Detecting Network Events via T-Entropy

Ulrich Speidel, Raimund Eimann, and Nevil Brownlee
Department of Computer Science
The University of Auckland
Private Bag 92019
Auckland, New Zealand
Email: {ulrich,raimund,nevil}@cs.auckland.ac.nz

***Abstract*— The detection of significant events in heterogeneous networks, such as DDoS attacks, presents a challenge, both because of the diversity and unpredictable nature of events, and because the "normal" background traffic often varies quite naturally. Conventional approaches for detection of such events usually involve either monitoring for specific event signatures, or a statistical approach, which usually requires monitoring a large number of statistical features. In recent years, several papers have proposed the use of entropy- and complexity-based measures as a viable alternative. The present paper argues that T-entropy is a suitable measure in this sense and provides some experimental evidence in support.**

I. Introduction

Many large organisations operate complex heterogeneous networks which they connect to the wider Internet via a gateway router. Typical examples of this type of organisation are large companies, universities, public sector authorities, and retail ISPs. Users within these networks often engage in a wide variety of network-related activities, from e-mail and web surfing to the use of VoIP, network games, or other peer-to-peer applications. Moreover, new applications appear; events unrelated to the network generate changes in traffic volume and characteristics, and there are natural changes in traffic patterns as a result of the time of day as well as weekday/weekend and other changes. This spontaneous and changing nature presents a challenge in the detection of events, especially if there is no known "signature" or effect to look out for in advance.

However, it has been observed that both Shannon entropy and Kolmogorov complexity (approximated by means of Lempel-Ziv compression) of network traffic measured at such gateways appear to be relatively stable in "normal" traffic conditions. Several authors have suggested and demonstrated that this can be used for DDoS detection. Among them are Kulkarni, Bush, and Evans [11] and Kulkarni and Bush [18] and Wagner and Plattner [13], who advocate the use of approximated Kolmogorov complexity, and Feinstein, Schnackenberg, Dalupari, and Kindred [12], who use conventional Shannon entropy.

The usual approach in complexity/entropy-based detection is to extract salient features from the traffic observed during a certain window at the gateway or other vantage point. The window may be time- or volume-based. Features extracted may include the ports used, flags set or not set, addresses, trace timestamps, flows, etc. These are then mapped to symbols such as bytes. In complexity-based detection, these symbols are concatenated into a string. This string is then compressed with a universal string compressor such as a Lempel-Ziv algorithm. The algorithm's output length is then used as an estimate for the Kolmogorov complexity C_K. In entropy-based detection, the symbol frequencies are used to estimate symbol probabilities p_i for each symbol, which in turn are used to compute the Shannon entropy $H = -\sum_i p_i \log p_i$.

The present paper proposes the use of Titchener's T-entropy as a substitute for the above. The next section discusses why a substitute may be useful.

II. Complexity and entropy

A reader with a background in information theory may already be familiar with the concept of the Kolmogorov complexity [1] of a string x of symbols as a measure of the string's information content. If one divides the information content by $|x|$, the length of x, one obtains an (average) information rate. Shannon entropy is also an information rate. The underlying approach is thus the same in both cases, save for the fact that the method of computation differs significantly and that the results are estimates in both cases.

From an information-theoretic perspective, "normal" Internet traffic is generally a low-entropy source, despite the challenges it poses in network management. This is because much of it is actually quite predictable: If one sees the first two datagrams of a TCP handshake, chances are one will see the third. If one observes port numbers used, chances are high that the next port number seen will belong to a known service. If one observes addresses, it is unlikely that all addresses will appear even approximately equally often. If one sees a TCP packet directed to port 80 on a particular host, the probability of the payload starting with a "GET" or a "POST" is quite high. The reader will probably find it easy to find further examples. In some attack traffic, in particular in a DDoS scenario, the uniformity of attack packets necessarily implies a further drop in entropy – if all packets are attack packets, then the entropy should ultimately drop to zero. In conclusion, the ideal complexity/entropy measure for network measurement purposes is one that performs well for low and very low entropies.

So how well do the existing measures work in this entropy range? Before we refer to experimental data, it pays to look at the the fundamental theoretical limitations of these tools. Shannon entropy in the form stated above requires the symbols to

1–4244–0983–7/07/$25.00

be *independent*. That is, p_i, the probability of occurrence of a symbol as the next symbol must not depend on the occurrence of previous symbol(s). It is relatively easy to find networking examples where this assertion does not hold: Server responses are vastly more likely after an actual request; if a host transmits from a slower tributary network to a faster busy backbone, it is highly unlikely to observe another transmission from the host as the next datagram after the first one appears on the backbone, etc.

With respect to the Kolmogorov complexity, we need to look at possible approximations other than via the Shannon entropy. In this context, the well-known Lempel-Ziv family of algorithms are commonly used. Broadly speaking, algorithms of this family try to identify as yet unparsed parts of the string as a combination of a previously encountered substring and a "new" extension. LZ77 [3] and LZ78 [4] are the two fundamental and widely known algorithm concepts.

What seems less well-known is that both LZ77 and LZ78 are derivates of an earlier parsing algorithm published by Lempel and Ziv in 1976 [2], designed to compute the complexity of finite strings, the *production complexity*. In simple terms and in most cases, the LZ production complexity of a string x is the number of steps taken by an LZ77 compressor whose search window is large enough to cover the entire string. One may think about LZ production complexity as being an LZ compressor that parses the string but doesn't actually encode any compressed output. The time complexity of this algorithm (which is not a compression algorithm) is $O(|x|^2)$, which renders it impractical as a compressor for all but very short strings.

LZ77 circumvents the $O(|x|^2)$ regimen by restricting the search for known patterns to a small search window of size w, thus rendering the time complexity $O(|x|w)$. This has the obvious consequence that patterns that have occurred more than w symbols earlier are treated as "new" if they occur again. If w is much smaller than $|x|$, a significant number of candidate strings are disregarded. In network traffic measurement terms, this means that repeated patterns in our network data do not lower the complexity if they are spaced too far apart, so the complexity is generally overestimated. This applies in particular to strings where there is plenty of potential for overestimation, i.e., strings with a low entropy.

LZ78 takes a different approach: It restricts its search space from substrings starting at arbitrary positions in the string to substrings starting at previous positions in the string to achieve a linear time behaviour. The obvious trade-off here is that substrings that start between two such positions may go unrecognized if they are encountered again. This trade-off is significant, given that usually only a small subset of positions in the string qualify as start positions for a search. In terms of complexity/entropy, this also results in over-estimation, once again especially of low entropy strings.

By measuring strings from sources with known entropies, one may gain insight into the real performance of such measures. Speidel, Titchener, and Yang [17] used a large number of 100,000 bit strings derived from the bipartition of the logistic map in order to investigate this problem. They found that the simple 1-gram Shannon entropy above overestimates across almost the entire entropy range, and shows no correspondence to the actual entropy in most strings with entropy levels below about 0.5 bits per bit. For larger symbol blocks ("n-grams") this improves, but even for $n = 15$, sensitivity to entropy is lost at around 0.1 bits per bit.

Among the LZ measures, LZ production complexity (as LZ77 with a string-sized window) performs best, however, as we already know this is bought at a computational price which means that it does not scale well at all. LZ77 with a window size of 1024 bits overestimates by almost a factor of 2 at 0.1 bits per bit. The smaller the window sizes become, the worse the overestimation – both in terms of onset and total amount: A window size of 16 produces an estimation quality comparable to the 4-gram Shannon entropy. LZ78 shows a performance slightly worse than a 1024 bit window LZ77.

As has been shown, this performance is sufficient for DDoS and worm detection in many cases. However, a better measure could increase the dynamic range of observations and hence the reliability and sensitivity of detection. Kulkarni, Bush and Evans point this out, as do Wagner and Plattner, who compare several compressors in this respect. The next section discusses another candidate for detection, the T-entropy.

III. T-ENTROPY

The alternative measure proposed here, the Titchener T-entropy, is the gradient of a measure called T-information, which in turn is a "linearized" version of another string parsing measure, the T-complexity. This section only provides a brief introduction into these topics. An in-depth discussion of these measures is beyond the scope of this paper and is readily available from several sources ([19], [17], [7], [8], [9], [10], [14] and references therein).

Consider building a vocabulary from an alphabet as follows:

1) Use the alphabet as the starting vocabulary and set a counter $i = 0$.
2) Increment i, choose an entry (T-prefix) from the current vocabulary and a positive integer k_i.
3) Create a new vocabulary by prefixing the existing vocabulary up to k times with the T-prefix entry.
4) Add the new vocabulary to the existing one, removing all entries that consist only of 1 to k copies of the T-prefix
5) Jump back to step 2 if you would like to build a larger vocabulary.

The longest entries in the resulting vocabulary only differ in the last symbol a. The remaining common prefix is a series of k_i-long runs of the respective chosen T-prefixes, called x here as this is the string of interest in our upcoming analysis. Titchener and Nicolescu discovered and proved that all finite strings can be built in this way, and that given a finite string x, it is possible to derive the construction parameters, including the k_i, needed to build x, via an algorithm called *T-decomposition*. Speidel and Yang have since developed a T-decomposition algorithm which runs in $O(|x| \log |x|)$ time and space complexity [15]. This algorithm may be conceptually thought of as a decoder

that simultaneously decodes over all the vocabularies until the final one is found.

Since the vocabularies represent highly self-synchronizing codes, the decoder synchronizes at least partially into instances of repeated patterns regardless of their position or separation in the string. Each repeated pattern signals a long T-prefix or a large k_i and therefore lowers the number of steps we need to build x. The T-complexity C_T of a string xa is defined as the weighted number of construction steps required:

$$C_T(xa) = \sum_i \log_2(k_i + 1). \quad (1)$$

There are thus two factors that contribute to a low T-complexity: repetition of T-prefixes (high k_i) and the choice of long T-prefixes (which contain many earlier T-prefixes in their makeup). Note that both factors lower the number of terms in the sum, and both describe effects traditionally associated with lower entropy/complexity in the Shannon and LZ sense.

As T-complexity measures the number of steps in the constuction of xa, it can be compared directly to the LZ production complexity, which also measures such steps but does not weight them. Because of the weighting, T-complexity offers a higher resolution than the LZ production complexity. The latter yields integer complexities between $\log_2 |xa|$ and $|xa|$, whereas T-complexity yields complexities of the form $\log_2 n$ where n can be any integer with $|xa| \leq n \leq 2^n$.

A complexity measure that is an information measure also ought to have a linear upper bound (so one can add information to it sustainably at a constant maximal rate). The T-complexity does not meet this requirement at least for shorter strings. In fact, the upper bound seems to follow the logarithmic integral, which is only asymptotically linear. Titchener therefore proposed a "linearization" by defining the T-information I_T as the inverse logarithmic integral li^{-1} of C_T:

$$I_T(xa) = \mathrm{li}^{-1}(C_T(xa)). \quad (2)$$

The average T-entropy is then defined as $\bar{H}_T(xa) = I_T(xa)/|x|$.

Speidel, Titchener, and Yang's results [17] show that this approach leads to significantly lower overestimation of low entropies from known sources. At 0.1 bits/bit, average T-entropy overestimates by about the same amount as an LZ77 with a string-sized window. However, it does so at a much reduced computational complexity of $O(|x| \log |x|)$ rather than the $O(|x|^2)$ that the LZ77 would take in this case. This is beneficial under aspects of scalability.

The next section presents experimental results to demonstrate that T-enropy is indeed capable of event detection.

IV. Experimental results

Our experiments use three-hour packet traces caputured at the DMZ border gateway of the University of Auckland. Like Feinstein et. al. and Wagner and Plattner, we also use a sliding window with a fixed number of packets, which we shift by a regular time interval between entropy measurements.

The size of the window has two effects: In the presence of normal traffic, a larger window attenuates the natural fluctuation of the entropy between measurements. However, it also reduces the sensitivity to short and light anomalies. The window size is thus always a compromise between the number of false positives and the number of false negatives. Moreover, one also needs to consider the general traffic rate at the observation point and the size of anomaly one wishes to be able to observe. Our results presented here were obtained with a window size of 5000 packets, which is less than the 10000 packets recommended by Feinstein et. al. as "reasonable" but maximises the entropy signal-to-noise ratio of the SPIM signal in the plots below.

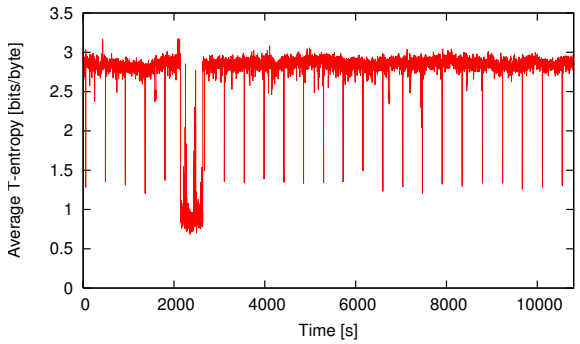

Fig. 1. T-entropy plot of unfiltered trace file.

Between measurements, the window was shifted by 100 ms. The average packet rate was about 8000 s^{-1}. It is also important to note which properties of the datagrams were used in the symbols. Depending on the properties chosen for observation, some events may produce T-entropy increases and some may produce drops. For example, if one only maps SYN flags in TCP packets, one obtains a low entropy baseline. A (moderate) SYN flood may make the ratio between SYN and non-SYN packets more balanced and thus yield a higher average T-entropy. If one uses destination addresses instead, the same SYN flood causes a drop in T-entropy. The packet properties taken into account in this T-entropy calculation were the complete IPv4 information including the first few bytes of the payload.

A first glance at the T-entropy analysis in Figure 1 shows three things: Firstly, the T-entropy of the "normal" traffic was between 2.3 and 2.5 bits per symbol over the entire observation period – a figure also observed in other traces from the same location. Secondly, we observe a series of sharp T-entropy drops roughly once every ten minutes. Thirdly, there is a sharp and deep drop of about 10 minutes' duration during the first hour of observation. This confirms the existing observation by other authors that the entropy of normal network traffic only varies within a narrow band.

Inspection of the traces files revealed a large presence of SPIM packets (Microsoft Instant Messenger SPAM) during the short drops. The large drop contained a large number of outgoing packets with SYN flags set to TCP port 5406 at a single address in the UK, i.e., a SYN flood attack. To verify that these packets did indeed cause the drops in T-entropy, they were removed. Figure 2 (top) shows the same measurement

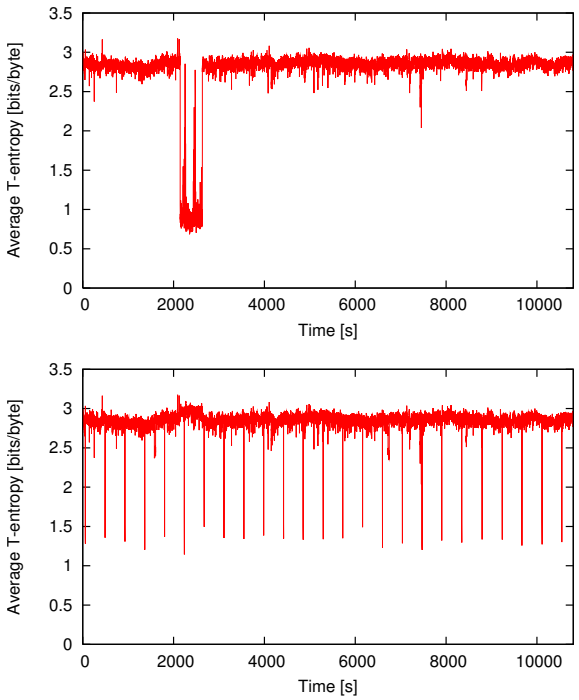

Fig. 2. T-entropies of the trace with SPIM (top) and SYN packets (bottom) removed.

with SPIM packets removed. The overall entropy remains virtually unchanged, however, all the regular drops associated with the SPIM have disappeared. Similarly, if one removes **all** TCP packets which have their SYN flag set, the T-entropy of the "normal" traffic drops only by an almost unnoticeable amount, but the large drop associated with the SYN flood vanishes completely. This is shown in the bottom plot of Figure 2. Note however that it is replaced by a noticeable "hump", which indicates that the attack has had some sort of effect on the remaining network traffic and may hence have been of interest to the network operator. Also note that the SPIM now also appears at the time that the SYN flood was observed.

If both SPIM packets and SYN packets are removed, some drops remain, albeit with less depth. For many of these, trace analysis has yielded traffic that would have been of interest under aspects of anomaly detection. The first drop to below 2 bits/symbol is caused by a large web download from within the DMZ – not a common originator for this kind of traffic. The second drop is a large batch of identical e-mails being delivered – most likely SPAM, the third is most likely a BitTorrent download, etc. The big drop between 6,000 and 8,000 seconds is a very strange e-mail transaction with a burst of 52 byte datagrams originating from port 25.

Another way of giving a proof-of-concept is by injecting an artificial event into "normal" traffic. This route is followed by Feinstein et. al., for example. Figure 3 shows the effects of a synthetic SYN flood event for a range of packets. Note that this method has its limitations, for two reasons: Firstly, the interarrival time distribution of the artificial event, which is al-

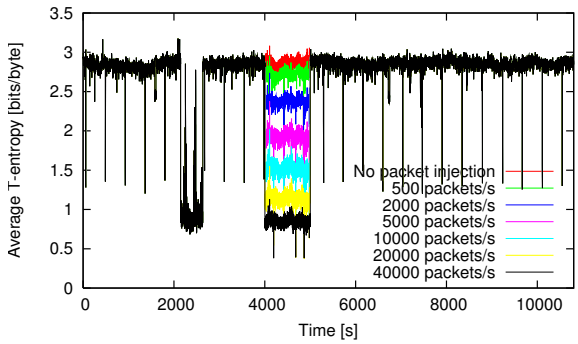

Fig. 3. A synthetic SYN flood at various packet rates

ways a function with many parameters, has some influence on the entropy. Secondly, this method does not account for other network participants being affected or reacting to the event. In other words: If one filters the traffic in Figure 3, only the real SYN flood leaves a "hump". Nevertheless, it demonstrates that T-entropy also works in this respect. Synthetic port scans are also detectable from about 350 scan packets per second.

V. T-ENTROPY VS. LZ-BASED MEASURES

As discussed in Section II, LZ production complexity is the most promising complexity measure from the LZ family of algorithms, at least under aspects of thoroughness. If Figure 1 is recreated using LZ production complexity, it looks almost the same, except that we are now measuring the production complexity of our mapped sliding window rather than its average T-entropy. Computation takes about 20 times longer, though, and this drawback increases almost quadratically with window size. LZ78, on the other hand, produces what appears to be a less "noisy" plot at face value. However, this is due to the reduced sensitivity (resolution) at "normal" entropy levels, and a similar effect can be achieved by restricting T-complexities to integer values. That is, smaller and shorter events would be somewhat easier to detect using T-entropy.

VI. CONCLUSIONS

T-entropy is a versatile measure that can be used in any application where one would normally use a compression ratio or a Shannon entropy as complexity or entropy estimator. This includes the entropy-based network anomaly detection schemes presented in this paper and in the selection of papers referenced here. Others like Wagner and Plattner have already pointed out that there are limits to entropy-based detection and that its strength is in its generality [13], but our results show that there is scope for improvement in the measure used. T-entropy combines scalability with good estimation accuracy (at least for sources with known entropy), especially at low entropies. The results above demonstrate that it is able to detect SYN floods, SPIM, certain forms of e-mail anomalies and other unusual traffic. In an earlier publication [16], we have also been able to demonstrate that it would have detected the arrival of the slammer worm reliably. Wagner and Plattner's statement that entropy-based detection "does not work well for

slow worms or small-scale attacks" can be explained relatively well by theoretical argument. However, an accurate measure can assist in pushing the boundaries.

We would like to thank the staff at the University's Information Technology Services department for their assistance in this research, and Mark Titchener and Yang Jia for helpful ideas and challenging discussions.

REFERENCES

[1] A. N. Kolmogorov: *Three approaches to the quantitative definition of information*, Probl. Inform. Transmis., 1, 1965, pp. 4–7

[2] A. Lempel and J. Ziv: *On the Complexity of Finite Sequences*. IEEE Trans. Inform. Theory, 22(1), January 1976, pp. 75-81.

[3] J. Ziv and A. Lempel: *A Universal Algorithm for Sequential Data Compression*, IEEE Trans. Inform. Theory, Vol 23, No. 3, May 1977, pp. 337-343.

[4] J. Ziv and A. Lempel: *Compression of Individual Sequences via Variable-Rate Coding*, IEEE Trans. Inform. Theory, Vol 24, No. 5, September 1978, pp. 530-536.

[5] M. R. Titchener: *A Deterministic Theory of Complexity, Information and Entropy*, IEEE Information Theory Workshop, February 1998, San Diego.

[6] R. Nicolescu and M. R. Titchener, *Uniqueness Theorems for T-Codes*, Romanian Journal of Information Science and Technology, 1(3), March 1998, pp. 243–258.

[7] M. R. Titchener, *A novel deterministic approach to evaluating the entropy of language texts*, Third International Conference on Information Theoretic Approaches to Logic, Language and Computation, June 16-19, 1998, Hsi-tou, Taiwan.

[8] M. R. Titchener, *Deterministic computation of string complexity, information and entropy*, International Symposium on Information Theory, August 16-21, 1998, MIT, Boston.

[9] M. R. Titchener: *A measure of Information*, IEEE Data Compression Conference, Snowbird, Utah, March 2000.

[10] W. Ebeling, R. Steuer, and M. R. Titchener: *Partition-Based Entropies of Deterministic and Stochastic Maps*, Stochastics and Dynamics, 1(1), p. 45., March 2001.

[11] A. B. Kulkarni, S. F. Bush, and S. C. Evans: *Detecting Distributed Denial-of-Service Attacks Using Kolmogorov Complexity Metrics*, Technical Information Series Report 2001CRD176, GE Research & Development Center, December 2001, `http://www.crd.ge.com/~bushsf/ftn/2001crd176.pdf`

[12] L. Feinstein, D. Schnackenberg, R. Balupari, D. Kindred: *Statistical Approaches to DDoS Attack Detection and Response*, Proceedings of the DARPA Information Survivability Conference and Exposition (DISCEX03), 2003

[13] A. Wagner and B. Plattner: *Entropy Based Worm and Anomaly Detection in Fast IP Networks*, 14th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises (WET ICE 2005), STCA security workshop, Linköping, Sweden, June, 2005

[14] M. R. Titchener and A. Gulliver and R. Nicolescu and U. Speidel and L. Staiger: *Deterministic Complexity and Entropy*, Fundamenta Informaticae, v. 64(1-4), 443-461, 2005

[15] Jia Yang, Ulrich Speidel: *A T-Decomposition Algorithm with $O(n \log n)$ Time and Space Complexity*. Proceedings of the IEEE International Symposium on Information Theory, 4-9 September 2005, Adelaide, pp. 23–27.

[16] R. Eimann, U. Speidel, N. Brownlee: *A T-Entropy Analysis of the Slammer Worm Outbreak*, Proceedings of the 8th Asia-Pacific Network Operations and Management Symposium (APNOMS), 27-30 September 2005, Okinawa, Japan, pp. 434–445.

[17] U. Speidel, M. Titchener, J. Yang: *How well do practical information measures estimate the Shannon entropy?*. P120, Proceedings of the 5th International Symposium on Communication Systems and Digital Signal Processing (CSNDSP) 2006, 19-21 July 2006, Patras, Greece.

[18] A. Kulkarni, S. F. Bush: *Detecting Distributed Denial-of-Service Attacks Using Kolmogorov Complexity Metrics*, J. Network Syst. Manage. 14(1): 69-80 (2006)

[19] U. Speidel: *T-Complexity and T-Information Theory – an Executive Summary*, 2nd revised version, CDMTCS Report 286, Centre for Discrete Mathematics and Theoretical Computer Science, The University of Auckland, October 2006. `http://www.tcs.auckland.ac.nz/CDMTCS/researchreports/286ulrich.pdf`.