some of connections in the neighborhood.

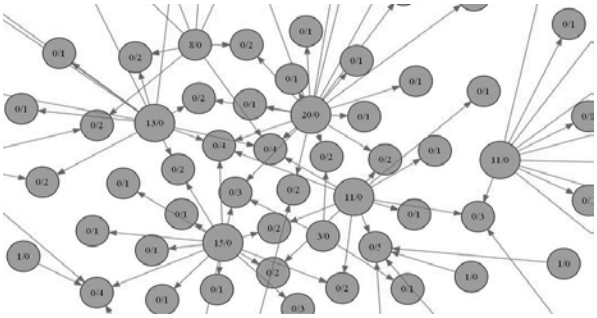

Fig. 2. Chunk of constructed graph from netflow data. Labels show number of incoming and outgoing edges

The graph was growing fast the first 15 minutes (Fig. 3) in closed network within limited range of hosts (total 1500 unique IP devices). Most of active hosts send at least one packet in this time. Number of edges is increasing as there are a lot of possible combinations between any two vertices.

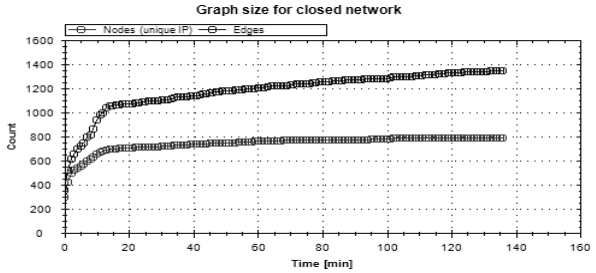

Fig. 3. Graph size for closed network (Upper line is edges)

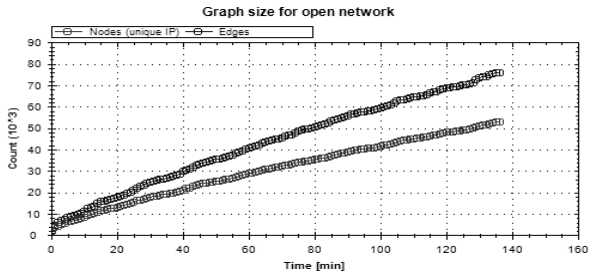

Fig. 4. Graph size for open network (Upper line is edges)

In the case of open networks – connected to internet (Fig. 4) graph size keep growing linearly due to much larger address space and "Background radiation" [9] (flooding, scans for vulnerabilities, worms, misconfigurations, etc.). This graph refers to Large Sparse Graph. The optimal similarity search algorithm must be chosen for handling a large amount of vertices and to obtain any useful data with limited computing resources in limited time.

Host similarity

Every host is described (case description language) by its place in the graph, thus a context similarity measure can by applied. Simple and intuitive method for the similarity measure is a SimRank [10]. This method is based on hypothesis "similar objects are related to similar objects".

I propose the hypothesis that network connections also represent relations of similar systems running the same applications, the same protocols, controlled by the same servers or bonnets, etc.

The metric system must satisfy four conditions/ axioms: nonnegativity (3), identity (4), symmetry (5), and triangle inequality (6):

$$d(a,b) \geq 0, \quad (3)$$

$$d(a,b) = 0 \text{ if } a = b, \quad (4)$$

$$d(a,b) = d(b,a), \quad (5)$$

$$d(a,c) \leq d(a,b) + d(b,c). \quad (6)$$

It is still possible to build classifier that use measure, that is not the proper metric system. Therefore I will call this measure as similarity, as it does not satisfy triangle inequality.

For a node v the set of in-neighbors are $I(v)$, where individual neighbor are denoted as $I_j(v)$, for $1 \leq i \leq |I(v)|$. Thus similarity between objects a and b are $s(a,b) \in [0,1]$. If $a = b$, then $s(a,b) = 1$ otherwise (7), where C is constant between 0 and 1

$$s(a,b) = \frac{C}{|I(a)||I(b)|} \sum_{i=1}^{|I(a)|} \sum_{j=1}^{|I(b)|} s(I_i(a), I_j(b)). \quad (7)$$

In cases when objects a or b do not have any in-neighbors, there is no way to calculate similarity, thus $s(a,b) = 0$.

As it seen in formula (7), there must be used recursive and multiple iterations k to converge to limits satisfying the equation (8)

$$a,b \in V, \lim_{k \to \infty} s_k(a,b) = s(a,b). \quad (8)$$

The first iteration starts with diagonal matrix $s_0(a,b)$ (9). Initially other cells are set 0 as there is the only known fact – the object is similar to itself

$$s_0(a,b) = \begin{cases} 1, if & a = b, \\ 0, if & a \neq b. \end{cases} \quad (9)$$

For estimating the impact of constant C and the optimal number of iterations we apply the mean squared error.

I use the mean squared error between the current *(k)* and the previous *(k-1)* distance matrix $MSE(s_k, s_{k-1})$ (10) to estimate these parameters (Fig. 5).

$$MSE_k = \frac{1}{n^2}\sum_{i=0}^{n}\sum_{j=0}^{n}(s_k(i,j) - s_{k-1}(i,j))^2 \ . \qquad (10)$$

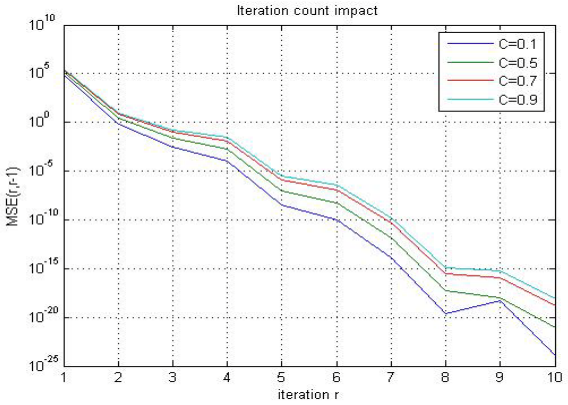

Fig. 5. Iteration count and C impact on SimRank

No significant changes were observed changing constant C values. The differences between lines are explained by the direct impact of coefficient on the result. The appropriate number of iteration very depends on data. In the most cases it is from 2 to 35 giving the MSE between iterations less than 10^{-5}.

Experimental part

Netflow v5 operates on unidirectional information [8], returning states of active connections each 60sec. For the graph construction we need bidirectional information and for that we need to do some additional steps:

1) Collect netflow records from all sensors (40 routers are exercised simultaneously in the experiment);
2) Search and align reverse traffic records;
3) Multisensory data alignment (more than one sensor can see the same traffic);
4) Reconstruct complete flow in time dimension;
5) Determine the flow direction (packets timestamp, protocol flags, and port numbers);
6) Filter background noise (single packets, unidirectional data, and corrupted connection by capture windows boundaries);
7) Construct the graph by the edge on the first flow filter;
8) Calculate the distance matrix;
9) Extract pairs having significant similarity;
10) Sort list and represent results.

An evaluation of similarity methods rely on external measure of similarity. Using real network data, there are no reliable external data to compare with. The results obtained by other similarity methods [2], [3] underscore different relations in data. For the results reported in this paper, I use domain-specific properties for manually verified results. Top similar host pairs are extracted and sorted according similarity (Table 1). These pairs are formed from natural clusters or subgroups (Fig. 6), where hosts are similar between each other in the group and have low similarity between the groups. Most or the groups contain sequential IP address ranges, representing pools of huge internet sites, time server pools, DNS servers and other services with common functions.

Table 1. Top similar host pairs in sample trace

Host pairs	Common features
192.168.111.129 192.168.111.193	Found as a internal server pool
192.168.111.129 192.168.111.65	
192.168.111.129 192.168.111.1	
192.168.111.129 192.168.110.1	
192.168.111.65 192.168.111.193	
192.168.111.65 192.168.111.1	
192.168.111.65 192.168.110.1	
192.168.111.1 192.168.111.193	
192.168.111.1 192.168.110.1	
192.168.110.1 192.168.111.193	
62.85.117.118 62.85.117.117	Delfi
62.85.117.118 62.85.117.71	
62.85.117.117 62.85.117.71	
89.111.15.53 89.111.15.52	DEAC hosting
89.111.15.53 89.111.15.48	
89.111.15.53 89.111.15.19	
89.111.15.52 89.111.15.48	
89.111.15.52 89.111.15.19	
89.111.15.48 89.111.15.19	
213.175.75.59 213.175.75.57	Social network frype.lt, draugiem.lv
213.175.75.59 213.175.75.56	
213.175.75.59 213.175.75.55	
213.175.75.59 213.175.75.54	
213.175.75.59 213.175.75.52	
213.175.75.58 213.175.75.59	
213.175.75.58 213.175.75.57	
213.175.75.58 213.175.75.56	
213.175.75.58 213.175.75.55	
213.175.75.58 213.175.75.54	
213.175.75.57 213.175.75.55	
….	

The source code is written in Microsoft Visual Studio C#, using memory hash tables for the fast indexing using IP address pair hash. GraphViz graph visualization component [11] is utilized for graph drawings (Fig. 2). Graphics (Fig. 3, Fig. 4, and Fig. 6) are rendered by ZedGraph library [12]. The complete source code are available here http://ekmanis.id.lv/g-src.zip

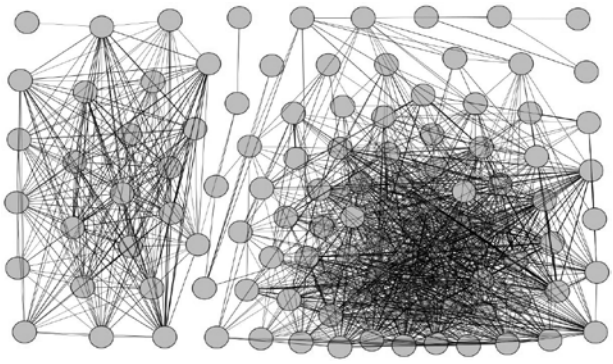

Fig. 6. Similarity graph – host grouping tendency

Conclusions

This paper presents an attempt to apply graph based similarity technique to data network environment. Part of important information is hidden in communication

topology and is unreachable using feature vector methods, sequences or patterns. This information comes from upper layers – applications, end users and their human experience and behavior. Analogous mechanisms are successfully used in web search, e-shops and banner systems for calculating similarity between web resources, products, query strings etc., based on user feedback and hyperlink formed object to detect relationships.

The gain of the concept implementation shows the ability of the algorithm to detect similarity between hosts even if we have no evidence of direct communication between these hosts. In the most cases when we have information about the only gateway router the data will be transformed as bipartite graph.

The calculated similarity does not cover all information, thus is neither perfect nor complete. It must be used in the composition with other similarity measures to improve classification results.

References

1. **Cunningham P.** A taxonomy of similarity mechanisms for case–based reasoning // IEEE Transactions on Knowledge and Data Engineering. – 2009. – No. 21(11). – P. 1532–1543.
2. **Ekmanis M., Novikovs V., Rusko A.** Unauthorized Network Services Detection by Flow Analysis // Electronics and Electrical Engineering. – Kaunas: Technologija, 2008. – No. 5(85). – P. 53–56.
3. **Ekmanis M.** Genetic Programming Based Network Traffic–Profiling System // Electronics and Electrical Engineering. – Kaunas: Technologija, 2009. – No. 4(92). – P. 49–52.
4. **Gentner D.** Structure–Mapping: A Theoretical Framework for Analogy // Journal Cognitive Science. – 1983. – No. 2, Vol. 7. – P. 155–170.
5. **Desikan P., Srivastava J.** Analyzing Network Traffic to Detect E–Mail Spamming Machines // Proc. ICDM Workshop on Privacy and Security Aspects of Data Mining. – 2004. – P. 67–76.
6. **Chakrabarti D., Wang Y., Wang C., Leskovec J., Faloutsos C.** Epidemic Thresholds in Real Networks // ACM Transactions on Information and System Security (TISSEC). – ACM Press, 2008. – Vol. 10, iss. 4. – P. 1–26.
7. **Iliofotou M., Pappu P., Faloutsos M., Mitzenmacher M., Singh S., Varghese G.** Network Traffic Analysis using Traffic Dispersion Graphs (TDGs). // Proceedings of the 7th ACM SIGCOMM conference on Internet measurement. – ACM Press, 2007. – P. 315–320.
8. **Cisco IOS NetFlow**. http://www.cisco.com/en/US/products/ps6601/products_ios_protocol_group_home.html.
9. **Pang R., Yegneswaran V., Barford P., Paxson V., Peterson L.** Characteristics of Internet Background Radiation // Proceedings of the 4th ACM SIGCOMM conference on Internet measurement. – ACM Press, 2004. – P. 27–40.
10. **Jeh G., Widom J.** SimRank: a Measure of structural–context similarity // KDD'02 Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining. – ACM Press, 2002. – P. 538–543.
11. Graphviz – Graph Visualization Software. Available at: http://www.graphviz.org.
12. ZedGraph. Available at: http://zedgraph.org.

Received 2010 02 15

M. Ekmanis. Graph Mining for Traffic Source Similarity Search // Electronics and Electrical Engineering. – Kaunas: Technologija, 2010. – No. 5(101). – P. 39–42.

This work represents a method of describing individual host role in the network, based on their relationships with other hosts. The method relays only on a fact that there is communication between pair of hosts, without taking into account unreliable information about payload, port numbers, protocols, etc. Single and multi sensor traffic flow captures aspects are analyzed. The gain of the concept implementation shows the ability of algorithm to group similar traffic sources together. Ill. 6, bibl. 12, tabl. 1 (in English; abstracts in English, Russian and Lithuanian).

М. Екманис. Анализ графов для поиска подобных источников трафика // Электроника и электротехника. – Каунас: Технология, 2010. – № 5(101). – С. 39–42.

Анализируется отличительный метод описания роли источников сетевого трафика в сети, основанный на их взаимодействии с другими источниками. Метод основывается только на факте, что осуществляется коммуникация между источниками, без учета недостоверной информации о полезной нагрузки, номерах портов, протоколов и т.д. Проводится анализ одного и нескольких датчиков захвата потока трафика. Доказательство реализации концепции показывает способность алгоритма группировать аналогичные источники трафика вместе. Ил. 6, библ. 12. табл. 1. (на английском языке; рефераты на английском, русском и литовском яз.).

M. Ekmanis. Srautų šaltinių paieška analizuojant grafus // Elektronika ir elektrotechnika. – Kaunas: Technologija, 2010. – Nr. 5(101). – P. 39–42.

Pateikiamas originalus metodas tinklo srautų šaltiniams įvertinti. Pasiūlytas metodas išsamiai įvertina tinklo šaltinių komutacinius režimus, kai nepakanka informacijos apie apkrovą, prievadus, protokolus ir t. t.. Analizuojamos vieno ir kelių jutiklių srautų pažeidimų galimybės. Gauti rezultatai realizuoti algoritmų, kurie puikiai grupuoja srautų analogiškus šaltinius, pavidalu. Il. 6, bibl. 12, lent. 1 (anglų kalba; santraukos anglų, rusų ir lietuvių k.).