Detecting Anomalies in Inter-hosts Communication Graph

Jan, 14, 2009

Keisuke ISHIBASHI*, Tsuyoshi KONDOH*, Shigeaki HARADA§, Tatsuya MORI§, Ryoichi KAWAHARA§, Shoichiro ASANO¶

*NTT Information Platform Labs.

§NTT Service Integration Labs.

¶National Information Institute

Outline

- Anomalous traffic detection
- Inter-host communication graph
- Anomalies in communication graph
- Detecting method for graph anomaly
  - Similarities between graphs
- Experimental results
  - Synthesized traffic
  - Actual traffic

Anomalous traffic detection

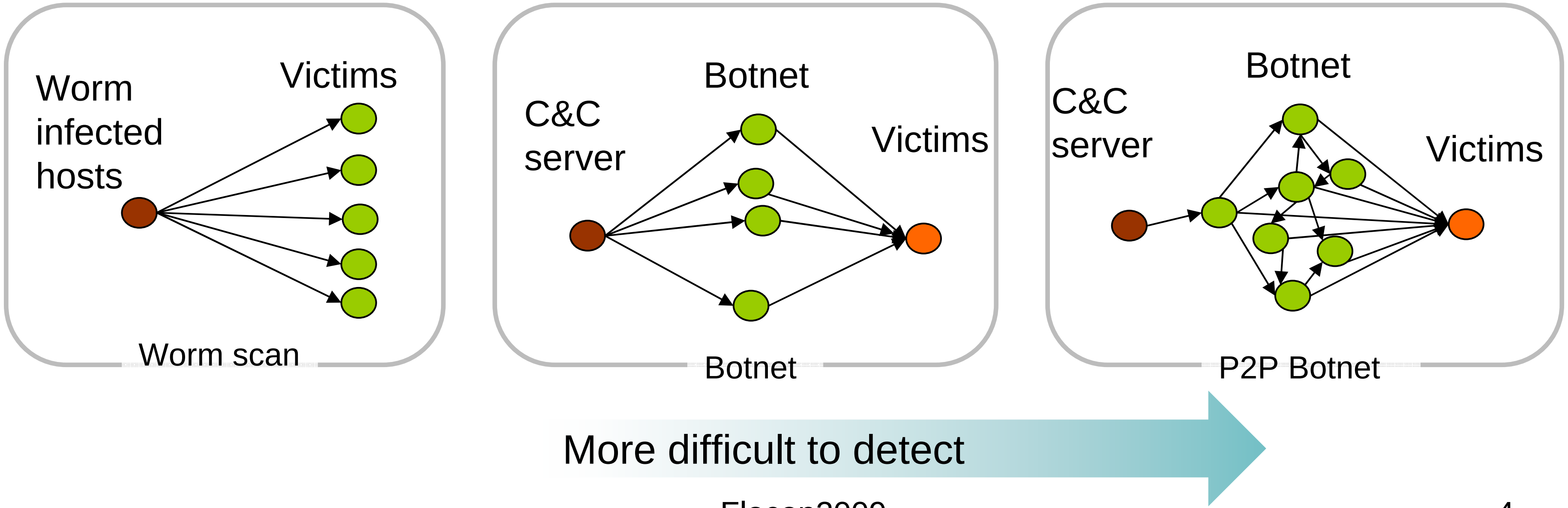
- DDoS attacks, Network failure etc: can be detected as sudden change in traffic volume
- Worm scans or botnet C&C traffic: cannot be found as volume change
  - Whose traffic volume is very small, and buried in normal traffic

⇩

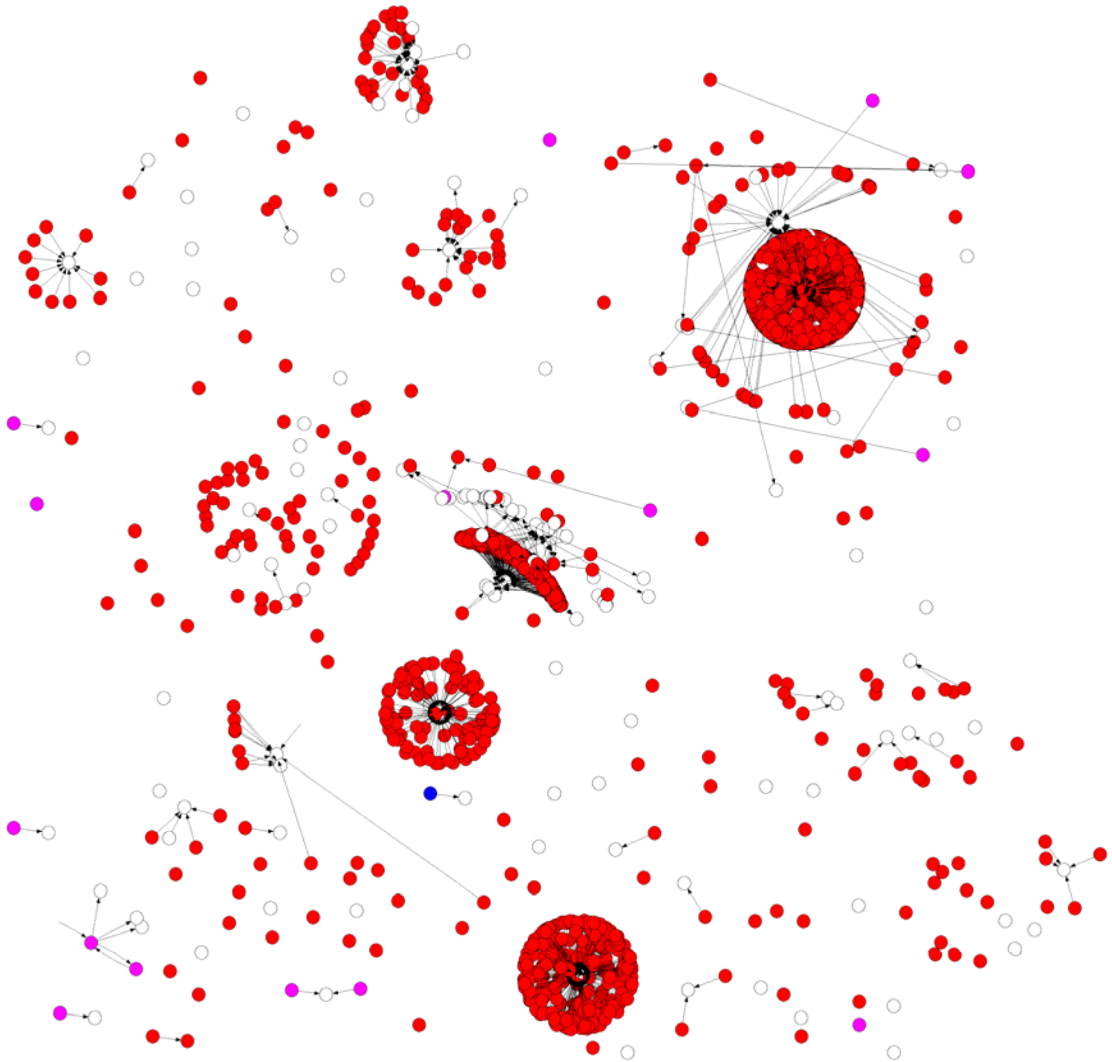
- May be found as sudden change in traffic pattern, not volume
- Traffic pattern
  - Entropy: can reveal traffic characteristic per hosts.
  - Communication pattern between hosts: can reveal anomalous traffic which appears as inter-hosts communication pattern

Communication pattern between hosts

- Can be represented as graph
- Communication graphs for anomalous traffic
  - Some of them are difficult to detect with conventional methods
    - Conventional methods: monitoring entropies in number of flows, etc

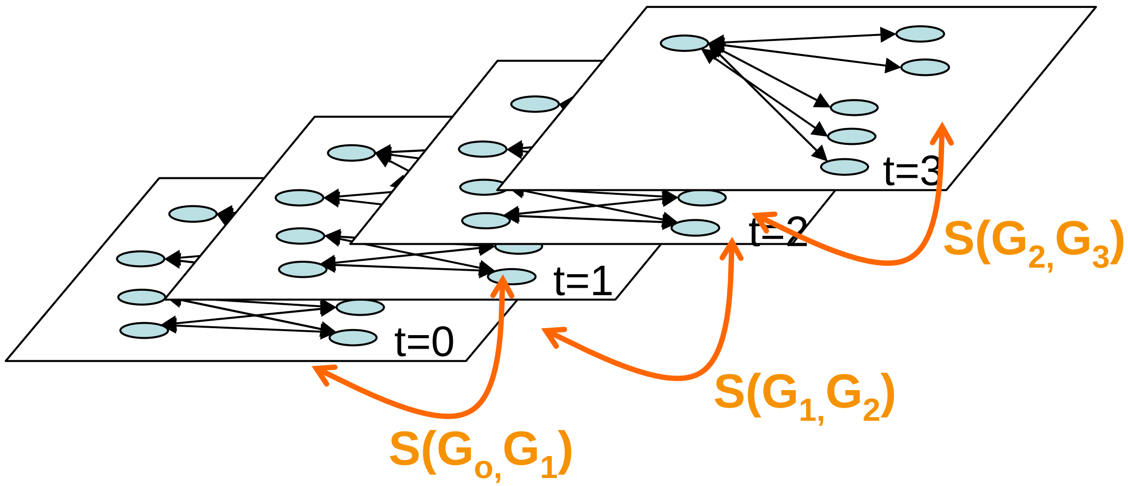


Time series of communication graph



Challenge

- How to detect anomaly (change) in time series of graph?
- Visualization or animation of commutation graph[Yurcik06]
  - Useful especially for digging anomalous event by hand
  - However, eyeballing by human operator is needed to detect anomalous event
- Automated detection: need to define similarity between graphs $S(G_t,G_{t+1})$, where G_t and G_{t+1} are graphs of time t and t+1
  - Can judge as an anomaly if $S(G_t,G_{t+1})$ suddenly decreases



- [Yurcik06] William Yurcik, “VisFlowConnect-IP: A Link-Based Visualization of NetFlows for Security Monitoring,” 18th Annual FIRST Conference, June 2006.

Similarities between graphs

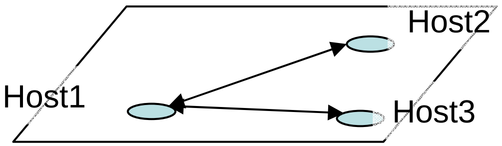
- Graph Kernel
  - Define “inner product” like function f(•, •), a.k.a kernel, on the space of non-linear spaces [Kashima03]
- Edit distance
  - Number of operations to change graph G to G’ [Bunke06]
  - operations: add/remove edges/nodes

- Can be used to detect anomalies in graph time-series
- Difficult to identify the source of anomaly

- [Kashima03] H. Kashima, et.al , “Marginalized kernels between labeled graphs,” In Proc. ICML 2003, pp.321-328.
- [Bunke06] H. Bunke et.al, “Computer Network Monitoring and Abnormal Event Detection Using Graph Matching and Multidimensional Scaling, ” LNCS Vol. 4065 2006.

Linear feature space projection

- Linear feature space projection[Ide04]
  – Mapping a graph to a vector in the linear space that represents the feature of the graph
- As feature vectors, adopt a principal eigenvector of adjacency matrix for the graph
  – ≈Page Rank vector
  – Dimension of linear space: Number of nodes in graphs

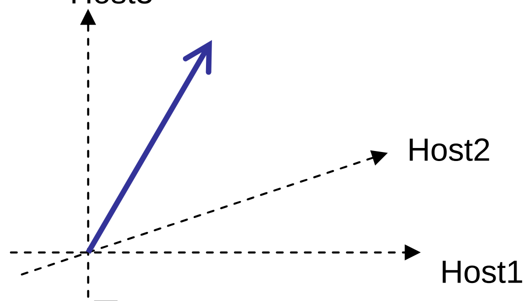


Communication graph

	1	2	3
1	-	1	1
2	1	-	0
3	1	0	-

Adjacency matrix

Principal eigen vector



Feature vector

• [Ide04] Tsuyoshi Ide and Hisashi Kashima: Eigenspace-based Anomaly Detection in Computer Systems, In Proc. 10th ACM SIGKDD Conference (KDD2004), Seattle, WA, USA, 2004.

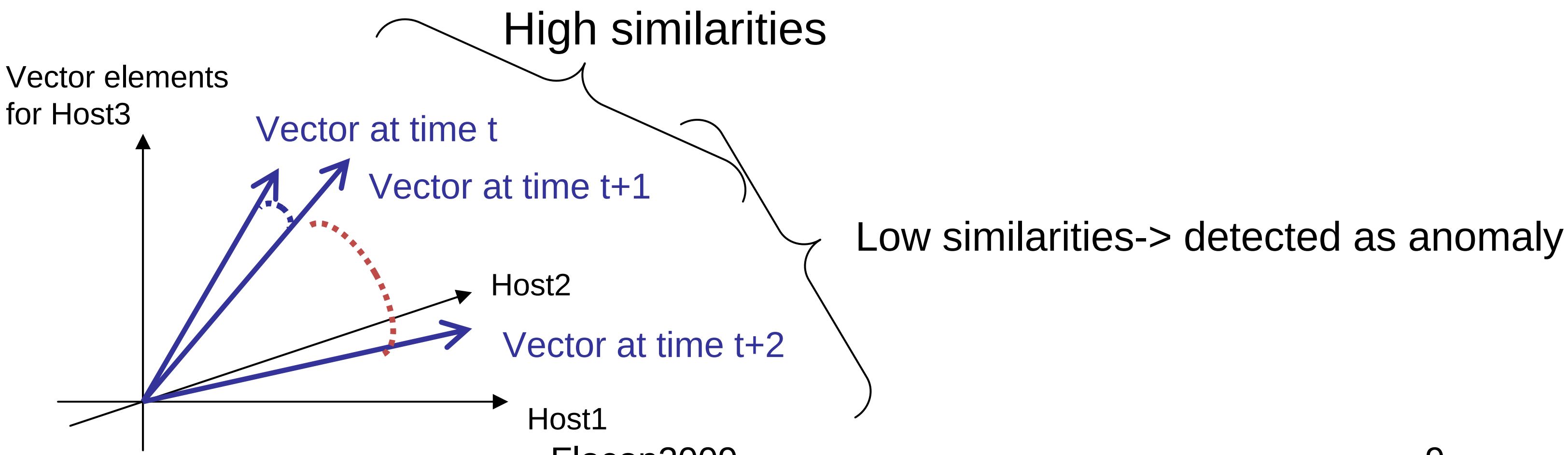
Anomaly detection using feature vector

- Periodically generate communication graph from observed traffic data, and calculate feature vectors of the graphs
- Calculate similarity between the graph and the previous one

$$S(G_t, G_{t+1}) := \frac{V_{G_t} \cdot V_{G_{t+1}}}{|V_{G_t}||V_{G_{t+1}}|}$$

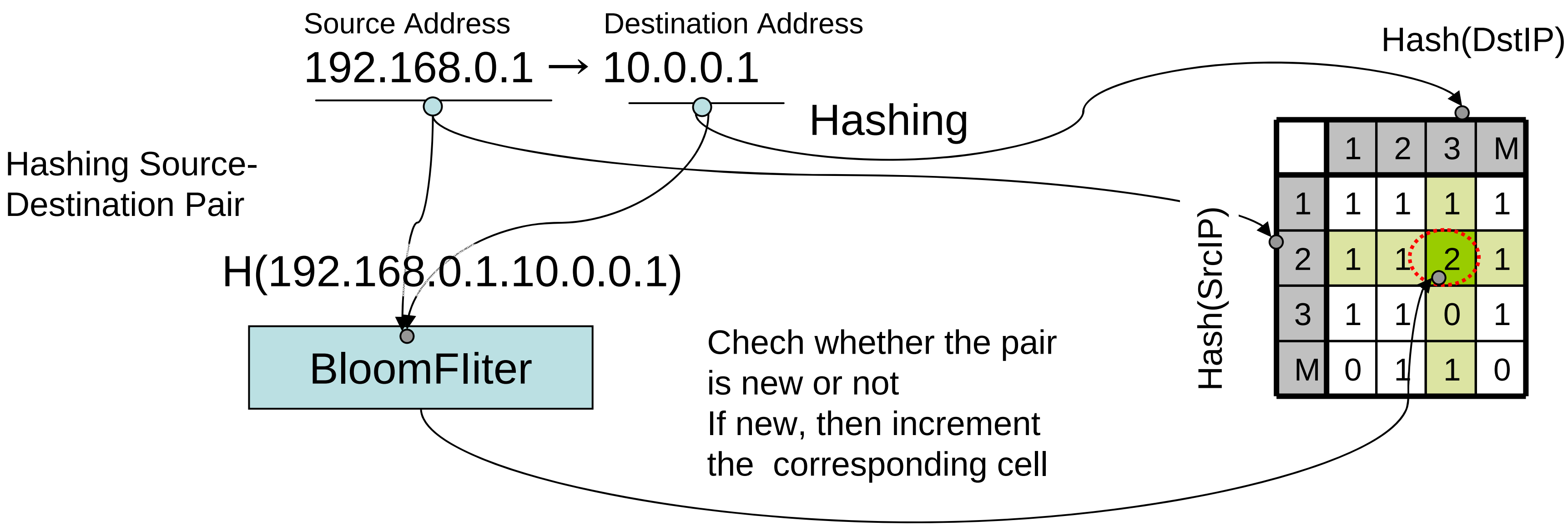
Cosine similarity

- Judge as anomaly if the similarity suddenly decreases



Compressing adjacency matrix

- In large communication graph, calculating principal eigen vector of adjacency matrix may be difficult.
- Compress adjacency matrix by combining hash matrix and bloom filter



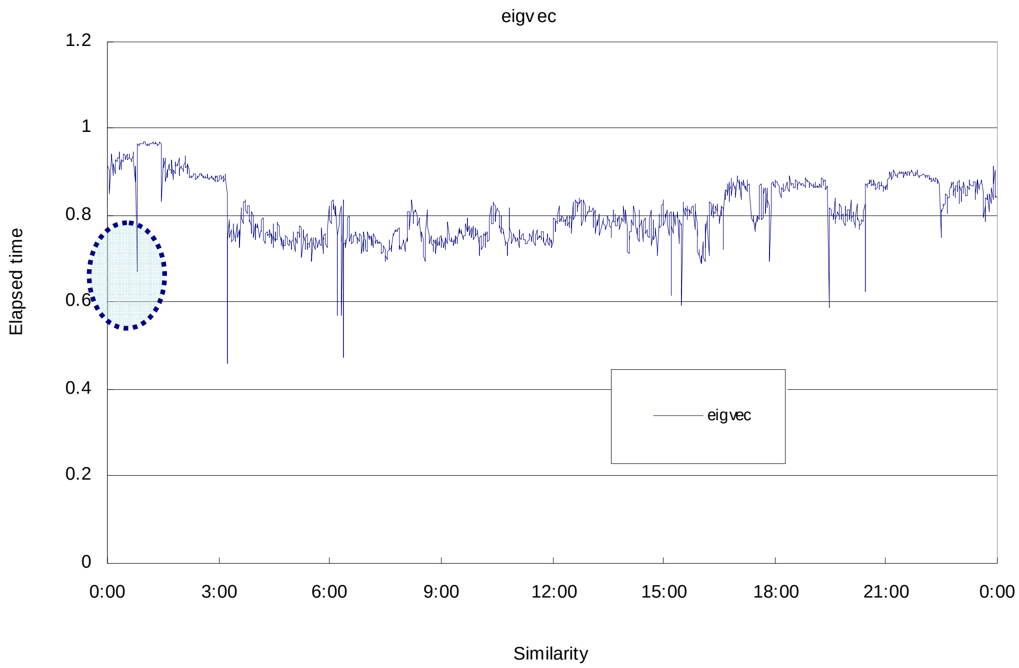
	1	2	3	M
1	1	1	1	1
2	1	1	2	1
3	1	1	0	1
M	0	1	1	0

Experimental results

- Observed data: packet capture data of 24-hour long at 1Gbps link
- Use packets with ports 135/445(scans)/6667(IRC)
  - Current python implementation cannot handle whole traffic
  - Focus on botnet related traffic
- Generate graphs every minutes
- Hash matrix size：1280×1280

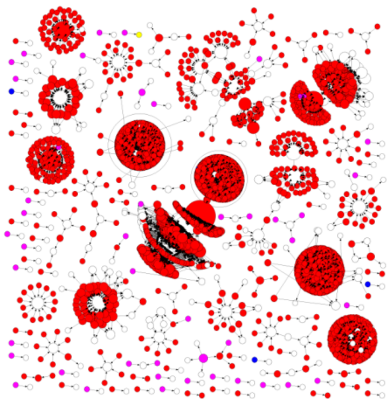
Time series of simulates of feature vectors

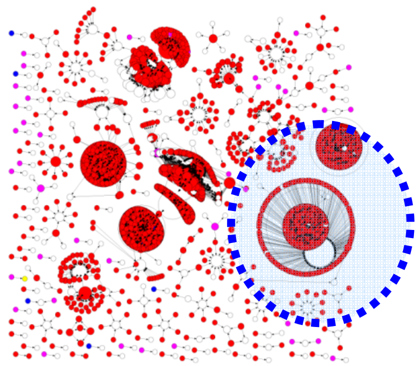
- Several sudden decreases in similarities
- Try to find the source of anomaly for the first one

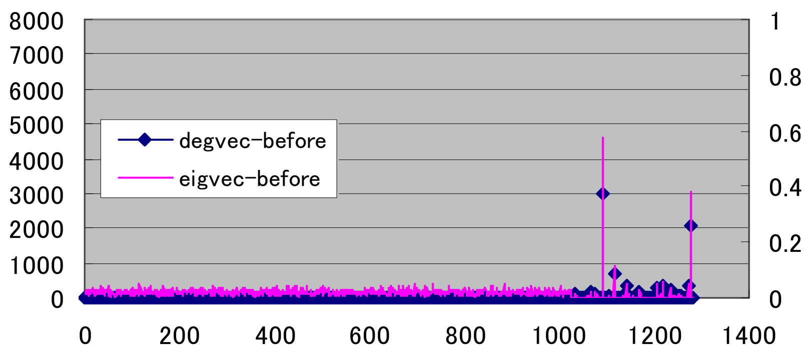


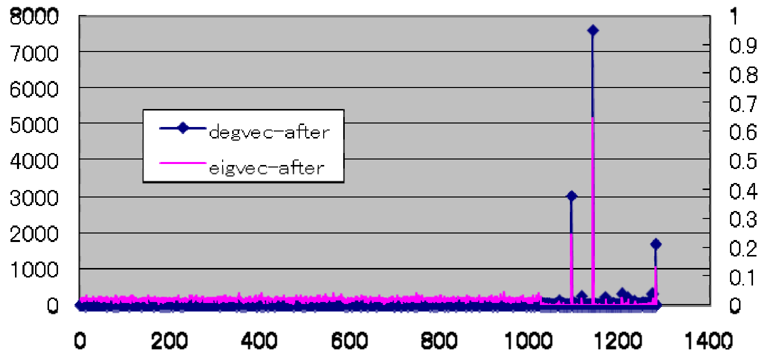
Comparison of graphs before/after the anomaly

- By comparing graphs and/or vectors before/after the anomaly, we can identify the source of anomaly
  - Comparing vectors is fit for automated identification
- In this case: sudden large virus scan



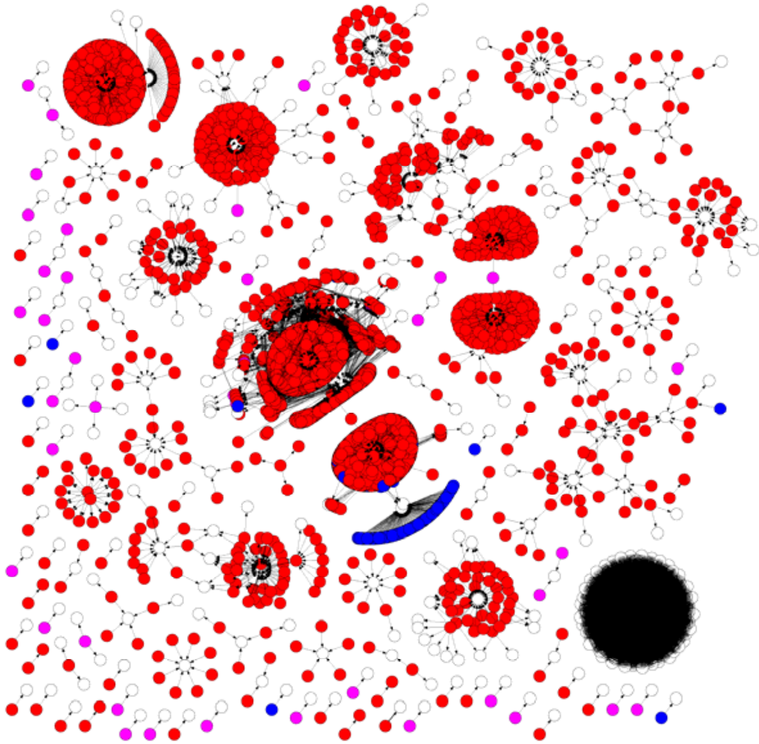






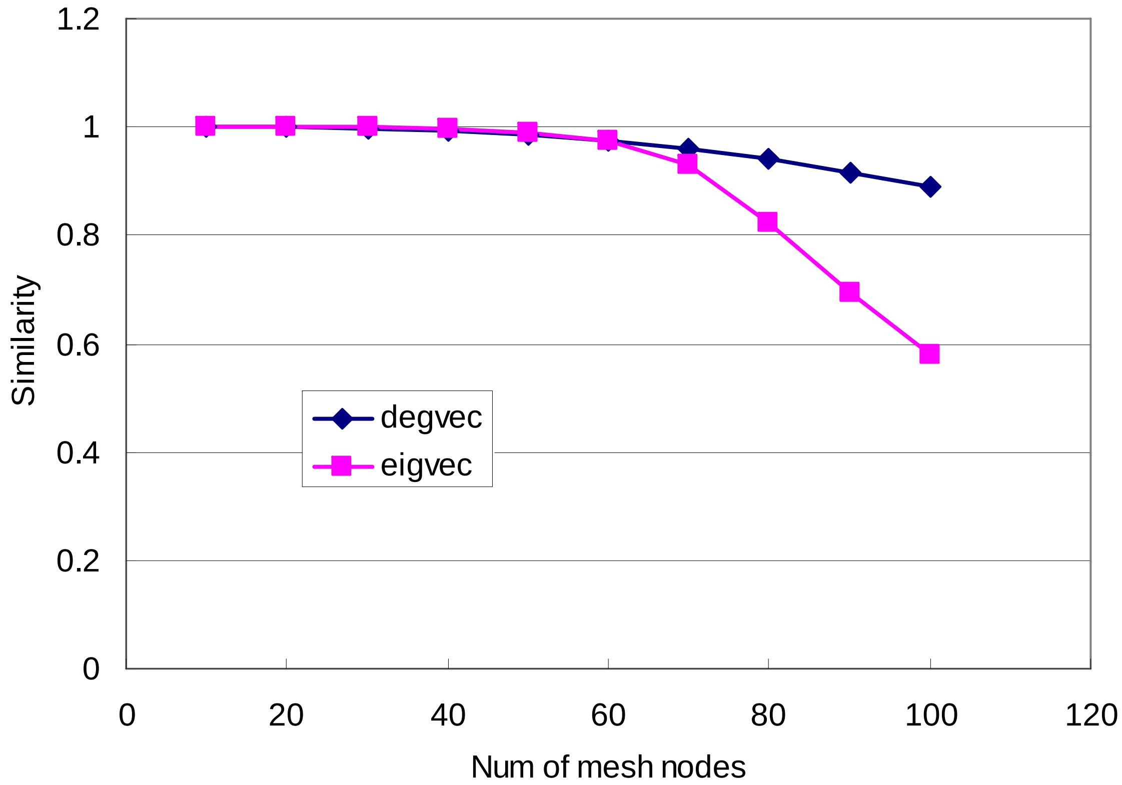
Evaluation with synthesized anomaly cluster

- Which type of anomaly and how large anomaly can be detected by the proposed method?
- Evaluation using synthesized anomaly can answer the above question
- Firstly, mesh cluster of various size is inserted to actual communication graph and calculate the similarity between the original graph



Evaluation with synthesized anomaly cluster

- With mesh size > 70, similarity decreases and the anomaly can be found



Conclusion

- Summary
  - Propose a method to detect anomalies in communication graphs
    - Projection of graph into linear feature spaces, and compare the simulates between feature vectors
  - Evaluate using actual traffic data
    - Found a sudden large worm scan
- Future works
  - Apply to other traffic data to find out which type of anomaly the proposed method can detect
  - Faster implementation

Acknowledgement

- This study was supported in part by the Ministry of Internal Affairs and Communications of Japan.