Proceedings of the 5th WSEAS Int. Conf. on MULTIMEDIA, INTERNET AND VIDEO TECHNOLOGIES, Corfu, Greece, August 17-19, 2005 (pp101-107)

Emphasizing Anomalous Events in Computer Networks for Improved Security

John C. McEachen
Department of Electrical and Computer Engineering
Naval Postgraduate School
Monterey, California

John M. Zachary
Innovative Emergency Management, Inc.
Baton Rouge, Louisiana

Abstract: - This paper describes an effort to provide a holistic view of network conversation exchanges for the purpose of real-time network monitoring and anomaly detection. We argue that monitoring and anomaly detection are necessary mechanisms for ensuring secure and dependable network computing infrastructure. The model for network traffic exchange is based on a modified Ehrenfest urn model and combines statistical physics and queuing theory to provide macrostate descriptions of complex networked systems when the exact microstate parameters of each element in the system precludes global understanding from first principles such as throughput and utilization. The conversation exchange dynamics model for real-time network monitoring and anomaly detection is formally presented in this context as a system-driven data reduction model. The model induces a unique real-time visualization capability for network monitoring and detection of anomalous events. This aids in identifying violations of network policy such as network attacks and misconfigurations. This approach has been verified in several environments. Example responses from network attacks simulated in the laboratory including those contained in the DARPA Lincoln Lab IDS test data as well as from operational network traffic are presented. These results suggest that our approach presents a unique perspective on anomalies in computer network traffic.

Key-Words: Intrusion detection, network diagnostics, statistical mechanics.

1 Introduction

As distributed network intrusion detection systems expand to integrate hundreds and possibly thousands of sensors, managing and presenting the associated sensor data becomes an increasingly complex task. Methods of intelligent data reduction are needed to make sense of the wide dimensional variations. We present a new approach for monitoring network behavior we call conversation exchange dynamics (CED) that accentuates anomalies in traffic flow. This approach provides an aggregated primitive that may be viewed in its own right or used by intrusion detection systems to base detection strategies upon.

Understanding network behavior for the purposes of diagnosis and intrusion detection is currently a major effort in the quest to build secure, robust and dependable computing systems. Specifically, intrusion detection systems (IDS) are detection security mechanisms that monitor a computer system or network, attempt to detect malicious activity, and raise an alarm to system or security administrators. IDSs can be classified as either anomaly-based detection or signature-based detection [1]. The former approach detects *anomalous* behavior, which may be a superset of *undesirable* behavior, and generally suffers from high false alarm rates. The latter signature-based approach may reduce false alarm rates but generally depends on a well-defined security policy to base detection on. Furthermore, signature-based intrusion detection systems are unable to detect events for which a signature is not defined in their signature database.

We present a novel approach to modeling distributed system with a high number of interacting entities. This problem is notoriously complex, and our model seeks to provide some level of data reduction so as to distinguish what is an anomaly from what is typical network activity. Consequently, this technique has the potential to find applicability to a wide variety of systems (beyond computer network systems). We do not address the issue of automated diction itself in this paper. We only present a original method for characterizing network exchanges based on microstate information. We argue that this level of characterization provides a more robust signal that traditional first order measurements such as throughput and utilization.

This paper presents a model for real-time network monitoring and anomaly detection that provides a holistic view of network conversation exchanges. We argue that monitoring and anomaly detection are necessary mechanisms for ensuring secure and dependable network computing infrastructure. The model for network traffic exchange is based on a modified Ehrenfest urn model and combines statistical physics and queuing theory to provide macrostate descriptions of complex networked systems when the exact microstate parameters of each element in the system precludes global understanding from first principles such as throughput, utilization, packet size and packet counts. The conversation exchange dynamics model for real-time network monitoring and anomaly detection is formally presented in this context as a system-driven data reduction model. The model induces a unique real-time visualization capability for network monitoring and detection of anomalous events. An implementation of the model and visualization capability is presented along with laboratory tests and successful detection of computer network attacks, including a Code Red worm attack.

This paper expands upon the work found in [2]. Efforts related to our approach can be found in [3] and [4]. These approaches all consider the problem from the abstraction of determining statistical properties of the network. In this

paper, however, we intend to focus on the analysis of the underlying state descriptors with the hope of extending these to global properties in the future.

2 Describing Network Conversation Flow

As was stated before, the goal of our approach is to reduce standard network data into a useful, reproducible, and meaningful form, ultimately to allow accurate detection of network anomalies. The notion of state will be more carefully defined below, but quickly summarizes into activity levels of various conversation groups within the network. One end product is a real time graphical description of the configuration of the network.

The network is constructed as a state space of information sources and sinks. As information quanta move throughout a network, the state space is updated accordingly.

States are represented as a vector of sources and sinks. The analogy used is that of *buckets* and *balls*. Information moves between nodes represented as buckets as indivisible balls. As the network moves information around, this is represented as balls being passed between buckets. For example, a series of *n* packets transmitted from a node N_a to another node N_b would be modeled as *n* balls moved from bucket *X* and placed in bucket *Y*. The association of node N_a with either bucket *X* or *Y* depends on the nature of the conversation. The bucket can be defined using any combination of conversation characteristics including the affiliation of who is talking (individual hosts or networks), the language they are speaking (TCP, UDP, or ICMP), or the job they are performing (client or server).

In its simplest form, each node in a network is associated with one or more buckets and the total number of packets exchanged between nodes is modeled as moving balls from bucket to bucket. The collection of all buckets together with the allowable distribution range of balls forms a *bucket state space*.

2.1 Bucket Space Definition

A *bucket* b_i is a scalar representing an ordinal number of balls for conversation entity *i*. As noted, a bucket represents any combination of conversation characteristics, including who is talking (hosts or internal nodes), the language they are speaking (TCP, UDP, or ICMP), or the job they are performing (client or server). A *bucket state space* of *M* conversation entities is represented as an *M*-dimensional vector space $\vec{b} \equiv (b_1,\ldots,b_M)$. The state vector is a discrete time dependent variable, $\vec{b}_t$; the initial configuration is $\vec{b}_0 \equiv (b_{1,0},\ldots,b_{M,0})$. The number of possible bucket states, *N*, can be determined as:

$$N = \begin{pmatrix} M+K-1 \\ M-1 \end{pmatrix} \quad (1)$$

where *K* is the total number of balls in the system. As an example of the breadth of the bucket space, in our laboratory experiments below, $M = 4$ and $K = 24$, which allows for *N* = 118,755 distinct bucket states.

A state transition causes a shift in the distribution of information between the buckets. In other words, this model translates network behavior into bucket state transitions by selecting a ball from the bucket matching the source characteristics of the packet (b_i) and moving that ball into the bucket matching the destination characteristics of the same packet (b_j), thereby redistributing the information and transitioning the state. In a single packet case, this is modeled by changing the bucket state vector $\vec{b}_t \equiv (b_1,\ldots,b_i,\ldots,b_j,\ldots,b_M)$ to $\vec{b}_t \equiv (b_1,\ldots,b_i-1,\ldots,b_j+1,\ldots,b_M)$. The number of balls that change buckets in a given time period depends on the number of conversations and the network rate. In general, the net difference in balls between two buckets and the number of buckets for which the net ball count changes is more than one.

Figures 1 and 2 show the importance of the state walk, and how the model provides more information than a simpler model. Both graphs represent a two bucket state space made up of two conversation groups, $\vec{b} \equiv (b_A, b_B)$. The initial state vector is $\vec{b}_0 = (5,5)$. A state transition occurs if a ball is moved from b_A to b_B or vice versa; a removed from b_A and placed back into b_A results in stasis for that time period. Contrast the two bucket and 10 possible states shown in the simple example of figures 1 and 2 with the more realistic bucket state expansion described above.

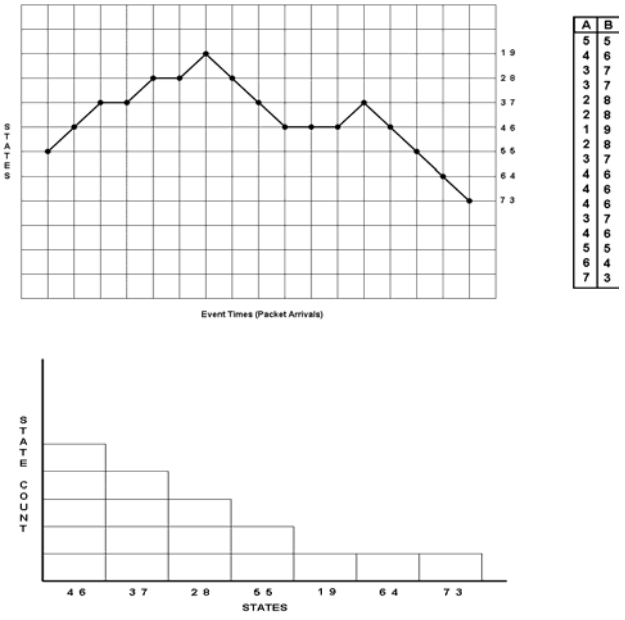

Fig 1. A sample state walk for a two bucket model. (Top left) A plot of the state walk over time. (Top right) A plot of the visited bucket states, b_i, over the course of the state walk. (Bottom) A ranked histogram of the bucket states, b_i, over the entire time period of this state walk. Note that not all states have been visited and thus are not included.

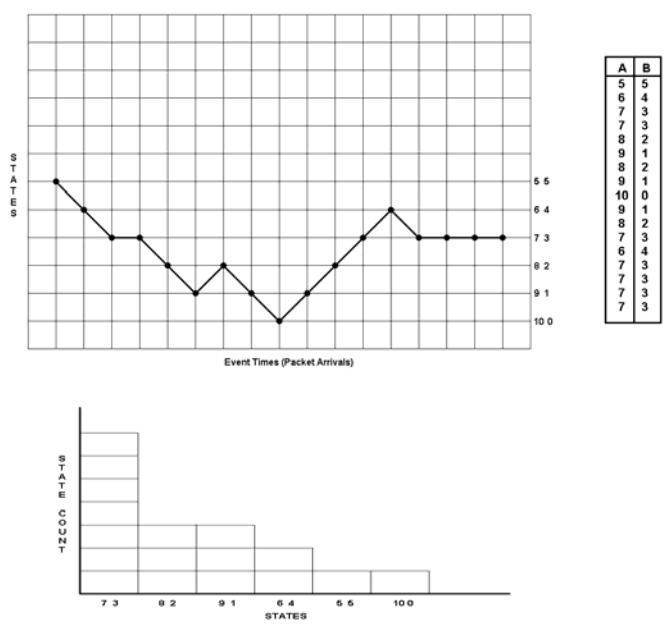

Fig 2. An alternative state walk for a same two bucket model as shown in fig 1. (Top left) A plot of the state walk over time. (Top right) A plot of the bucket sizes over the course of the state walk. (Bottom) A ranked histogram of the bucket states over the entire time period of this state walk. Note how the histogram varies for this state walk even though both examples end in the same state.

2.2 Thermal Manifolds and Anomaly Detection

By examining the manifold, or canyon, developed by collecting various state histograms over time, $\vec{b}_t$, anomalies can be easily spotted as perturbations in the normal flow of the canyon. The cause of such dramatic changes in practice ranges from a single transition to many thousands of packets.

The sequence of *k* state space vectors $\vec{b}_t,\ldots,\vec{b}_{t+k}$ forms a *random state space walk*. The variable *k* defines a time window because the system is a discrete random system. The random walks can be reduced into another usable form by ranking states based on state counts, p_i, for a fixed time period. Define $C_{t,t+k}(\vec{b}_i)$ as the number of instances that $\vec{b}_i$ appears in the random walk $\vec{b}_t,\ldots,\vec{b}_{t+k}$. This function defines a histogram $\vec{h} = C_{t,t+k}(\vec{b}_t),\ldots,C_{t,t+k}(\vec{b}_{t+k})$. Rank order *h* by a permutation $\Pi(h) = p_0,\ldots,p_k$ such that $p_0 \geq p_1 \geq \ldots \geq p_k > 0$.

This permutation describes the frequencies that each state was visited in the random walk from most frequent to least frequent. This is analogous to the thermodynamic concept of Boltzmann curves of probability versus the energy state of molecules in a gas. Under the Boltzmann analogy, the area under the bucket state curve is a representation proportional to the temperature. Similarly, the slope of the state curve is a representation proportional to the entropy. Macroscopic energy fluctuations are proportionally represented by the changes between time slices, corresponding to endothermic (heat absorption) or exothermic (heat release) reactions.

The state counting method for constructing the thermal manifold described above can be extended to construct a probability density function for the set of states $\{\vec{b}_i\}$. The probability that a state $\vec{b}_i$ occurs in given time period is

$$P_{\vec{b}_i} = \frac{\text{time in state } \vec{b}_i}{\text{total time period}}. \tag{2}$$

We define the *occurrence rate* of state $\vec{b}_i$ as

$$q_{\vec{b}_i} = \frac{C_{0,t}(\vec{b}_i)}{\text{time in state } \vec{b}_i}. \tag{3}$$

Given a constant time measurement period τ, equation (2) is rewritten as

$$P_{\vec{b}_i} = \frac{1}{\tau}\frac{C_{0,t}(\vec{b}_i)}{q_{\vec{b}_i}} \tag{4}$$

to facilitate subsequent derivations.

The probability density function in equation (4) is used in a straightforward manner to calculate *entropy* as

$$H = \sum_{i=1}^{M_t} -P_{\vec{b}_i} \ln P_{\vec{b}_i}\ . \tag{5}$$

The *temperature* of the network system is derived by means of a transformation function Φ comparing the state progression of the current measurement period with the state progression of the previous measurement period. A recurrence relationship is established between the previous system temperature and the new temperature by this transformation:

$$\begin{aligned} T_0 &= 1 \\ T_t &= \Phi T_{t-1} \end{aligned} \tag{6}$$

The state progressions (m_t and m_{t-1}) that define Φ must be sufficiently comparable. This can be accomplished through the initial conditions, the boundary conditions, and an averaging mechanism. Given comparable state progressions, the following relationship holds:

$$\prod_{i=1}^{m_{t-1}} C_{0,t-1}(\vec{b}_i) = \left(\prod_{i=1}^{m_t} C_{0,t}(\vec{b}_i)\right)^{\Phi^{-1}} \tag{7}$$

Solving for Φ,

$$\Phi = \frac{\log \prod_{i=1}^{m_t} C_{0,t}(\vec{b}_i)}{\log \prod_{i=1}^{m_{t-1}} C_{0,t-1}(\vec{b}_i)} \tag{8}$$

In typical operations, m_t is on the order of 10^3 and $C()$ is on the order of 10^2. This results in a product growth that is difficult to handle in real-time with floating point arithmetic. A simplification of the logarithm of a product is used:

$$\log\left(\prod_{i=1}^{m_t} C_{0,t}(\vec{b}_i)\right) = \sum_{i=1}^{m_t} \log\ C_{0,t}(\vec{b}_i) = M. \tag{9}$$

Substituting into equation 8, the temperature at time *t* is calculated as:

$$\begin{aligned} T_0 &= 1 \\ M_0 &= 1 \\ T_t &= \left(\frac{M_t}{M_{t-1}}\right) T_{t-1} \end{aligned} \tag{10}$$

Other thermoinformative properties can be derived directly from the probability density function, such as energy [2]. The effectiveness of these thermal properties as discriminators for anomaly detection is currently under investigation. An example of their use is provided in the next section, however, the thermal manifolds are the current primary discriminator for this purpose.

An anomaly can cause one of two effects in the above mentioned figures. Either new states are visited, or previously visited states are seen more often. The first effect will cause a spike oriented along the "STATES" axis and the latter along the "STATE COUNT" axis. An anomaly that is orthogonal to the normal traffic flow will tend to cause a spike oriented along the states axis, due to the new states visited. An anomaly that is parallel to the normal traffic flow will tend to cause a spike oriented along the state counts axis, due to the revisiting of previously visited states. The magnitude of the potential spike is what determines the ability of the operator to detect the anomaly. The orientation of the anomaly with respect to the normal traffic flow will determine the magnitude of the perturbation. A single packet anomaly that is orthogonal to the normal traffic flow will cause a large perturbation in the graph (as will be shown later in this paper), where a packet that is parallel to the normal traffic flow will cause a relatively small perturbation. The less orthogonal the anomaly is to the normal traffic flow, the larger the number of anomalous packets required to cause a noticeable perturbation in the graph.

For example, figure 3(a) represents all of the possible bucket states that are contained in the bucket state space for a system consisting of three buckets (a, b, and c) each containing four balls. Each of the blue nodes represents a different bucket state. The number of balls in a given bucket is given by the lines parallel to the side opposite the vertex of interest. Each of the vertices corresponds to the case where all of the balls are in the associated bucket. The purple node represents the initial ball distribution, or initial bucket state of $\{4,4,4\}$. In figure 3(b), the number of balls in bucket 'c' is constant at four. The thick green line represents the nine possible bucket states based on a conversation between the remaining two buckets.

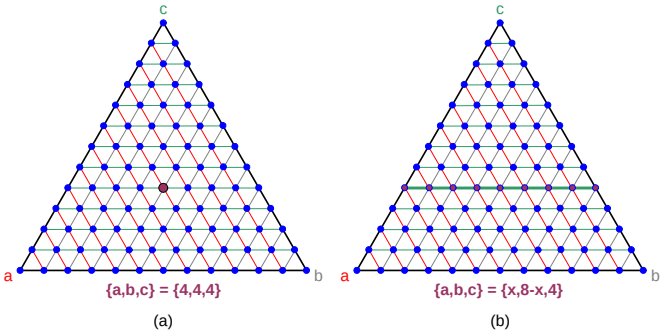

Fig. 3. Graphics that depict the total bucket state space for a system containing three buckets each with four balls. Each node corresponds to a different bucket state. – (a) The purple node corresponds to the bucket state of $\{4,4,4\}$. (b) The green line represents the range of possible bucket states for a conversation between buckets 'a' and 'b'.

An example of the results of a single packet anomaly, that is orthogonal to the normal traffic flow, can be seen in figure 4. In this case the packet caused a ball to move from bucket 'c' into the conversation between buckets 'a' and 'b'. The result is a new line of possible bucket states. This new line contains ten possible bucket states. Assuming we average data from any given sample window over subsequent sample windows, there are now nineteen possible bucket states, which is more than double the original number of nine. This results in a run out (in the z-axis direction) of the Thermal Canyon graph. This type of anomaly is very easy to detect even though it was caused by a single packet.

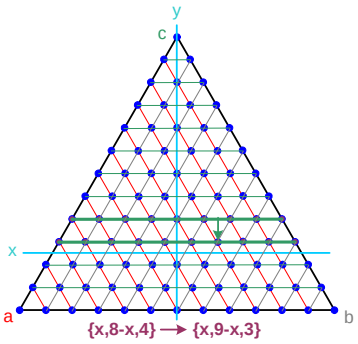

Fig. 4. A graphic depicting the results of an anomalous packet that is orthogonal to the normal traffic flow. The anomalous packet causes a ball to move from bucket 'c' into the conversation between buckets 'a' and 'b'. The results is the state walk moves from the line at $c=4$ to the line at $c=3$.

Given that the manifold displays the average ball count per bucket per sample time period, it is less sensitive to anomalies that represent only a small percentage of the traffic. The average bucket size per sample window shows significant changes in the traffic flow. Therefore, if an anomaly is to be noticed in the average bucket size, it must comprise an appreciable percentage of the total traffic in the affected buckets. For example, a few packets that are parallel to the normal traffic flow of 500 packets per second (pps) will not be seen, but if the packets are orthogonal to the normal traffic flow, regardless of traffic rate, they will be seen.

The more orthogonal anomalous traffic is to the normal traffic flow, the greater effect the anomaly will have on the system graphs. Since it is not possible to know all the expected anomalous traffic in advance, the key is to create a bucket space that provides tight classification of critical traffic. For example, traffic should be parsed by functional group, like web servers, as opposed to grouping servers and clients together. There is a limit to the number of buckets a configuration can have, ideally, multiple instances of the system should be run concurrently to allow for smaller bucket spaces. This is also beneficial in reducing the complexity of interpreting the graphs which increases with the number of buckets. The next section presents some of the analysis on actual and lab-generated network traffic.

3 Experimentation and Analysis

This section is divided into two parts: controlled experiments conducted laboratory test equipment and results from real traffic on operational networks.

3.1 Laboratory experiments

The purpose of these experiments was to show how the bucket state histogram varies as the bucket space description deviates from the actual network configuration. Two categories of experiments are discussed. The first shows the effect of an increasing number of rogue web servers on the bucket space histogram. The second shows the effect of a single out-of-profile packet on the bucket space histogram.

The simulation network consisted of a trusted subnet of 10 web servers connected to an untrusted internet of 1000 clients. In order to simulate typical network traffic, Spirent TeraMetrics traffic generators running TeraCaw software were used. This system is capable of simulating millions of client/server sessions using various application level protocols, beyond the initial three-way handshake. The system was configured such that any where between 400 and 800 users (using the 1000 client machines) would randomly access the 10 web servers. Each web access consisted of establishing a TCP connection with the server, an HTTP GET message and then a 64-byte HTTP Response message. The connection was then terminated with a RESET.

The bucket space definition included a list of "authorized" web servers, identified an address space associated with trusted users, and considered ports below 1024 to be service ports. Consequently, for figures 5 through 9 the bucket space was partitioned as follows:

1. A trusted IP address, a web server, and a service port
2. A trusted IP address, a web server, and not a service port
3. A trusted IP address, not a web server, and a service port
4. A trusted IP address, not a web server, and not a service port
5. Not a trusted IP address and a service port
6. Not a trusted IP address and not a service port

Figures 5 – 13 are expansions of figures 1 and 2 in three dimensions with the added dimension being time. In other words, figures 1 and 2 represent just one time sample. If we were to make time x-axis (as shown in figures 5 and 6), then we would have the type of manifold display depicted in figures 5 – 13. In these figures, the y-axis represents frequency of state occurrences and the z-axis represents the state identifier. The tail of displayed in the z-axis is indicative of the number of states visited.

Specifically, figure 5 illustrates the average bucket sizes and bucket space histograms over a period of two minutes (120 seconds) for our laboratory experiment. The typical load on the network was approximately 1000 packets per second.

Since the bucket space definition is aligned with the actual traffic patterns, the number of non-zero bucket states is small. Hence the histogram tail is very short.

Perhaps more importantly, this display also illustrates the smoothing effect defined by the central limit theorem on traffic that has been shown to be highly self-similar in nature on a per-client basis ([5], [6]). In other words, even though all clients arguably have the same heavy-tailed exchange characteristics, the actual distribution of states as shown by the bucket state histograms is highly normal and smooth, particularly when the bucket definitions are aligned with the configuration (i.e. all web servers in the web server list).

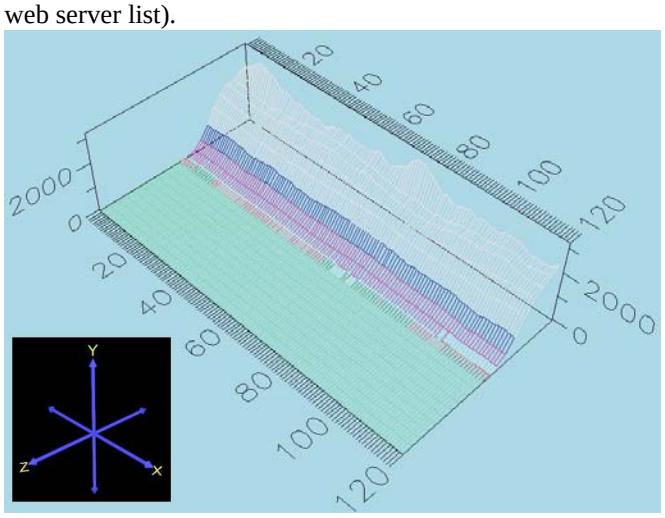

Fig 5.The bucket state histogram over time. Decreasing frequency of bucket states comes out of the graphic. Traffic is confined to external clients visiting internal web servers so the number of bucket states is small.

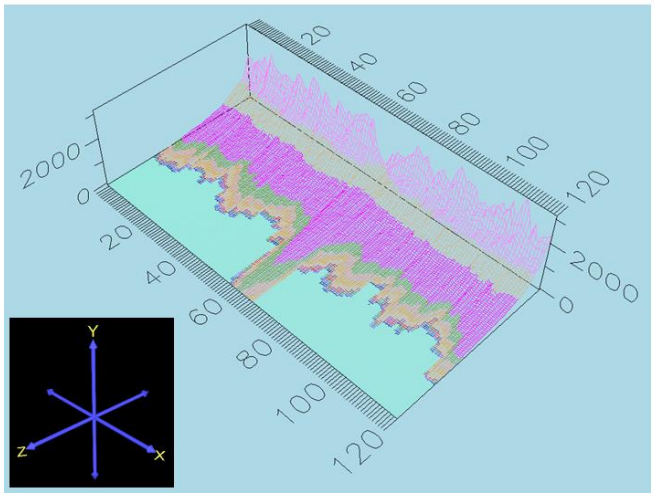

Fig 6. The bucket state histogram with a single UDP injected into the over 200,000 web traffic packets. Compare to figure 5.

Next an anomalous packet was injected into the network for each during the above scenario. The anomalous packet was an UDP packet from one of the web servers to one of the clients. The packet originated from an ephemeral port (1025) with a destination service port on the client (53). Figure 6 shows the effect on the bucket state histogram for this packet a scales comparable to figure 5.

The difference caused by the anomaly of the UDP packet should be readily apparent when comparing figures 5 and 6. Keep in mind that during this two minute period, over 200,000 packets were exchanged. The reason for the significant protrusion is because the UDP packet forces a ball to be transferred to a state that would not otherwise be visited, reducing the counts of the "normal" buckets and altering the frequency of the histograms (hence the notch in the graphics.)

3.2 DARPA Lincoln Lab IDS Test Data

For additional analysis and configuration of the system we considered the *Tcpdump* files that were collected by Lincoln Labs using their 1999 Simulation Network. Per references [7], the simulation network was created to conduct evaluations of intrusion detection systems by measuring detections and false alarm rates. We recognize use the of these files does not come without some controversy [8], however we consulted these packet traces only as a publicly available source of cataloged attacks. Additional, real world attacks are also shown in the subsequent section.

Figure 9 shows the response to attack #41213446, an ICMP flood or "Smurf" attack. This attack is very common and generally easy to detect.

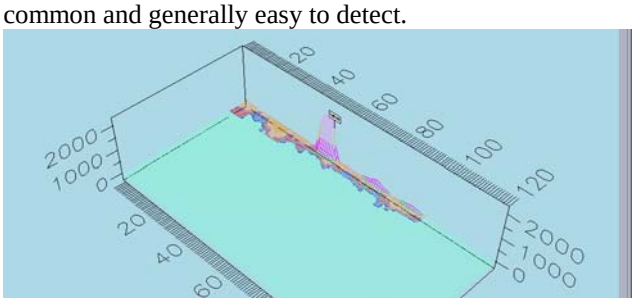

Fig. 9 Response of the system to an ICMP flood, #41213446.

Figure 10 displays the response to a Mailbomb attack, #42155148. This is a denial of service attack directed against the sendmail program. This is accomplished by sending a unique set of strings to the sendmail server.

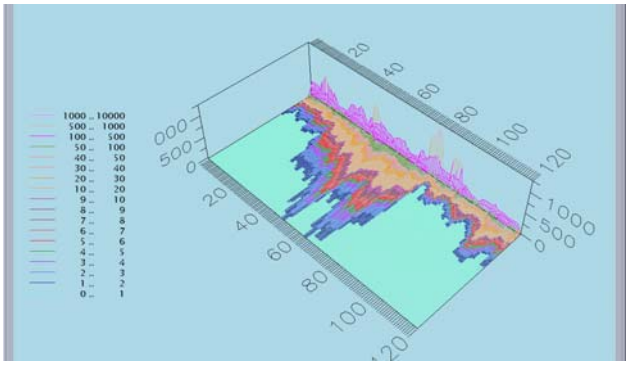

Fig 10. Response of the system to a Mailbomb attack, #42155148.

A similar type of attack to the Mailbomb is the Apache2 attack, #51140100. The response of the system to this attack is shown in Figure 11.

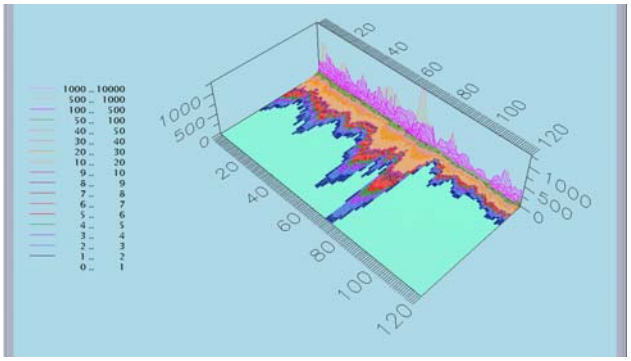

Fig. 11. Response of the system to an Apache2 attack, #51140100.

Finally, the response of the system to a sweep of IP addresses is shown in figure 12. The shape of this response is particularly worth noting.

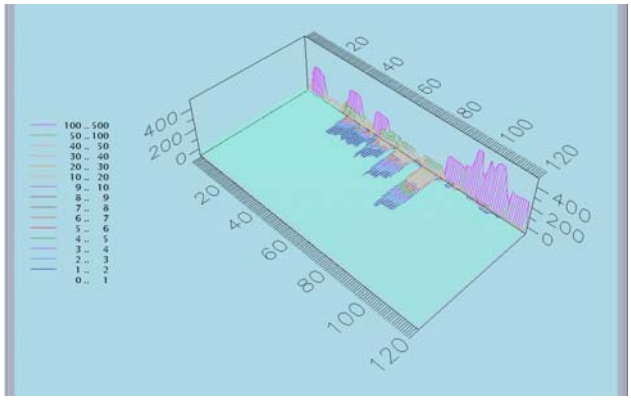

Fig. 12. Response of the system to an IP sweep, #52211313.

3.3 Operational Examples of Thermoinformative Calculations

Figure 14 below illustrates a three-dimensional plot of entropy, temperature and energy collected and computed over a six hour period during a normal work day on an operation network. Figure 15 below is the same plot collected and computed during a six hour Denial-of-Service attack on this operational network. Note how energy remains constant while wide variations in entropy and temperature are observed. As stated above, these usefulness of these results in real-time anomaly detection are still under investigation but are presented to demonstrate their potential.

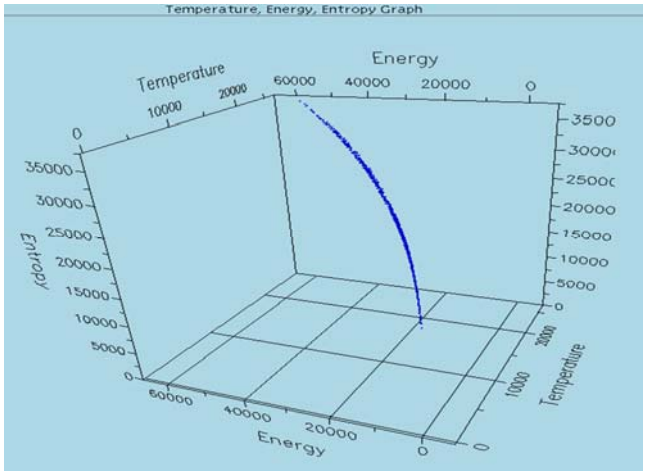

Fig 14. A plot of entropy, temperature and energy over a six hour period under normal operating circumstances.

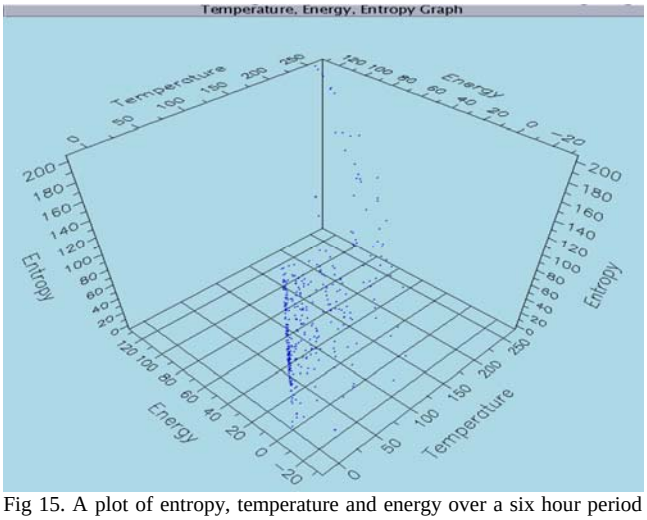

Fig 15. A plot of entropy, temperature and energy over a six hour period during which a denial-of-service attack was occurring on the same network. Note the large variation in entropy and temperature.

4 Conclusion

We have presented a novel approach to characterizing the conversation flow of a computer network. Specifically, a modified Ehrenfest urn model provides a theoretical basis for efficient data reduction and visualization capability in complex, interacting information systems, including anomaly detection in computer networks. The model provides a robust way to describe conversation exchange dynamics between large sets of network nodes. An implementation of this model and demonstrations on experimental and live network traffic illustrate the utility of the model in capturing conversation flow of network traffic and the dynamics of early anomalous events.

Additionally, a general theory of network intrusion detection as an ill-posed problem is unexplored. This development is necessary to handle protocol ambiguities and the unconstrained nature of network protocols exploited for malicious intent. Classification of methods for regularlization and development of new regularization methods are critical for managing the network security and assurance problem**.**

Finally, we did not address the actual problem of detection in this paper – only a novel way of characterizing data and presenting a signal that may be used for detection. The issue of evaluating this signal for automated detection of anomalous events is still unresolved.

References

[1] S. Axelsson, "Intrusion detection systems: A survey and taxonomy", Chalmers University Technical Report 99-15, March 2000.

[2] Donald, S. D., McMillen, R. V., Ford, D. A., and McEachen, J. C., "Modeling Network Conversation Flux for Patternless Intrusion Detection", *Proc. of the 6th WSEAS Int. Conf. on Communications*, pgs. 441 – 446, Crete, July 2002.

[3] Burgess, M., "Thermal, nonequilibrium phase space for networked computers," *The American Physical Society*, Volume G2 Number 2, pgs. 1738-1742, August 2000.

[4] Evans, S. C. and Barnett, B., "Network security through conservation of complexity," *Proceedings of IEEE Military Comms. Conf. (MILCOM 2002)*, pgs. 1133 – 1138, Los Angeles, October 2002.

[5] Crovella, M. and Bestavros, A. "Self-Similarity in world-wide web traffic: Evidence and possible causes," *Proc. ACM Sigmetrics Conf. on Meas. And Mod. Of Comp. Sys.*, May 1996.

[6] Arlitt, M. and Jin, T. "A workload characterization study of the 1998 world cup web site," *IEEE Network*, May/June 2000.

[7] Massachusetts Institute Of Technology, Lincoln Laboratory, DARPA Intrusion Detection Evaluation. [http://www.ll.mit.edu/IST/ideval/index.html]. 14 September 2003.

[8] McHugh, John, "Testing Intrusion Detection Systems: A Critique of the 1998 and 1999 DARPA Intrusion Detection System Evaluations as Performed by Lincoln Laboratory," *ACM Trans. On Info. And Systems Security*, Volume 3, Number 4, pgs. 262 – 294, November 2000.