Empirical and Theoretical Evaluation of Active Probing Attacks and Their Countermeasures

Xinwen Fu[1], Bryan Graham[1], Dong Xuan[2], Riccardo Bettati[1] and Wei Zhao[1]

[1] Department of Computer Science, Texas A&M University
E-mail: {xinwenfu, bwg7173, bettati, zhao}@cs.tamu.edu
[2] Department of Computer and Information Science, Ohio State University
E-mail: xuan@cis.ohio-state.edu

Abstract. A variety of remote sensing attacks allow adversaries to break flow confidentiality and gather mission-critical information in distributed systems. Such attacks are easily supplemented by active probing attacks, where additional workload (e.g., ping packets) is injected into the victim system. This paper presents statistical pattern recognition as a fundamental technology to evaluate the effectiveness of active probing attacks. Our theoretical analysis and empirical results show that even if sophisticated approaches of link padding are used, sample entropy of probing packets' round trip time is an effective and robust feature statistic to discover the user payload traffic rate, which is important for maintaining anonymous communication. Extensive experiments on local network, campus network, and the Internet were carried out to validate the system security predicted by the theoretical analysis. We give some guidelines to reduce the effectiveness of such active probing attacks.

1 Introduction

This paper analyzes a class of active attacks on traffic flow confidentiality. In particular, we are interested in attacks that disclose the traffic rate on a network link. Traffic rate is critical information in many scenarios. For example, if Alice communicates with Bob through an anonymity communication network[3], an attacker may infer this communication relationship (sender and receiver) if he determines that the rate of output traffic from Alice roughly equals the rate of input traffic to Bob. In [19], Serjantov and Sewell give more examples about the importance of hiding traffic rates in Mix networks, and the authors of NetCamo [11] show examples in other mission-critical applications.

To hide traffic rate, dummy traffic is typically used to pad the original traffic, i.e., user payload traffic rate. As a result, the padded traffic has a different rate from the original traffic, so that we achieve traffic rate hiding. Traffic padding[4] can be end-to-end padding (in which sender and receivers control the padding) and link padding (in

---

[3] Anonymity communication networks use *Mix* techniques pioneered by Chaum [3] and are often denoted as Mix networks.

[4] We distinguish traffic padding from packet padding. Packet padding hides the *length* of individual packets by adding padding data to packets. Traffic padding hides the temporal characteristics, for example rate, of a flow of packets. Traffic padding relies on packet padding and encryption to render dummy packets non-detectable.

which the intermediate hops control the padding). In either case, the original traffic is often padded to have a constant rate using a periodic timer; this technique is denoted as CIT (constant interval time) padding. The original traffic can also be padded to have a variant rate using a non-periodic timer, and this technique is denoted as VIT (variable interval time) padding. However, traffic padding is not a cure-all. Traffic analysis attacks have been developed to obtain the information of traffic rate even if traffic padding is used.

In terms of techniques, traffic analysis attacks can be passive and active. (a) In a *passive traffic analysis attack*, an adversary passively collects traffic data (e.g., traffic dumped using *tcpdump*) and performs an analysis on it. The authors of [9] describe statistical traffic analysis attacks that estimate the user payload traffic rate if CIT padding is used and how the effectiveness of this type of attack can be significantly reduced with the use of appropriate VIT padding. (b) In an active traffic analysis attack, the adversary interferes with the normal activity of a victim network in a seemingly innocuous way and tries to acquire critical information by analyzing the victim network's response to the interference.

One specific kind of active traffic analysis attack is active probing attack, in which the adversary injects probing traffic (e.g., FTP/TELNET/Ping/etc.) into the victim network and analyze the network's response on the probing traffic. Wei Dai [4] briefly describes cases of active probing attacks aimed at getting traffic rate between pairs of users to break Freedom anonymity systems [1] by insiders, such as malicious users.

This paper analyzes active probing attacks by *outsiders* and develops countermeasures against these forms of attacks for systems that use VIT traffic padding. As an illustrative example, we use a simple *ping-based* probing attack, where the adversary *pings* various locations in the network in order to gain information, such as the payload traffic rate. We define *detection rate* as the probability that the adversary correctly recognizes the payload traffic rate and use it to evaluate the information assurance of a security system. We systematically evaluate the detection rate of various statistical methods which the adversary can then use to analyze the probing traffic. Specifically, using statistical pattern analysis as the framework, we find that sample mean, sample variance, and sample entropy of the round trip time (RTT) of probing packets can help the adversary track the payload traffic rate's changing pattern and obtain the payload traffic rate. Of those statistics, sample entropy is robust (i.e., not sensitive to outliers) and effective in terms of detection rate.

We also report results from extensive experiments in various situations, including local area network in a laboratory, campus networks, and wide area networks. Our data consistently demonstrates the usefulness of our analytic model and correctness of detection rates predicted by the closed-form formulae.

As with countermeasures, active probing attacks (ping-based and others) can generally be made ineffective through simple means, for example, by randomly delaying all non-payload traffic. We will empirically and analytically evaluate the effectiveness of such countermeasures by measuring to what extent they reduce the effectiveness of probing attacks. We note that the methodology of delaying outgoing traffic from security gateways may be desired for security contexts in addition to the context described here.

The rest of this paper is organized as follows. Section 2 reviews traffic padding as the countermeasure to traffic analysis attacks and recent practical traffic analysis attacks in different scenarios. We present the network model, padding mechanism, and an adversary's analysis strategies in Section 3. In Section 4, we develop a theoretical model and derive closed-form formulae for detection rates for different statistics. Section 5 validates our theory through experiments. Based on empirical and analytic techniques, Section 6 gives countermeasures to active ping probing attacks. Section 7 summarizes this paper and discusses possible extensions.

2 Related Work

Shannon [20] describes his perfect secrecy theory, which is the foundation for the ideal countermeasure system against statistical analysis attacks. Traffic padding is a major class of countermeasures that researchers have proposed to counter traffic analysis attacks. Baran [2] proposes the use of heavy unclassified traffic to interfere with the adversary's tampering of the links of a security network system used for communicating classified information. He also suggests adding *dummy*, i.e. fraudulent, traffic between fictitious users of the system to conceal the true amount of traffic.

A survey of countermeasures for traffic analysis is given in [29]. To mask the frequency, length and origin-destination patterns of an end-to-end communication, dummy messages are used to pad the traffic to a predefined pattern. It is evident that such a predefined pattern is sufficient but not necessary based on the perfect secrecy theory [20].

The authors in [15, 16, 28] give a mathematical framework to optimize the bandwidth usage while preventing traffic analysis of the end-to-end traffic rates. Timmerman [27] proposes an adaptive traffic masking (hiding) model to reduce the overhead caused by traffic padding. When the rate of real traffic is low, the link padding rate is reduced as well, in order to conserve link bandwidth. Thus large-scale variations in traffic rates are still observable.

The authors of NetCamo [11] provide the end-to-end prevention of traffic analysis while guaranteeing QoS (the worst case delay of message flows) in time constraint communication networks.

To protect the anonymity of email transmissions, Chaum [3] proposes the use of a *Mix* - a computer proxy. One technique used by a Mix is to collect a predefined number of fixed-size message packets from different users and to shuffle the order of these packets before sending them out. Many researchers suggest using constant rate padding (i.e., make the traffic rate appear as constant) between the user and the proxy (e.g., [24]). Raymond in [18] gives an informal survey of several *ad hoc* traffic analysis attacks on systems providing anonymous services. For example, by correlating traffic rate or volume, the attacker may discover the end points of a communication. One of his conclusions is that traffic padding is essential to achieve communication anonymity. Danezis *et al.* [5] give a list of attacks to anonymity systems. Most of those attacks are only briefly discussed and lack systematic analysis. Tarzan [8] provides anonymity in a peer-to-peer environment by using link padding to counter possible attacks.

Recently researchers have disclosed some advanced statistical traffic analysis attack techniques. Song *et al.* [22] describe how SSH 1 and SSH 2 can leak user passwords under a passive traffic analysis attack. The authors illustrate how the inter-packet times

in a SSH session accurately reflect the typing behavior of the user by exposing the inter-keystroke timing information. This in turn can be used to infer plaintext as typed on the keyboard. To prevent this, the authors propose padding traffic on the SSH connections to make it appear to be a constant rate. When there are not enough packets to maintain the constant rate, fake (dummy) packets are created and sent.

Felten and Schneider [7] develop an active timing attack based on browsing a malicious web page. This malicious web page is able to determine if a user has recently browsed a different target web page. The malicious web page contains embedded attack codes, which try to download a web file from the target webpage. If the user has recently browsed the target webpage, it is highly possible that the target webpage is cached locally, in which case, the access time will be very small, otherwise it will be much larger. The malicious code reports the access timing to the attacker, and then the attacker can decide if the user has recently browsed the target webpage by this access timing. The malicious codes can be Javascript codes, or with a little more effort, time measurement HTML codes. Clearly this attack is very difficult to prevent, and the only perfect countermeasure is to turn off the cache.

SafeWeb [13] is a web service, that uses anonymizing servers, which in turn behave like mixes and act as proxies between users and the web servers. The proxy downloads the requested webpage on behalf of the user and forwards it to the user in an encrypted form. Hintz [12] shows how observers can take advantage of the HTML weakness of using a separate TCP connection for each HTML object (such as HTML texts, image files, audio annotations, etc.) to deploy passive traffic analysis attacks. The number of TCP connections and the corresponding amount of data transferred over each connection form a fingerprint, which allows an observer to identify the accessed webpage by correlating fingerprint data with traffic observed between the user and the anonymizing server. To invalidate these fingerprints, we have to merge all the connections into a single connection or add noise (fake messages, etc.) to the web traffic flows. Sun *et al.* [23] use many experiments to show the possibility and efficiency of the above exploit.

3   System Models

In this section, we present the network model used in our study and then discuss link padding mechanisms that are used as countermeasures for passive traffic analysis attacks. Finally, we formally define the model of the adversary that uses statistical pattern recognition strategies for active *ping* probing attacks on these security systems, which employ link padding mechanisms.

3.1   Network Model

In this work, we assume that the network consists of *protected subnets*, which are interconnected by *unprotected networks*. Traffic within protected subnets is assumed to be shielded from observers. Unprotected networks can be public networks (e.g., the Internet) or networks that are deployed over an easily accessible broadcast medium. These networks are accessible to observation by third-parties, and limited services such as
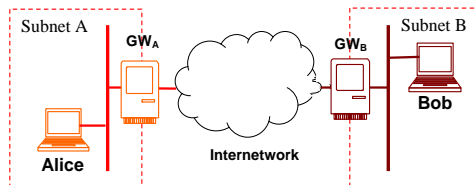
Fig. 1. Network Model

ping are also available. Thus, these networks are open to traffic analysis and other limited probing attacks. This model captures a variety of situations, ranging from battleship convoys (where the large-scale shipboard subnets are protected and the inter-ship communication is wireless) to communicating PDAs (where the protected subnets consist of single nodes).

Figure 1 illustrates the network setup. Two security gateways GW_A and GW_B are placed at the two boundaries of the unprotected network and provide the link padding necessary to prevent traffic analysis of the payload traffic exchanged between the protected subnets A and B.

Note that the gateways can be realized either as stand-alone boxes, modules on routers, switches, software additions to network stacks, or device drivers at the end hosts. In this paper, we assume that they are stand-alone boxes. (Please **note:** In the anonymous communication network such as Onion Routing [24], the link padding function can be implemented as device drivers at the hosts of end users (Alice's and Bob's machine), who connect to onion routers. Our result and analysis in this paper should also be valid in this case since the mechanism causing the problem of information leaking is similar.) To simplify the discussion, the communication is one-way from Subnet A to Subnet B. Consequently, GW_A and GW_B are also called *sender gateway* and *receiver gateway* respectively.

3.2   Link Padding Mechanism

The motivation of link padding is to ensure traffic flow confidentiality, that is, to prevent the adversary from performing traffic analysis and inferring critical characteristics of the payload traffic exchanged over the unprotected network. We limit the adversary's interest to the *payload traffic rate*, that is, the rate at which payload traffic is exchanged between protected subnets. Specifically, we assume that there is a set of discrete payload traffic rates $\{\omega_1, \cdots, \omega_m\}$. The rate of payload traffic from the sender will be one of those m rates at a given time. Consequently, the objective of the adversary is to identify at which of the m rates the payload is being sent. But, we will also demonstrate how the adversary may use the approaches in this paper to track the continuous changing pattern of the payload traffic.

One way to counter the traffic analysis attacks is to "pad" the payload traffic, that is, to properly insert "dummy" packets in the payload traffic stream so that the real payload status is camouflaged. There are many possible implementations of link padding

algorithms on the two gateways in Figure 1. The most common method uses a timer to control packet sending and works as follows: (a) On GW_A, incoming payload packets from the sender Alice are placed in a queue. (b) An interrupt-driven timer is set up on GW_A. When the timer times out, the interrupt processing routine checks if there is a payload packet in the queue: (1) If there are payload packets, one is removed from the queue and transmitted to GW_B; (2) Otherwise, a dummy packet is transmitted to GW_B. This timer can be a constant interval timer (CIT), which is a periodic one with a constant interval between two consecutive timeouts. This is the most commonly used method for traffic padding, i.e., the constant rate traffic padding. The timer can also be a variable interval timer (VIT) with a variable amount of time between two consecutive timeouts, where the interval satisfies some distribution. We denote padding using these two different timers as CIT padding and VIT padding, respectively.

3.3 Adversary Strategies

Recall that we assume that the objective of the adversary is to identify at which of the m possible rates the payload is being sent. We need to discuss the adversary's power before we proceed further.

We assume an external adversary. That is, the adversary is not a participant of either Subnet A or Subnet B. He does not compromise sender and receiver gateways. The adversary can only get access to the two subnets in seemingly legal ways such as pinging the two gateways.

Traffic flow confidentiality is ensured in the system in Figure 1 by VIT padding under passive traffic analysis attack. Packet contents are perfectly encrypted, all packets have a constant size (padded or manipulated), and dummy packets cannot be distinguished from payload packets. The authors of [9] proposed using VIT padding as an alternative to the commonly used CIT padding and show how CIT padding is extremely difficult to implement in practice and how minute disturbances make CIT padding subject to a sophisticated passive traffic analysis attack that measures the packet interarrival time of packets on the unprotected link.

We also assume that the adversary has complete knowledge about the gateway machines and the countermeasure algorithms used for preventing traffic analysis. Thus, the adversary can simulate the whole system, including the gateway machines, to obtain *a priori* knowledge about traffic behavior. In many studies on information security, it is a convention that we make worst-case assumptions like this. But, we will also show in this paper, even without the capability of simulating the system, the adversary can also track the traffic rate changing pattern by the method introduced in this paper.

Based on these assumptions, the adversary may deploy a sophisticated ping probing attack aimed at determining the payload traffic rate from $\{\omega_1, \cdots, \omega_m\}$. In the attack, the adversary pings the sender gateway GW_A, analyzes the statistics of round trip time of these ping packets and tries to figure out Subnet A's payload traffic rate even if GW_A uses VIT padding (If the padding is implemented as a device driver on Alice's host, the ping probing is aimed at getting Alice's real payload traffic rate). We use this ping attack as a model to analyze a much larger class of active probing attacks.

The adversary can analyze his sample of ping RTT data based on Bayes decision theory [6]. The entire attack strategy consists of two parts: Off-line training and run-time classification. We now describe them below.

Off-line training

The off-line training component can be decomposed into the following steps:

1. The adversary selects a statistic of the RTT sample of size n. This statistic is called *a feature* and will be used for traffic rate classification. Possible features we study in this paper are *sample mean*, *sample variance*, and *sample entropy*.
2. The adversary emulates the entire link padding system and collects RTT information at different payload traffic rates. From this information, the adversary derives the *Probability Density Functions* (PDF) of the selected statistical feature. As histograms are usually too coarse for the distribution estimation, we assume that the adversary uses the *Gaussian kernel estimator of PDF* [21], which is effective in our problem domain.
3. Based on the PDFs of statistical features for different payload traffic rates, Bayes decision rules are derived. Recall that there are m possible payload traffic rates ω_1, $\cdots$, ω_m. The Bayes decision rule can be stated as follows:
The sample represented by feature s corresponds to payload rate ω_i if

$$p(\omega_i|s) \geq p(\omega_j|s) \tag{1}$$

That is,

$$p(s|\omega_i)Pr(\omega_i) \geq p(s|\omega_j)Pr(\omega_j) \tag{2}$$

for all $j = 1, \cdots, m$.
Here $Pr(\omega_i)$ is the *a priori* probability that the payload traffic is sent at rate ω_i, and $p(\omega_i|s)$ is the *post priori* probability that the payload traffic is sent at rate ω_i when the collected sample has the measured feature s.

Run-time Classification

Once the adversary completes his training phase, he can start the classification at run-time. We assume the adversary has some means to ping the gateways GW_A and GW_B. In particular, when he wants to determine the current payload rate, the adversary collects a sample of ping RTTs. He calculates the value of the statistical feature from the collected sample and then uses the Bayes decision rules derived in the training phase to match the collected sample to one of the previously defined payload traffic rates.

4 Derivation of Detection Rate

Given the models described in the previous section, we would like to evaluate the security of the system in Figure 1 in terms of detection rate. *Detection rate* is defined as the probability that the adversary can correctly classify the payload traffic rate protected by security gateways. In this section, we derive the closed-form formulae for detection rates when the adversary uses sample mean, sample variance, or sample entropy, as the

statistical feature, respectively. Our formulae will be approximate ones due to the complexity of the problem. Nevertheless, these formulae do correctly reflect the impact of various system parameters, including the type of padded traffic, sample size, and statistical feature used. These relationships are extremely useful in understanding the nature of the attack and designing effective countermeasures. In the next section, we will see that experimental data well matches the detection rate predicted by our approximation formulae.

Let $\{X_1, X_2, \cdots, X_n\}$ be a sample of ping RTT with sample size n. The *sample mean* $\bar{X}$, sample variance Y, and sample entropy $\widetilde{H}$ are defined below:

$$\text{Sample Mean: } \bar{X} = \frac{\sum\limits_{i=1}^{n} X_i}{n} \tag{3}$$

$$\text{Sample Variance: } Y = \frac{\sum\limits_{i=1}^{n} (X_i - \bar{X})^2}{n - 1} \tag{4}$$

$$\text{Sample Entropy: } \widetilde{H} \approx -\sum_i \frac{k_i}{n} \log \frac{k_i}{n} + \log \Delta x \tag{5}$$

where in (5) we use the histogram-based entropy estimation developed in [14]. k_i is the number of sample points in the ith bin, and Δx is the histogram's bin size. In Appendix A, we provide a way to calculate the optimal bin size for the estimation of entropy.

Using sample mean, sample variance, and sample entropy as defined above, our experiments as (illustrated in Figure 4 (d)) show that the adversary can continuously track the changing pattern of the user payload traffic rate. Below we give close-form formulae for simple cases in which the user payload traffic has two statuses: low rate ω_l and high rate ω_h.

4.1 Detection Rate for Recognizing Two Payload Traffic Rates

Because of the page limit, we just list the major theorems about sample mean, sample variance, and sample entropy. Interested readers can refer to [10] for details. Before we introduce these theorems, let's first investigate the reason of the failure of VIT padding against the ping probing attack, which we will demonstrate below. The reason for this failure lies in the subtle interaction between the traffic padding system and the probing traffic. While GW_A's network subsystem processes payload packets from Subnet A in Figure 1, the processing of ping packets is delayed. A higher rate of payload traffic causes more possible delay on ping packets. This means that sample mean, sample variance, and sample entropy of the RTT of the probing packets at a given sample size n are changed, and there is some kind of correlation between the user payload traffic rate and sample mean, sample variance, and sample entropy of the RTT of the probing packets. The adversary can explore this correlation to discover the user payload traffic rate.

The ping RTT can be represented as a random variable RTT. As analyzed above, under different user payload traffic rates, i.e., low rate and high rate in our illustrative

case, we will have random variables RTT_{low} and RTT_{high}, whose means are denoted as μ_l and μ_h respectively, and whose variances are denoted as σ_l^2 and σ_h^2 respectively. Also we define r as the ratio between σ_h^2 and σ_l^2.

$$r = \frac{\sigma_h^2}{\sigma_l^2} \tag{6}$$

The following theorem provides closed-form formulae for estimation of detection rate when the adversary uses sample mean, sample variance, and sample entropy as feature statistics.

Theorem 1. *The detection rate by sample mean, $v_{\bar{X}}$ can be estimated as follows:*

$$v_{\bar{X}} \approx 1 - (e^{-\frac{1}{4}\frac{(\mu_h - \mu_l)^2}{\sigma_h^2 + \sigma_l^2}})^n \frac{1}{\sqrt{2(1/\sqrt{r} + \sqrt{r})}} \tag{7}$$

The detection rate by sample variance, v_Y, can be estimated as follows:

$$v_Y \approx \max(1 - \frac{C_Y}{n-1}, 0.5) \tag{8}$$

where C_Y is calculated by

$$C_Y = \frac{1}{2(1 - \frac{1}{r-1}\log r)^2} + \frac{1}{2(\frac{r}{r-1}\log r - 1)^2} \tag{9}$$

The detection rate by sample entropy, v_H, can be estimated as follows:

$$v_{\bar{H}} \approx \max(1 - \frac{C_H}{n}, 0.5) \tag{10}$$

where $C_{\bar{H}}$ is calculated by

$$C_{\bar{H}} = \frac{1}{2(\log(\frac{r}{r-1}\log r))^2} + \frac{1}{2(\log(\frac{r-1}{\log r}))^2} \tag{11}$$

We have a few observations from the above Theorem:

1. For sample mean, the detection rate is exponentially increasing with sample size n. This implies that a small difference between μ_h and μ_l may cause detection rate to dramatically increase with the increase of sample size. Furthermore, the detection rate decreases with an increase in variance σ_h^2 and σ_l^2.
2. For sample variance, the detection rate is an increasing function in terms of sample size n. When $n \to \infty$, the detection rate is 100%. This means that if the payload traffic lasts for sufficient time at one rate (either low or high), and the adversary can get a sample of a sufficiently large size, he may detect the payload traffic rate by sample variance of ping RTT. Furthermore, the detection rate is an increasing function of r in (6), where $r \geq 1$. That is, the smaller r, the closer the two variances under different payload traffic rates, and intuitively the lower the corresponding detection rate. When $r = 1$, the detection rate is 50%. That is, the probing attack using sample variance will fail.
3. For sample entropy, the detection rate is also an increasing function in terms of sample size n. Also, the detection rate is also an increasing function of r in (6), where $r \geq 1$. When $r = 1$, the detection rate reaches 50%.

4.2   Detection Rate for Payload Traffic with Periodically Changing Rate

In practice, the rate of payload traffic from Subnet A in Figure 1 changes with time. Here, we consider the case in which the payload rate changes periodically. We deduce the detection rate in Theorem 2 and its corollary. For a rigid proof of these formulae, please refer to [10]. Here we briefly introduce the principle. To deploy probing attacks, an adversary pings the sender gateway and collects a sample of n RTTs of probing packets. This sample may be partitioned into a few segments, for example, the first l RTTs are collected when the user payload traffic rate is low and the other $n-l$ RTTs are collected when the user payload traffic rate is high. Assuming that we have L possible partitions: $\{Partition_i : 1 \leq i \leq L\}$. For $Partition_i$, we can derive its occurrence probability $P(Partition_i)$ and the average recognition error rate conditioned on this case of partition $Pr(error|Partition_i)$. We also assume that the correct recognition is that the recognition result matches the user payload traffic rate when the first packet of the sample is collected. Then we have the general form of detection rate formula in Theorem 2.

Theorem 2. *The detection rate v_d for payload traffic with periodically changing rate is*

$$v_d = 1 - \sum Pr(error|Partition_i)P(Partition_i) \tag{12}$$

For the case of two payload traffic rates, assuming traffic of each rate lasts for half of a single period, M is the number of ping RTT sample point in half of a period (ping packets are sent out at a constant rate) and n is the sample size, we have the following corollary from Theorem 2.

Corollary 1. *In case of $n < M$, a closed form of detection rate is given in (13),*

$$v_d = 1 - (\frac{M - n + 1}{M}\epsilon + \frac{n - 1}{2M}) \tag{13}$$

where ϵ is the classification error: $\epsilon = 1 - v$, where v can be calculated in (7), (9) and (10) for different features.

Please refer to Appendix B for the proof. From Corollary 1, we can see that when the ping packet rate is fixed, the larger the payload rate changing period, the larger M and thus the bigger v. This is intuitive. v has a complicated relation with n because of ϵ's relation with n. Given M, v has maximum value at some n.

5   Evaluations

In this section, we evaluate how the theoretical analysis of detection rate from the previous section compares to results from experiments that are designed to reflect real-life situations.

In the experiments, we assume that the adversary uses a high-performance network analyzer, such as Agilent's J6841A [25], to dump ping packets. A series of experiments were carried out. In terms of experimental environments, we consider the following
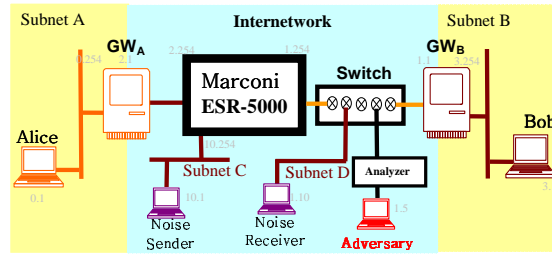
Fig. 2. Experiment setup in laboratory

cases: a) a laboratory environment (LAN), b) a campus network (MAN), and c) a wide area network (WAN).

GW_A and GW_B in Figure 1 run TimeSys Linux/Real-Time [26]. To counter traffic analysis attacks, VIT padding is used. The timer interval satisfies a normal distribution $N(10ms, 3ms^2)$, which is a very powerful setting for resisting passive traffic analysis attacks [9]. Thus, the average rate of padded traffic between the two security gateways is 100 packets per second (pps). The payload has two average rate states: 10 pps and 40pps. We assume both rates occur in equal probability. Note that for such a system with two possible payload traffic rates, the detection rate for the adversary is lower-bounded at 50% corresponding to random guessing. For all the experiments, the adversary uses an appropriate rate of ping packets whose size is 512 bytes.

5.1 Experiments in a Laboratory Environment

The advantage of performing the experiments in a laboratory environment is that we can control the cross traffic over the network. The disadvantage is that the generated cross traffic may not have the same characteristics as that in a real network. Nevertheless, our experiment setup is shown in Figure 2.

The two gateways are connected by a Marconi ESR-5000 enterprise switching router [17]. Subnet C is connected to the router as the cross traffic (noise) generator while the cross traffic receiver is located in Subnet D. Note that the cross traffic shares the outgoing link of the router, creating a case where the cross traffic makes an impact on the padded traffic. The adversary is behind the Marconi router and tries to ping the sender gateway GW_A.

Results of Probing Attacks on Stable Payload Traffic

By stable payload traffic, we mean that the traffic from user subnets lasts for a relatively long time at a roughly constant rate.

Figure 3 (a) and (b) shows the detection rate by different features for cases of without cross traffic and with cross traffic (users in Subnet C communicate with users in Subnet D). We have the following observations:
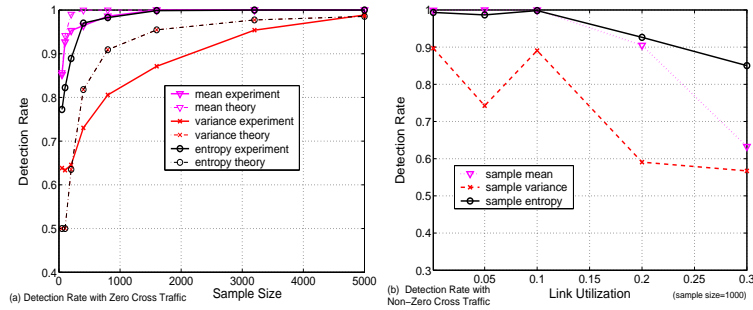
Fig. 3. Detection Rate for Stable Payload Traffic

1. As the sample size increases, as shown in Figure 3 (a), detection rates for sample mean, sample variance, and sample entropy increase and approach 100%. This shows that when payload traffic lasts for enough time at some rate, these three features can determine the payload traffic rate with 100% accuracy, even if the powerful VIT padding is used. **Security systems using padding fail under probing attacks**. Furthermore, the trend of theoretical detection rate curves coincides well with the trend of empirical curves for the three features.

2. From Figure 3 (a) and (b), sample entropy is a fairly robust feature in detecting the user payload traffic rate. This is because sample entropy defined in (5) is not sensitive to outliers, which influence the performance of sample mean and sample variance, especially when there is cross traffic.

3. In Figure 3 (b), overall, as the link utilization increases, the detection rates of the three features decrease. Intuitively, this is because the cross traffic between Subnet C and Subnet D interferes with ping traffic. In theory, compared to the ping RTT variances σ_l^2 and σ_h^2 in the no cross traffic case, both these variances in case of with cross traffic are increased by a quantity caused by cross traffic. This will cause a decrease in r. As Theorem 1 predicts, the detection rate by all three features drops.

Results of Probing Attacks on Payload Traffic with Periodically Changing Rate

Figure 4 (a), (b) and (c) gives the detection rate for payload traffic with periodically changing rate. Payload traffic of 10pps lasts for 1 minute and traffic of 40pps lasts for the next 1 minute. Figure 4 (d) illustrates how the adversary can track continuously changing payload traffic rate by probing attacks.

We have the following observation from Figure 4.

1. The theoretical curves well match the empirical curves. This validates Theorem 2 and its Corollary 1.

2. As Corollary 1 predicts, there exists a maximum detection rate at some sample size. So, in practice, when the ping probing attack is deployed, the adversary has to choose an appropriate sample size to get an optimal detection rate. A large sample size for payload traffic with small rate changing period may cause a bad detection rate because a sample includes mixed rates of payload packets.
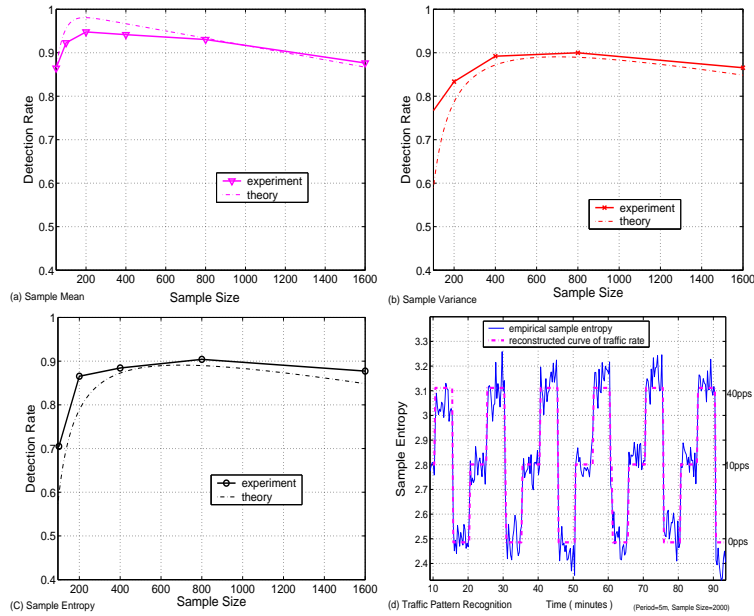
Fig. 4. Detection Rate for Payload Traffic with Periodically Changing Rate

3. In Figure 4 (d), sample entropy (sample size = 2000) is used to track the changing pattern of the user payload traffic rate while the user payload traffic rate has three statuses: 0 pps, 10 pps, and 40 pps. The rate changes for 5 minutes on average. It is clear that the adversary can use sample entropy to reconstruct the payload traffic rate's changing pattern very well. This further validates probing attacks' validity in the general problem of tracking user payload traffic pattern.

5.2   Experiments over Campus and Wide Area Networks

In this subsection, we examine the detection rate when the adversary's ping traffic traverses a campus network and the internet respectively.

Figure 5 shows the setup for the experiments discussed in this subsection. In both cases, the observation point of the adversary is located right in front of the receiver gateway and thus maximally far from the sender. Figure 5 (a) is a setup for experiments over our local campus network[5]. That is, the ping traffic goes through our local campus network before it reaches the sender's gateway. Figure 5 (b) is a setup for experiments over the Internet between a remote campus network and our local campus network. Here, the sender workstation and the sender gateway are located at the remote campus network. The ping traffic goes through the Internet and arrives at the remote campus network. We note that in this case, the path from the sender's workstation to the receiver's workstation spans 15 or more routers.

---

[5] Because the requirement of anonymous submission, related institute information is dropped.

(a) Experiment Setup within Local Campus Network (b) Network Setup between Remote and Local Campus
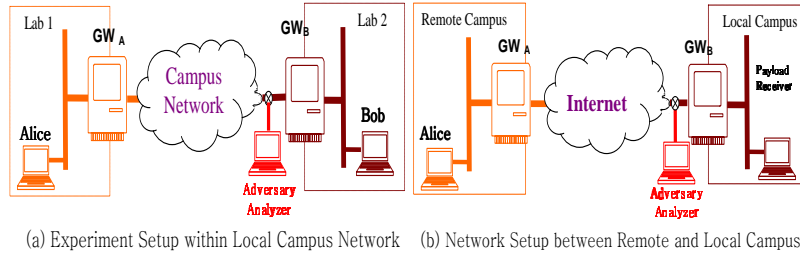
Fig. 5. Experiment setup over campus and wide area networks (WAN)
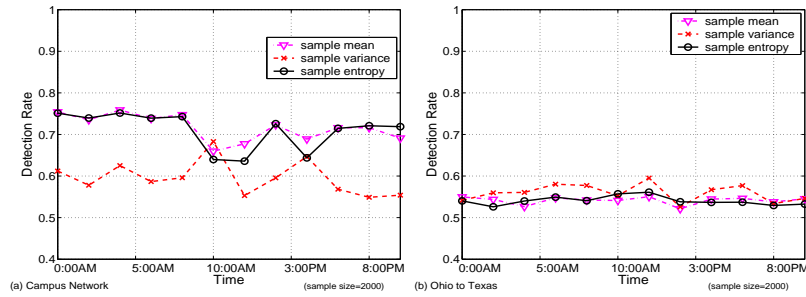

Fig. 6. Empirical detection rates for experiments over campus and WAN (sample size=2000)

In each case, we collect data continuously for a complete day (24 hours). The data for the case of our local campus network was collected on July 16, 2003 while the data for the wide area network case was collected on July 14, 2003.

Figures 6 (a) and (b) display the detection rate throughout the observation period. We have the following observations:

1. When ping traffic traverses just our local campus network, the detection rates of sample entropy and sample mean can approach about 75%. This means that over a medium-sized enterprise network like our local campus network, the cross traffic does have an influence on the ping traffic, but systems using VIT padding scheme alone still cannot resist ping probing attacks effectively.

2. When the padded traffic traverses more network elements, such as the span of the Internet between the remote campus network and our local campus network, the detection rates are much lower. This is because ping traffic has a low scheduling priority at a large number of routers and switches, and the RTT of ping packets is seriously distorted.

6 Countermeasures

To counter the active traffic analysis attacks, there are several possible approaches. The first approach is to disable the ping service on security gateways, but the disadvantage

of this is that ping often is a useful service for debugging a network, e.g., to check if GW_A is alive. Sometimes we cannot sacrifice functionality for the sake of security.
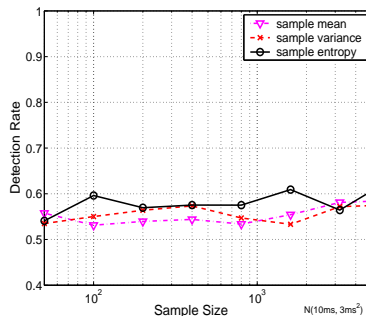

Fig. 7. Detection Rate by RTT of Delayed Ping Packets with Zero Cross Traffic

The second approach is inspired by our theories and experiments. We know that the smaller r and the bigger σ_l^2 and σ_h^2 in (6), the smaller the detection rate. To reduce r and increase σ_l^2 and σ_h^2, we intentionally introduce a random delay to ping packets. This is similar to adding noise to the RTT of ping packets and has the same effect as cross traffic does in Figure 3 (b). This delay satisfies a normal distribution $N(\mu_T, \sigma_T^2)$. It can be perceived that an appropriate selection of μ_T and σ_T will dramatically reduce the detection rate. To validate this approach, we again use the configuration in Figure 1 as the experiment network setup. There is no cross traffic. Figure 7 gives the detection rate by different statistics when ping packets are delayed by a random interval, which satisfies a normal distribution $N(10ms, 3ms^2)$. We can see that even though the attacker has the best-case (no cross traffic) the detection rate by different feature statistics approaches 50% (the minimum detection rate for two classes recognition) at a large sample size.

A third guideline of countering active ping probing attacks is that we should avoid the case in which user traffic possibly lasts for a long time at a roughly constant rate. For example, in a peer-to-peer anonymous file sharing system, the file should be split into small pieces before uploading and downloading.

7   Conclusions and Final Remarks

In this paper, we evaluate the security of sophisticated traffic padding schemes under active probing attacks. To demonstrate the threat from such attacks, we use ping probing attacks aimed at deriving user payload traffic rates. It is found that by measuring statistics of the round trip time of ping packets injected into security gateways, the adversary can break the padding system, track the user payload traffic changing pattern, and discover exactly the payload traffic rate that security gateways try to protect even if a strong link padding scheme such as VIT padding is used by these gateways. We define detection rate as the probability that the adversary correctly recognizes the payload traffic rate. The ping probing attack can achieve a 100% detection rate.

Of the possible statistics used for probing attacks, sample entropy is an effective and robust feature statistic to explore the correlation between user payload traffic rate and the round trip time of the probing ping packets. The reason for the success of the exploit is that users' payload traffic causes small disturbances to the RTT of ping packets. Moreover, the higher the user traffic rate, the larger this disturbance; therefore, the bigger the entropy. Sample mean and sample variance of ping RTT may also be used for ping probing attacks, but they are sensitive to outliers and not robust in a noisy network.

Under the framework of statistical pattern recognition, we formally model different statistics' detection rates of the payload traffic rate. Our empirical results match our theoretical analysis. This framework can be easily extended to analyze other statistical analysis attacks because of statistical pattern recognition's maturity and abundance of analytical techniques.

We also conducted extensive experiments in various situations including local area network in a laboratory, campus networks, and wide area networks. We found that for a metropolitan area network such as a campus network, the ping probing attack can still obtain a good detection rate. These extensive empirical data consistently demonstrates the usefulness of our formal model and correctness of detection rate predicted by the closed-form formulae.

Following our theory, after a careful analysis we propose randomly delaying the ping packets to counter the active probing attack. Our experiments and theories validate the effectiveness of this scheme. Other guidelines are also provided.

References

1. Adam Back, Ian Goldberg, and Adam Shostack. Freedom systems 2.1 security issues and analysis. White paper, Zero Knowledge Systems, Inc., May 2001.
2. P. Baran. On distributed communications: Ix security, secrecy, and tamper-free considerations. *Memo RM-3765-PR, Rand Corp.*, Aug. 1964.
3. D. L. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2), Feb. 1981.
4. Wei Dai. Freedom attacks. *http://www.eskimo.com/ weidai/freedom-attacks.txt*, 1998.
5. George Danezis, Roger Dingledine, and Nick Mathewson. Mixminion: Design of a Type III Anonymous Remailer Protocol. In *Proceedings of the 2003 IEEE Symposium on Security and Privacy*, May 2003.
6. R. O. Duda and P. E. Hart. *Pattern Classification*. John Wiley & Sons, 2001.
7. Edward W. Felten and Michael A. Schneider. Timing attacks on web privacy. *ACM Conference on Computer and Communications Security (CCS)*, 2000.
8. Michael J. Freedman and Robert Morris. Tarzan: A peer-to-peer anonymizing network layer. In *Proceedings of the 9th ACM Conference on Computer and Communications Security (CCS 2002)*, Washington, DC, November 2002.
9. Xinwen Fu, Bryan Graham, Riccardo Bettati, and Wei Zhao. On effectiveness of link padding for statistical traffic analysis attacks. *ICDCS*, 2003.
10. Xinwen Fu, Bryan Graham, Dong Xuan, Riccardo Bettati, and Wei Zhao. Active probing attacks. *Technical Report TR2003-8-8, Texas A&M University*, August 2003.
11. Y. Guan, X. Fu, D. Xuan, P. U. Shenoy, R. Bettati, and W. Zhao. Netcamo: Camouflaging network traffic for qos-guaranteed critical allplications. In *IEEE Transactions on Systems,*

Man, and Cybernetics Part A: Systems and Humans, Special Issue on Information Assurance, volume 31 of *4*, pages 253–265, July 2001.

12. Andrew Hintz. Fingerprinting websites using traffic analysis. *http://guh.nu/projects/ta/safeweb/safeweb.html*, 2002.

13. SafeWeb inc. Safeweb. *http://www.safewebinc.com/*, 2002.

14. R. Moddemeijer. On estimation of entropy and mutual information of continuous distributions. *Signal Processing*, 16(3):233–246, 1989.

15. R. E. Newman-Wolfe and B. R. Venkatraman. High level prevention of traffic analysis. *Computer Security Applications Conference, Seventh Annual*, pages 102 –109, 1991.

16. R. E. Newman-Wolfe and B. R. Venkatraman. Performance analysis of a method for high level prevention of traffic analysis. *Computer Security Applications Conference, Eighth Annual*, pages 123 –130, 1992.

17. Marconi Corporation plc. Esr-5000 and esr-6000 enterprise switch routers. http://www.marconi.com/html/products/esr50006000.htm, 2003.

18. J. Raymond. Traffic analysis: Protocols, attacks, design issues and open problems. In H. Federrath, editor, *Designing Privacy Enhancing Technologies: Proceedings of International Workshop on Design Issues in Anonymity and Unobservability*, volume 2009 of *LNCS*, pages 10–29. Springer-Verlag, 2001.

19. Andrei Serjantov and Peter Sewell. Passive attack analysis for connection-based anonymity systems. In *European Symposium on Research in Computer Security (ESORICS)*, 2003.

20. C. E. Shannon. Communication theory of secrecy systems. *Bell Sys. Tech. J.*, 28:656–715, 1949.

21. B. W. Silverman. *Density estimation for statistics and data analysis*. Chapman and Hall, London, New York, 1986.

22. D. X. Song, D. Wagner, and X. Tian. Timing analysis of keystrokes and timing attacks on ssh. *10th USENIX Security Symposium*, 2001.

23. Qixiang Sun, Daniel R. Simon, Yimin Wang, Wilf Russell, Venkata N. Padmanabhan, and Lili Qiu. Statistical identification of encrypted web browsing traffic. *IEEE Symposium on Security and Privacy*, 2002.

24. P. F. Syverson, D. M. Goldschlag, and M. G. Reed. Anonymous connections and onion routing. In *IEEE Symposium on Security and Privacy*, pages 44–54, Oakland, California, 4–7 1997.

25. Agilent Technologies. Agilent j6841a network analyzer software. *http://onenetworks.comms.agilent.com/NetworkAnalyzer /J6841A.asp*, March 2002.

26. TimeSys. Timesys linux docs. *http://www.timesys.com/index.cfm? hdr=home_header.cfm&bdy=home_bdy_library.cfm*, 2003.

27. Brenda Timmerman. a security model for dynamic adaptive traffic masking. *New Security Paradigms Workshop*, 1997.

28. B. R. Venkatraman and R. E. Newman-Wolfe. Performance analysis of a method for high level prevention of traffic analysis using measurements from a campus network. *Computer Security Applications Conference, 10th Annual*, pages 288 –297, 1994.

29. V. Voydoc and S. Kent. Security mechanisms in high-level network protocols. *ACM Computing Surveys*, pages 135 – 171, 1983.

Appendix A

In this Appendix, we develop theory calculating the optimal bin size for histogram based entropy estimation. In [14], for a histogram based entropy estimation,

$$E\{\widetilde{H}\} \approx H - \frac{I-1}{2n} + \frac{1}{24}(\frac{\Delta x}{\sigma})^2 \tag{14}$$

where $\widetilde{H}$ is the estimated entropy, H is original entropy of random variable X, n is the sample size, Δx is the bin size and $sigma$ is the X's standard deviation.

If random variable X is normally distributed,

$$VAR\{\widetilde{H}\} \approx \frac{1}{2}n^{-1} \tag{15}$$

Moreover, for a normaly distributed random variable, since the probability of $x > 3\sigma$ is very small and ignorable, we have the following relationship between bin size Δx, the number of bins I and random variable X's standard deviation σ,

$$6\sigma = I\Delta x \tag{16}$$

We can calculate the mean square error (MSE) of the entropy estimation, $E[(H - \widetilde{H})^2]$, in (17),

$$E[(H - \widetilde{H})^2] = VAR(\widetilde{H}) + [BIAS(\hat{H})]^2 \tag{17}$$

where $BIAS[\hat{H}] = E(\hat{H}) - H$.

Substitute (14) and (15) into (17),

$$E[(H - \hat{H})^2] = \frac{1}{2}n^{-1} + \left(\frac{1}{24}(\frac{\Delta x}{\sigma})^2 - \frac{I-1}{2n}\right)^2 \tag{18}$$

To derive a minimum mean square error, the second term in (18) should be minimized,

$$\frac{1}{24}(\frac{\Delta x}{\sigma})^2 = \frac{I-1}{2n} \tag{19}$$

From (16), we have $\frac{\Delta x}{\sigma} = \frac{6}{I}$. Substitute it into 19,

$$\frac{1}{24}\frac{6^2}{I} = \frac{I-1}{2n} \tag{20}$$

Thus

$$(I-1)I^2 = 3n \tag{21}$$

$$\Delta x = \frac{6\sigma}{I} \tag{22}$$

Appendix B

In this appendix, we prove Corollary 1. We assume that the user payload traffic rate is either low (denoted as ω_l) or high (denoted as ω_h) and the rate varies periodically with each rate lasting for an interval of Δ. During Δ, M *padded* packets are sent out. The problem is: given a sample of size n $(< M)$, what is the detection rate of the user payload traffic rate?

Recall how the adversary does the rate recognition. After getting a sample Z, we calculate its feature measurement F, whose PDF is $p(s)$. Off-line he obtains the decision boundary d. For two class (ω_l, ω_h) classification problem, when $s < d$, the sample is from ω_l, otherwise from ω_h.

First let's check the elements of sample Z of size n, there are m elements, X_1, $\cdots$, X_m, from rate ω_l, whose RTTs satisfy $n(\mu_l, \sigma_l^2)$. The other $n - m$ elements, Y_1, $\cdots$, Y_{n-m}, are from rate ω_h, whose RTTs satisfy $n(\mu_h, \sigma_h^2)$. Thus, sample Z is $\{X_1, \cdots, X_m, Y_1, \cdots, Y_{n-m}\}$.

For a sample consisting of packets from both ω_l and ω_h, we classify the sample as belonging to the rate whose packets show up first in the sample. Let's calculate the error rate under this definition.

Define $\omega_{l,k}$ as the event that k elements from ω_l show up first, and $\omega_{h,k}$ as the event that k elements from ω_h show up first.

$$Pr(error) = \sum Pr(error|\omega_{l,k})Pr(\omega_{l,k}) + \sum Pr(error|\omega_{h,k})Pr(\omega_{h,k}) \quad (23)$$

where

$$Pr(error|\omega_{l,k}) = Pr(s_{\omega_{l,k}} > d) \quad (24)$$
$$Pr(error|\omega_{h,k}) = Pr(s_{\omega_{h,k}} < d) \quad (25)$$

We assume that $n < M$. We have to check different cases of $\omega_{l,k}$ and $\omega_{h,k}$. In a sample of size n $(< M)$

Cases in which the traffic rate is classfied as ω_l:
$(M - n + 1) \times \omega_{l,n}, \{\omega_{l,n-1}, \omega_{h,1}\}, \{\omega_{l,n-2}, \omega_{h,2}\}, \cdots, \{\omega_{l,1}, \omega_{h,n-1}\}$
Cases in which the traffic rate is classified as ω_h:
$(M - n + 1) \times \omega_{h,n}, \{\omega_{h,n-1}, \omega_{l,1}\}, \{\omega_{h,n-2}, \omega_{l,2}\}, \cdots, \{\omega_{h,1}, \omega_{l,n-1}\}$

Thus

$$Pr(\omega_{l,k}) = \begin{cases} \frac{M-n+1}{2M} & k = n \\ \frac{1}{2M} & 1 \leq k \leq n - 1 \end{cases} \quad (26)$$

and

$$Pr(\omega_{h,k}) = \begin{cases} \frac{M-n+1}{2M} & k = n \\ \frac{1}{2M} & 1 \leq k \leq n - 1 \end{cases} \quad (27)$$

When we don't consider the order of elements in a sample,

$$\omega_{l,k} = \omega_{h,n-k}, \text{ when } k \neq n \quad (28)$$

$$(29)$$

Thus

$$p(s_{\omega_{l,k}}) = p(s_{\omega_{h,n-k}}), \text{ when } k \neq n \quad (30)$$

Noticing this fact, we have

$$P(s_{\omega_{l,k}} > d)\frac{1}{2M} + P(s_{\omega_{h,n-k}} < d)\frac{1}{2M} = \frac{1}{2M} \tag{31}$$

Substitute (26) and (27) into (23) and reorder the elements,

$$Pr(error) = \sum Pr(error|\omega_{l,k})Pr(\omega_{l,k}) + \sum Pr(error|\omega_{h,k})Pr(\omega_{h,k}) \tag{32}$$

$$= \frac{M-n+1}{2M}(P(error|\omega_{l,n}) + P(error|\omega_{h,n})) \tag{33}$$

$$+\frac{1}{2M}\sum_{k=1}^{n-1}(P(error|\omega_{l,k}) + P(error|\omega_{h,n-k})) \tag{34}$$

$$= \frac{M-n+1}{M}P(error|\text{static user payload rate}) \tag{35}$$

$$+\frac{1}{2M}\sum_{k=1}^{n-1}(P(s_{\omega_{l,k}} > d) + P(s_{\omega_{h,n-k}} < d)) \tag{36}$$

$$= \frac{M-n+1}{M}P(error|\text{static user payload rate}) + \frac{n-1}{2M} \tag{37}$$

$$= \frac{M-n+1}{M}\epsilon + \frac{n-1}{2M} \tag{38}$$

$$\tag{39}$$

Since success rate $v = 1 - Pr(error)$, we have (13) instantly.