Detection of Information Flows

Ting He, *Member, IEEE*, and Lang Tong, *Fellow, IEEE*

***Abstract*—The detection of information flows by timing analysis is considered. Given transmission timestamps of monitored nodes, the problem is to decide whether there is an information flow through these nodes by analyzing the transmission patterns. Due to constraints that packets from an information flow need to be delivered within certain delay or the relay nodes have bounded memory, transmission patterns of an information flow are statistically different from those of independent traffic. The main result of this paper is a tight characterization of the maximum amount of chaff noise such that Chernoff-consistent detection is achievable. The direct part of the result is an explicit construction of a detector that has vanishing false alarm and miss probabilities as the sample size increases whenever the noise level is below certain threshold. Conversely, when the noise level is above this threshold, there exist means to hide the information flow such that it is indistinguishable from independent traffic. Explicit characterization of the noise threshold is provided for Poisson transmission schedules. It is also shown that while information flows can be hidden among chaff noise for a small number of hops, the rate of information flow diminishes as the number of hops increases.**

***Index Terms*—Information flow, intrusion detection and security, network flows, point processes and inference, timing analysis and timing channels.**

I. Introduction

CONSIDER a wireless ad hoc network illustrated in Fig. 1. We want to know if there is an information flow through nodes S and R. Suppose that we can only observe the node transmission timestamps[1] as shown in Fig. 2. Then, from the transmission patterns, we can probably infer that S is using R as a relay.

Manuscript received July 2, 2007; revised July 20, 2008. Current version published October 22, 2008. The work is supported in part by the National Science Foundation under Award CCF-0635070, by the U.S. Army Research Laboratory under the Collaborative Technology Alliance Program DAAD19-01-2-0011, and by the Army Research Office MURI Program under Award W911NF-08-1-0238. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation thereon. The material in this paper was presented in part at the IEEE Conference on Information Sciences and Systems, Baltimore, MD, March 2007, and at the Military Communications Conference, Washington, DC, 2006.

T. He was with Cornell University, Ithaca, NY 14853 USA. She is now with the IBM T. J. Watson Research Center, Hawthorne, NY 10532 USA (e-mail: the@us.ibm.com).

L. Tong is with the School of Electrical and Computer Engineering, Cornell University, Ithaca, NY 14853 USA (e-mail: lt35@cornell.edu).

Communicated by E. Modiano, Associate Editor for Communication Networks.

Color versions of Figures 1–4, 9–11, and 13–18 in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIT.2008.929944

[1]For example, timestamps can be measured by detecting transmissions on the channels of the monitored nodes if the nodes transmit on orthogonal channels, or deploying eavesdroppers in the vicinity of the monitored nodes equipped with energy detectors that can distinguish the monitored transmissions from the interference.

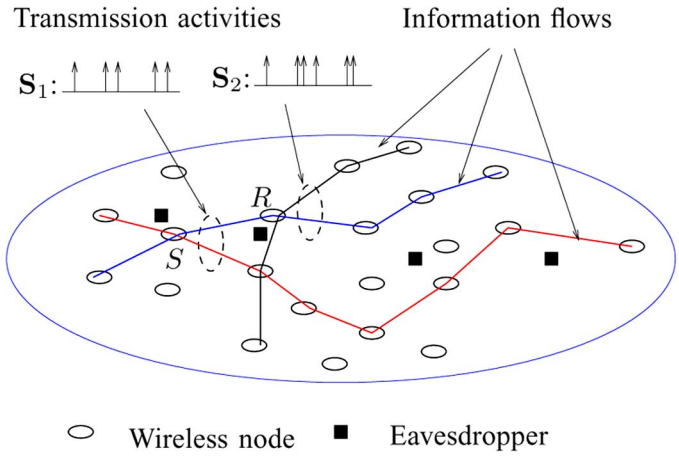

Fig. 1. Detecting information flows in a wireless ad hoc network by measuring transmission activities.

This example illustrates the problem of information flow detection. In general multihop networks such as the one in Fig. 1, multiple flows may exist simultaneously. We consider the problem of deciding whether a particular route is being used from the timing of transmission (or reception) activities. Such a problem is also broadly referred to as traffic analysis.

Direct applications of information flow detection include network surveillance and intrusion detection (e.g., the detection of stepping-stone attack [1]). Indirectly, this problem is related to traffic analysis attack in which an adversary may obtain networking information by recording the transmission patterns before launching an attack (such as denial-of-services). Thus, characterizing flow detectability is crucial for designing intrusion detection systems as well as network protocols that prevent traffic analysis attacks (see [2]).

In many applications, we may have limited knowledge about information flows. In addition, nodes carrying an information flow can manipulate transmission patterns to hide the flow. Thus, the problem of information flow detection is not tractable unless certain networking constraints are imposed. In this paper, we assume that packets that are part of an information flow are preserved by the relay nodes although packet timing may be altered. We further assume that relay nodes have a limited ability to alter the timing of packets due to delay or memory constraints. For example, we may impose a bounded delay constraint such that a relay node cannot hold a packet indefinitely and must deliver it within a fixed delay bound. Alternatively, we may impose a bounded memory constraint under which a relay node only has a finite amount of memory and thus cannot hold an infinite number of packets.

There are, in general, transmissions that are not part of an information flow, but the detector cannot distinguish them from

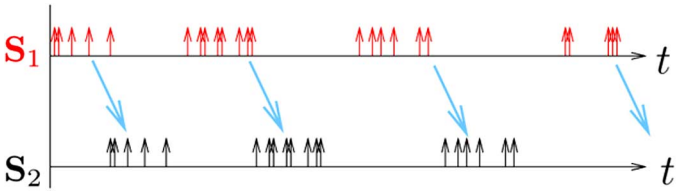

Fig. 2. Transmission patterns of S and R suggest that S is using R as a relay.

those corresponding to the information-carrying packets. We referred to these transmission timestamps as *chaff* noise. Chaff noise may come from different sources, and it may, in general, be statistically correlated with the information flows. Chaff noise may come from, for example, the multiplexing of multiple flows by the relay node. Another example is that the relay nodes, attempting to hide the information flow, may introduce dummy transmissions as chaff noise, and an intelligent relay will introduce chaff transmissions in such a way to make the detection most difficult. The presence of chaff noise imposes a serious challenge to the detection of information flows.

A. Summary of Results

The main results of this paper are the characterization of the detectability of information flows. In particular, we show that if the rate of information flow (compared with that of chaff noise) is too low, then the flow cannot be detected with arbitrarily small error probability no matter how long the detector observes the transmissions. On the other hand, if the rate of information flow is high (or the level of chaff noise is low), then the flow will be detected reliably with vanishing error probabilities. We are able to obtain a *tight* characterization of detectability for Poisson traffic in the sense that there is a single threshold on the level of chaff noise below which the flow is detectable and above which undetectable.

The main results involve two parts for both the bounded delay and the bound memory cases: the direct part that gives an explicit construction of a detector and the converse showing that there exists a way to schedule transmissions such that the traffic patterns at all relay nodes are statistically independent.

For the converse, we develop optimal schemes that embed information-carrying packets in statistically independent transmission patterns. As a result of such embedding, it is impossible to detect the information flow. Because the embedding schemes are optimal, they represent the maximum rate of undetectable information flow. They also represent upper bounds on the amount of chaff noise below which the flows are detectable. Moreover, we show that as the number of hops in a flow increases, the rate of undetectable information flow diminishes.

For the direct part, we propose detectors that are Chernoff-consistent as long as the amount of chaff noise is below certain threshold. To our best knowledge, our detectors are the first timing-based detectors that are consistent in the presence of unconstrained chaff noise, which constitutes a positive fraction of the total traffic. For Poisson traffic, we show that the thresholds for which our detectors are consistent match the upper bounds provided in the converse, and thus our characterization of flow detectability is tight.

B. Related Work and Organization

The problem cast in this paper has been considered in the context of intrusion detection. In 1995, Staniford and Heberlein [3] first considered the problem of stepping-stone detection. The key problem in stepping-stone detection is to reconstruct the intrusion path by analyzing various characteristics of the attacking traffic. Related work in the literature only considers pairwise detection.

Early detection techniques are based on the traffic content; see, e.g., [3] and [4]. To deal with encrypted traffic, timing characteristics are used in detection, such as the on–off detection by Zhang and Paxson [5], the deviation-based detection by Yoda and Etoh [6], and the packet interarrival-based detection by Wang *et al.* [7]. The drawback of these approaches is that they are vulnerable to active timing perturbation by the attacker. To address this issue, flow transformations under certain physical constraints are considered. There are two types of constraints in the literature—the bounded delay constraint proposed in [1] and the bounded memory constraint proposed in [8]. Several practical detectors have been developed under these constraints, e.g., [8]–[10]. The problem becomes much more challenging in the presence of chaff noise, with only incomplete solutions in the literature, e.g., [1] and [10]–[12]. Specifically, if chaff noise is independent of the information flows, then it was shown in [1] that there will be noticeable difference between flow-containing traffic and independent traffic; otherwise, previous detectors [10]–[12] can only tolerate a limited number of chaff packets.

We significantly advanced the state of the art by a threshold detector developed in [13] based on the estimated fraction of chaff noise, which can achieve consistent detection under a number of chaff packets that grows proportionally to the traffic size. We are also the first to consider joint detection over multiple hops, which further improves the robustness against chaff noise. As a follow-up to [13], the current paper extends the work to bounded memory flows and contains more comprehensive analysis and extensive simulation results. In [14], we performed parallel studies on distributed information flow detection under limited capacities in collecting the timing information, where a notion similar to the distortion-rate function is used to analyze the detection performance as a function of the capacity constraints.

The dual problem of information flow detection is how to randomize transmission activities to conceal maximally information flows. This is a critical problem in protecting anonymous communications against timing analysis attacks [2], [15]. In the context of wireless ad hoc networks, Hong *et al.* in [16] proposed to add random delays to prevent correlation of specific packets; at flow level, however, transmissions of nodes on the same information flow are still correlated. Zhu *et al.* in [17] proposed to make traffic on all the outgoing links of a certain node identical by inserting chaff noise. Although this approach completely hides the information flow, it is inefficient in terms of the required amount of chaff noise. More efficient methods to hide information flows have been developed based on the chaff-inserting algorithms in this paper; see [2].

The rest of this paper is organized as follows. Section II defines the problem. Section III summarizes our results on

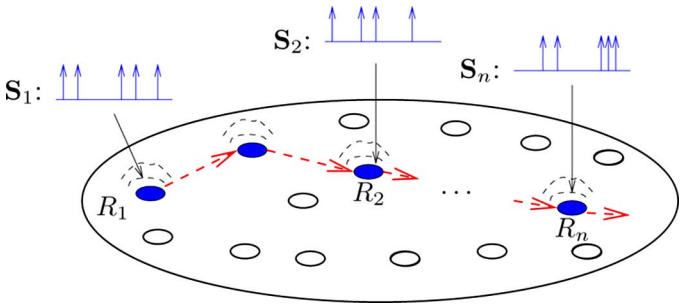

Fig. 3. Detecting information flows through nodes $R_1, R_2, \ldots, R_n$ by measuring their transmission activities; dotted lines denote a potential route.

the detectability of information flows. Sections IV and V present the optimal chaff-inserting algorithms. Section VI presents a detector based on these algorithms and analyzes its performance. The analysis is supported by simulation results in Section VII. Section VIII comments on the Poisson assumption. Then, Section IX concludes this paper with remarks on its contributions. Appendix I includes all the proofs, and Appendix II contains pseudocode implementations of all the proposed algorithms.

II. Problem Formulation

A. Notation

We introduce the following notations. Upper bold letters (e.g., $\mathbf{S}$) denote point processes and the corresponding lower bold letters (i.e., $\mathbf{s}$) their realizations. Similarly, $S(k)$ is a random variable denoting the kth timestamp of a process $\mathbf{S}$ and $s(k)$ the realization. Given two realizations of point processes $(a_1, a_2, \ldots)$ and $(b_1, b_2, \ldots)$, $\bigoplus$ is the *superposition operator* defined as $(a_k)_{k=1}^{\infty} \oplus (b_k)_{k=1}^{\infty} = (c_k)_{k=1}^{\infty}$, where $c_1 \le c_2 \le \cdots$ and $\{a_k\}_{k=1}^{\infty} \cup \{b_k\}_{k=1}^{\infty} = \{c_k\}_{k=1}^{\infty}$. Given a realization $\mathbf{s}$, we use $\mathcal{S}$ to denote the set of elements in this realization (similarly, let $\mathcal{F}$ and $\mathcal{W}$ denote the sets of elements in realizations $\mathbf{f}$ and $\mathbf{w}$).

B. Flow Models

Suppose that we are interested in detecting information flows through n ($n \ge 2$) nodes, as illustrated in Fig. 3. Let $\mathbf{S}_i$ ($i = 1, \ldots, n$) be the process of transmission timestamps of node R_i, i.e.,

$$\mathbf{S}_i = (S_i(1), S_i(2), S_i(3), \ldots), \qquad i = 1, 2, \ldots, n \tag{1}$$

where $S_i(k)$ ($k \ge 1$) is the kth transmission timestamp[2] of R_i.

If $(\mathbf{S}_i)_{i=1}^n$ contains an information flow, then it can be decomposed into an information-carrying part $(\mathbf{F}_i)_{i=1}^n$ and a chaff part $(\mathbf{W}_i)_{i=1}^n$

$$\mathbf{S}_i = \mathbf{F}_i \oplus \mathbf{W}_i, \qquad i = 1, \ldots, n \tag{2}$$

where the information-carrying part consists of packets sent by R_1 and relayed sequentially by R_i ($i = 2, \ldots, n$) as illustrated in Fig. 4. Note that chaff noise is not subject to any constraints on information flows and can be correlated with the information flows.

[2]Assume no simultaneous transmissions almost surely.

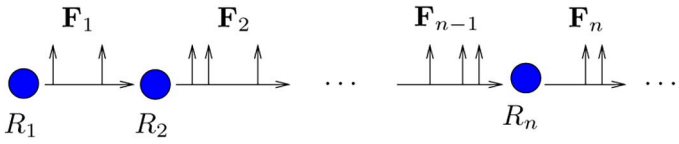

Fig. 4. Information flow along the path $R_1 \to \cdots \to R_n$.

We formally define the notion of information flow as follows.

Definition 2.1: A sequence of processes $(\mathbf{F}_1, \ldots, \mathbf{F}_n)$ is an *information flow* if for every realization $\mathbf{f}_i$ ($i = 1, \ldots, n$), there exist bijections $g_i : \mathcal{F}_i \to \mathcal{F}_{i+1}$ ($i = 1, \ldots, n-1$) such that $g_i(s) - s \ge 0$ for all $s \in \mathcal{F}_i$. For an information flow with bounded delay Δ, $g_i(s) - s \le \Delta$ for all $s \in \mathcal{F}_i$; for an information flow with bounded memory M, g_i satisfies

$$0 \le |\mathcal{F}_i \cap [0, t]| - |\mathcal{F}_{i+1} \cap [0, t]| \le M \tag{3}$$

for any $t \ge 0$.

The bijection g_i is a mapping between the transmission timestamps of the same packets at nodes R_i and R_{i+1}. The condition that g_i is a bijection ensures *packet conservation*, i.e., every information-carrying packet generates one and only one relay packet at each relay node. The condition $g_i(s) - s \ge 0$ is the *causality* constraint, which means that a packet cannot leave a node before it arrives. In addition, we consider two types of commonly encountered communication constraints imposed by the requirement of reliable communications: bounded delay constraint and bounded memory constraint. The condition $g_i(s) - s \le \Delta$ implies that the maximum delay at each relay node is uniformly bounded by Δ. This condition was first proposed by Donoho *et al.* in [1]. The condition in (3) implies that each relay node has a limited memory that can store at most M relay packets. Specifically, the condition requires the difference between the cumulative numbers of incoming packets (i.e., $|\mathcal{F}_i \cap [0, t]|$) and outgoing packets (i.e., $|\mathcal{F}_{i+1} \cap [0, t]|$) to be bounded between 0 and M at any time t (assume the flow starts at time 0). This condition was first considered in [8]. The constants Δ and M are assumed to be known.

Throughout this paper, we use a point process model for the network traffic to study fundamental aspects of information flow detection. While other networking specifics such as packet content, sizes, and addresses may also be exploited to improve detection performance, the use of these specifics will make the approach less general. Consequently, our results provide performance lower bounds under the minimum amount of available information.

C. Problem Statement

We are interested in testing the following hypotheses:

$$\begin{aligned} \mathcal{H}_0 &: \quad \mathbf{S}_1, \mathbf{S}_2, \ldots, \mathbf{S}_n \text{ are jointly independent} \\ \mathcal{H}_1 &: \quad (\mathbf{S}_i)_{i=1}^n \text{ contains an information flow} \end{aligned}$$

by observing $\mathbf{S}_i$ $(i = 1,\ldots,n)$ for some time t $(t > 0)$. No statistical assumptions are made for $\mathbf{F}_i$ and $\mathbf{W}_i$ $(i = 1,\ldots,n)$ under $\mathcal{H}_1$, but the distributions of $\mathbf{S}_i$ $(i = 1,\ldots,n)$ are assumed to be known under $\mathcal{H}_0$ (they are assumed to be Poisson processes in our analysis). We point out that although Poisson assumption is needed to obtain explicit expressions, the idea of detection is applicable for general point processes.

Remark: The above is a test of independent traffic against end-to-end information flows. Because the complement of $\mathcal{H}_0$ is not $\mathcal{H}_1$, one should view this test as part of an overall detection scheme. For example, if we observe realizations $\mathbf{s}_1,\ldots,\mathbf{s}_n$, and we want to find out whether a subset of the processes contains an information flow, we can first apply the above hypothesis testing to every pair of realizations $(\mathbf{s}_i,\mathbf{s}_j)$ $(i,j \in \{1,\ldots,n\})$ to test if this pair contains an information flow, and then if there is no detection on pairs, we extend the scope to every triple, etc. That is, we can sequentially test $\mathcal{H}_0$ versus $\mathcal{H}_1$ on every subset $(\mathbf{s}_i)_{i\in I}$ $(I \subseteq \{1,\ldots,n\})$ for $|I| = 2,\ldots,n$. This procedure helps us to simplify the detection of partial information flows, which may only go through a subset of the monitored nodes to the detection of end-to-end flows.

To characterize the amount of chaff noise, we introduce the following definition.

Definition 2.2: Given realizations of an information flow $(\mathbf{f}_i)_{i=1}^n$ and chaff noise $(\mathbf{w}_i)_{i=1}^n$, the *chaff-to-traffic ratio* (CTR) is defined as

$$\mathrm{CTR}(t) \triangleq \frac{\sum_{i=1}^n |\mathcal{W}_i \cap [0,t]|}{\sum_{i=1}^n |\mathcal{S}_i \cap [0,t]|}, \quad \mathrm{CTR} \triangleq \limsup_{t\to\infty} \mathrm{CTR}(t). \tag{4}$$

In other words, $\mathrm{CTR}(t)$ is the fraction of chaff packets in the first t period of time and CTR its asymptotic value. We are interested in the asymptotic detection performance with respect to CTR.

Because we consider a nonparametric alternative hypothesis in which distributions of $\mathbf{F}_i$ and $\mathbf{W}_i$ $(i = 1,\ldots,n)$ are unknown, we borrow the notion of Chernoff-consistency in [18] to introduce the following performance measure.

Definition 2.3: A detector δ_t is called r-consistent $(r \in [0,1])$ if it is Chernoff-consistent for all the information flows with CTR bounded by r a.s.,[3] that is, the false alarm probability $P_F(\delta_t)$ and the miss probability $P_M(\delta_t)$ satisfy the following:

1) $\lim_{t\to\infty} P_F(\delta_t) = 0$ for any $(\mathbf{S}_i)_{i=1}^n$ under $\mathcal{H}_0$;
2) $\sup_{(\mathbf{S}_i)_{i=1}^n \in \mathcal{P}} \lim_{t\to\infty} P_M(\delta_t) = 0$, where

$$\mathcal{P} = \Big\{(\mathbf{S}_i)_{i=1}^n : (\mathbf{S}_i)_{i=1}^n \text{ contains an information flow and } \limsup_{t\to\infty} \mathrm{CTR}(t) \le r \text{ a.s.}\Big\}.$$

The *consistency* of a detector is defined as the supremum of r such that the detector is r-consistent.

[3]Here a.s. means "almost surely."

III. Flow Detectability

We first give the general detectability result, starting with the following definitions.

Definition 3.1: For n-hop information flows with bounded delay Δ, the level of weak detectability, denoted by $\overline{\alpha}_n^\Delta$, is defined as

$$\overline{\alpha}_n^\Delta \triangleq \sup\Big\{r : \forall\, (\mathbf{S}_i)_{i=1}^n \text{ containing an information flow with bounded delay } \Delta,\ \text{if } \limsup_{t\to\infty} \mathrm{CTR}(t) \le r \text{ a.s.,} \\ \text{then } \exists \text{ a Chernoff-consistent detector for } (\mathbf{S}_i)_{i=1}^n\Big\}. \tag{5}$$

The level of strong detectability, denoted by $\underline{\alpha}_n^\Delta$, is defined as

$$\underline{\alpha}_n^\Delta \triangleq \sup\{r : \exists\, \delta_t \text{ s.t. } \delta_t \text{ is } r\text{-consistent}\}. \tag{6}$$

For information flows with bounded memory, the levels of weak and strong detectabilities, denoted by $\overline{\alpha}_n^M$ and $\underline{\alpha}_n^M$, are defined similarly.

By definition, the weak detectability allows the detector to depend on the distribution of information flows, whereas the strong detectability does not. Thus, the level of weak detectability is no lower than that of strong detectability, i.e., $\underline{\alpha}_n^j \le \overline{\alpha}_n^j$ $(j = \Delta, M)$.

With a sufficient amount of chaff noise, the nodes can make traffic containing an information flow mimic arbitrary traffic patterns, including the traffic patterns under $\mathcal{H}_0$. Therefore, there must be some limits on the amount of chaff noise beyond which information flows are no longer detectable. A basic limit is the amount of chaff noise sufficient to make an information flow statistically identical with independent traffic. Specifically, we define a notion of the level of undetectability as follows.

Given $\mathcal{H}_0$, define the level of undetectability as[4]

$$\beta_n^\Delta \triangleq \inf\Big\{r \in [0,1] : \exists (\mathbf{F}_i)_{i=1}^n, (\mathbf{W}_i)_{i=1}^n \text{ satisfying :} \\ 1)\ (\mathbf{F}_i \oplus \mathbf{W}_i)_{i=1}^n \overset{d}{=} (\mathbf{S}_i)_{i=1}^n \text{ for some } (\mathbf{S}_i)_{i=1}^n \text{ under } \mathcal{H}_0; \\ 2)\ (\mathbf{F}_i)_{i=1}^n \text{ is an information flow with bounded delay } \Delta; \\ 3)\ \limsup_{t\to\infty} \mathrm{CTR}(t) \le r \text{ a.s.}\Big\}. \tag{7}$$

That is, β_n^Δ is the minimum CTR for an n-hop information flow with bounded delay Δ to be equal to traffic under $\mathcal{H}_0$ in distribution. The corresponding quantity β_n^M for bounded memory flows is defined similarly.

Our main results are the following relationships among the levels of weak and strong detectabilities and the level of undetectability.

[4]Here "$\overset{d}{=}$" means equal in distribution.

Theorem 3.2: If $\mathbf{S}_i$ $(i = 1, \ldots, n)$ are Poisson processes of bounded rates under $\mathcal{H}_0$, then

$$\overline{\alpha}_n^j = \underline{\alpha}_n^j = \beta_n^j, \qquad j = \Delta, M. \tag{8}$$

Remark: This theorem states that for Poisson null hypothesis, the levels of weak and strong detectabilities are equal and equal to the minimum fraction of chaff to mimic the null hypothesis. For CTR less than β_n^j $(j = \Delta, M)$, any information flow can be detected consistently by the same detector; for CTR above or equal to β_n^j, there is a method to hide the information flow among chaff noise such that consistent detection is impossible. We will give explicit expressions for β_n^j or its bounds later.

Proof: The proof contains a converse part and an achievability part. For the converse part, we need to show that $\overline{\alpha}_n^j \leq \beta_n^j$ $(j = \Delta, M)$. By the definition of β_n^j, there exists $(\mathbf{S}_i)_{i=1}^n$ such that it contains an information flow with β_n^j fraction of chaff, and $\mathbf{S}_1, \ldots, \mathbf{S}_n$ are truly independent Poisson processes. Thus, it is impossible to have a Chernoff-consistent detector for this information flow, which implies that β_n^j is an upper bound on the level of weak detectability.

For the achievability part, we need to show that $\underline{\alpha}_n^j \geq \beta_n^j$ $(j = \Delta, M)$. The approach is to design a detector, which is r-consistent for r arbitrarily close to β_n^j. The detector is presented later in Definition 6.2 and analysis of its consistency in Theorems 6.3 and 6.4. Combining the converse and the achievability results and the fact that $\underline{\alpha}_n^j \leq \overline{\alpha}_n^j$ $(j = \Delta, M)$ gives Theorem 3.2. □

In Sections IV–VIII, we will explain how to compute β_n^j $(j = \Delta, M)$ and how to do the detection.

IV. Detectability of Two-Hop Flows

In this section, we consider two-hop information flows (i.e., $n = 2$). Given the distribution of $(\mathbf{S}_1, \mathbf{S}_2)$ under $\mathcal{H}_0$, we aim at characterizing the value of β_2^j $(j = \Delta, M)$.

Our approach is to find first the algorithms, which optimally partition $\mathbf{S}_i$ $(i = 1, 2)$ into $\mathbf{F}_i$ and $\mathbf{W}_i$ such that $(\mathbf{F}_1, \mathbf{F}_2)$ is an information flow, and the CTR is minimized, and then calculate β_2^j by analyzing the CTR of these algorithms under $\mathcal{H}_0$. Such algorithms are called *chaff-inserting algorithms*, and the CTR of these algorithms is defined as the CTR of the partitioned traffic.

A. Two-Hop Flows With Bounded Delay

Suppose that nodes R_1 and R_2 want to send a two-hop information flow with bounded delay Δ, and they are allowed to design the insertion of chaff noise. The question is how to insert the minimum amount of chaff noise such that $\mathbf{S}_1$ and $\mathbf{S}_2$ become statistically independent.

To answer this question, Blum *et al.* [10] proposed a greedy algorithm called bounded greedy match (BGM), which works as follows: given a realization $(\mathbf{s}_1, \mathbf{s}_2)$

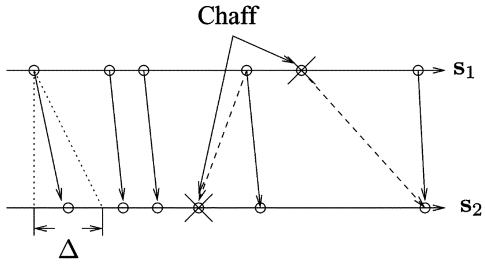

Fig. 5. BGM: a sequential greedy match algorithm that matches every packet in $\mathbf{s}_1$ with the first unmatched packet in $\mathbf{s}_2$ subject to the causality and the bounded delay constraints.

1) match every packet transmitted at time s in the first process $\mathbf{s}_1$ with the first unmatched packet transmitted in $[s, s+\Delta]$ in the second process $\mathbf{s}_2$;
2) label all the unmatched packets in $\mathbf{s}_1$ and $\mathbf{s}_2$ as chaff.

See Fig. 5 for an illustration of BGM. It is easy to see that BGM has complexity $O(|\mathcal{S}_1| + |\mathcal{S}_2|)$. For a pseudocode implementation of BGM, see Appendix II.

Algorithm BGM has been shown in [10] to be the optimal chaff-inserting algorithm for two-hop information flows with bounded delay, as stated in the following proposition.

Proposition 4.1 [10]: For any realization $(\mathbf{s}_1, \mathbf{s}_2)$, BGM inserts the minimum number of chaff packets in transmitting an information flow with bounded delay Δ.

The optimality of BGM allows us to characterize the minimum chaff needed to mimic completely independent traffic by analyzing the CTR of BGM. If, in particular, the independent traffic can be modeled as Poisson processes, then we prove the following results.

Theorem 4.2: If $\mathbf{S}_1$ and $\mathbf{S}_2$ are independent Poisson processes of rates λ_1 and λ_2, respectively, then with probability one, the CTR of BGM satisfies

$$\lim_{t\to\infty} \mathrm{CTR}_{\mathrm{BGM}}(t) = \begin{cases} \frac{(\lambda_2-\lambda_1)\left(1+\left(\frac{\lambda_1}{\lambda_2}\right)e^{\Delta(\lambda_1-\lambda_2)}\right)}{(\lambda_1+\lambda_2)\left(1-\left(\frac{\lambda_1}{\lambda_2}\right)e^{\Delta(\lambda_1-\lambda_2)}\right)}, & \text{if } \lambda_1 \neq \lambda_2 \\ \frac{1}{1+\lambda_1\Delta}, & \text{if } \lambda_1 = \lambda_2. \end{cases} \tag{9}$$

Proof: See Appendix I. □

It is easy to show that if $\lambda_i \leq \lambda$ $(i = 1, 2)$, then the CTR of BGM is lower bounded by $1/(1 + \lambda\Delta)$. By the optimality of BGM, we see that the following result holds.

Corollary 4.3: If under $\mathcal{H}_0$, $\mathbf{S}_1$ and $\mathbf{S}_2$ are independent Poisson processes with maximum rate λ, then the level of undetectability $\beta_2^\Delta = 1/(1 + \lambda\Delta)$.

With $1/(1 + \lambda\Delta)$ fraction of chaff noise, the two-hop traffic containing an information flow with bounded delay can be made identical with traffic under $\mathcal{H}_0$ so that no detector can detect this flow consistently. Note that as $\lambda\Delta \to \infty$, the value of β_2^Δ will decrease to zero, implying that it is easy to mimic $\mathcal{H}_0$ if the traffic load is heavy (large λ) or the delay bound is loose (large Δ).

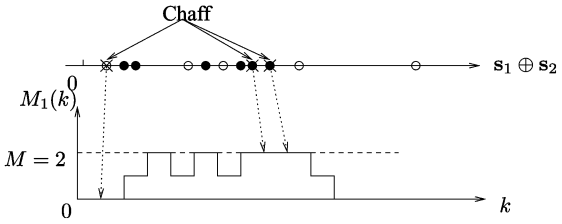

Fig. 6. Example. •: $s_k \in \mathcal{S}_1$; ∘: $s_k \in \mathcal{S}_2$; $M_1(k)$: the statistics calculated by BMR. Initially, $M_1(0) = 0$, indicating that the memory is empty. The first packet is a departure, and it is assigned as chaff because otherwise the memory will be underflowed. The second packet is an arrival, and thus the memory usage is increased by one. Such updating occurs at each arrival or departure.

B. Two-Hop Flows With Bounded Memory

Consider the transmission of a two-hop information flow with bounded memory M. We want to find a method that schedules transmissions according to independent traffic while inserting the minimum amount of chaff noise.

The bounded memory constraint requires that the number of relay packets stored at the relay node is always bounded between 0 and M. Thus, a feasible scheduling is to keep updating the memory usage for each arrival (i.e., a packet in $\mathbf{S}_1$) or departure (i.e., a packet in $\mathbf{S}_2$), and assign that packet to be chaff if the memory is overflowed or underflowed. Based on this idea, we develop a chaff-inserting algorithm called bounded memory relay (BMR). Given a realization $(\mathbf{s}_1, \mathbf{s}_2)$ and $(s_k)_{k=1}^{\infty} \triangleq \mathbf{s}_1 \oplus \mathbf{s}_2$, let $M_1(k)$ be the number of stored packets after the transmission of the kth packet in $\mathbf{s}_1 \oplus \mathbf{s}_2$. Algorithm BMR does the following. For $k = 1, 2, \ldots$

1) label a packet s_k as chaff if and only if this packet will cause a memory overflow, i.e., $s_k \in \mathcal{S}_1$ and $M_1(k-1) = M$, or underflow, i.e., $s_k \in \mathcal{S}_2$ and $M_1(k-1) = 0$; initially, $M_1(0) = 0$;
2) compute $M_1(k)$ by[5]

$$M_1(k) = \begin{cases} M_1(k-1), & \text{if } s_k = \text{chaff} \\ M_1(k-1) + I_{\{s_k \in \mathcal{S}_1\}} - I_{\{s_k \in \mathcal{S}_2\}}, & \text{o.w.} \end{cases}$$

A sample path of $M_1(k)$ $(k \geq 1)$ is shown in Fig. 6.

The complexity of BMR is $O(|\mathcal{S}_1| + |\mathcal{S}_2|)$. See Appendix II for an implementation of BMR. Note that unlike BGM, BMR does not specify the mapping between packets in the two processes because as long as the memory constraint is satisfied, the order of transmission is irrelevant.

The optimality of BMR is guaranteed by the following proposition.

Proposition 4.4: For any realization $(\mathbf{s}_1, \mathbf{s}_2)$, BMR inserts the minimum number of chaff packets in transmitting an information flow with bounded memory M.

Proof: See Appendix I. □

Because BMR is optimal, we can characterize β_2^M by the CTR of BMR, as stated in the following theorem.

[5]Here $I_{\{\cdot\}}$ is the indicator function.

Theorem 5.5: If $\mathbf{S}_1$ and $\mathbf{S}_2$ are independent Poisson processes of rates λ_1 and λ_2, respectively, then with probability one, the CTR of BMR satisfies

$$\lim_{t\to\infty} \mathrm{CTR}_{\mathrm{BMR}}(t) = \begin{cases} \dfrac{(\lambda_2 - \lambda_1)\left(1 + \left(\frac{\lambda_1}{\lambda_2}\right)^{M+1}\right)}{(\lambda_1 + \lambda_2)\left(1 - \left(\frac{\lambda_1}{\lambda_2}\right)^{M+1}\right)}, & \text{if } \lambda_1 \neq \lambda_2 \\ \dfrac{1}{1+M}, & \text{if } \lambda_1 = \lambda_2. \end{cases} \tag{10}$$

Proof: See Appendix I. □

It can be shown that the CTR is minimized when $\lambda_1 = \lambda_2$, based on which we have the following result.

Corollary 4.6: If under $\mathcal{H}_0$, $\mathbf{S}_1$ and $\mathbf{S}_2$ are independent Poisson processes, then the level of undetectability $\beta_2^M = 1/(1+M)$.

If nodes can insert at least $1/(1+M)$ fraction of chaff noise, then BMR gives a feasible transmission schedule for an information flow with bounded memory such that the overall traffic is statistically the same as traffic under $\mathcal{H}_0$. Therefore, $1/(1+M)$ establishes a limit on the maximum amount of chaff noise under the requirement of Chernoff-consistent detection. If $M \gg 1$, then very little chaff noise suffices to hide the information flows.

V. Detectability of Multihop Flows

The results in Section IV suggest that pairwise detection of information flows is vulnerable to chaff noise because a relatively small amount of chaff noise can make the information flow undetectable. These results indeed reveal the weakness of pairwise detection. As the number of hops increases, however, we see that the constraints imposed on information-carrying packets become tighter because only the packets satisfying the constraints at every hop can successfully reach the destination. This observation motivates us to extend the results in Section IV to information flows over multiple hops. Specifically, we will show that the fraction of chaff noise needed to make a multihop information flow mimic jointly independent traffic increases to one as the number of hops increases, which implies that joint detection may significantly improve the performance against chaff noise.

A. Multihop Flows With Bounded Delay

Consider the transmission of an n-hop $(n \geq 2)$ information flow with bounded delay Δ according to certain processes. Given a sequence of processes $(\mathbf{S}_i)_{i=1}^n$, we want to decompose $\mathbf{S}_i$ $(i = 1, \ldots, n)$ into $\mathbf{F}_i$ and $\mathbf{W}_i$ such that $(\mathbf{F}_i)_{i=1}^n$ is an information flow with bounded delay, and the CTR is minimized.

Given the two-hop chaff-inserting algorithm BGM, one might think that we can sequentially apply BGM to every pair of processes to obtain $(\mathbf{F}_i)_{i=1}^n$. Such an approach, however, does not give the optimal decomposition. For example, consider the realizations shown in Fig. 7. If we use BGM to match packets in $\mathbf{s}_1$ and $\mathbf{s}_2$, and then repeat BGM to match the matched packets in $\mathbf{s}_2$ with $\mathbf{s}_3$, we only find one sequence of matched packets [as shown in Fig. 7(a)]. There is, however, another way of matching that gives two sequences of matched packets [as shown in Fig. 7(b)]. The implication is that for

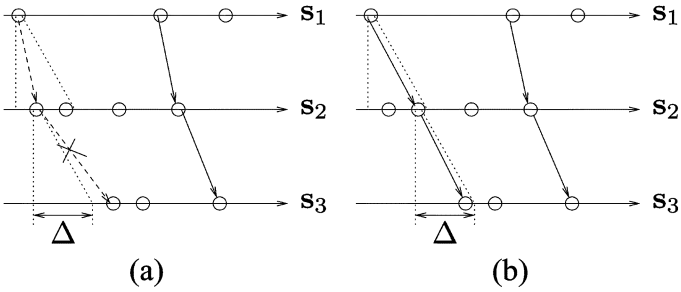

Fig. 7. Example. (a) The scheduling obtained by repeatedly using BGM. (b) Another scheduling. It shows that repeatedly using BGM is suboptimal.

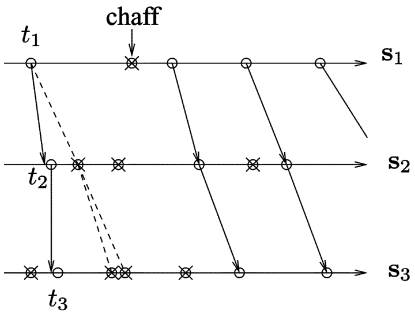

Fig. 8. MBDR: a recursive greedy match algorithm that matches every upstream packet with the first downstream packet (subject to the causality and the delay constraints), conditioned on that the downstream packet has a matching in its own downstream.

$n > 2$, a hop-by-hop greedy match is not sufficient. We have to jointly consider all the subsequent hops to find the optimal packet matching.

To solve this problem, we develop an algorithm called multi-bounded delay relay (MBDR). The idea of MBDR is that a packet at time t_1 in $\mathbf{s}_1$ can be matched with a packet at $t_2 \in [t_1, t_1+\Delta]$ in $\mathbf{s}_2$ only if t_2 has matched packets in $\mathbf{s}_i$ for all $i = 3, \ldots, n$. The matching of t_2 and its matched packets is done by recursions. Such recursions allow us to consider all the processes simultaneously and achieve a smaller CTR than repeatedly applying BGM. Specifically, MBDR works as follows. Given a realization $(\mathbf{s}_i)_{i=1}^n$:

1) match every packet at time t_1 in $\mathbf{s}_1$ with the first unmatched packet t_2 in $[t_1, t_1+\Delta]$ in $\mathbf{s}_2$, conditioned on that t_2 has a match in $\mathbf{s}_3$;
2) for $i = 2, \ldots, n-1$, match the packet t_i in $\mathbf{s}_i$ with the first unmatched packet t_{i+1} in $[t_i, t_i+\Delta]$ in $\mathbf{s}_{i+1}$, conditioned on that t_{i+1} has a match in $\mathbf{s}_{i+2}$ (assume every packet in $\mathbf{s}_n$ has a match);
3) after trying to match all the packets in $\mathbf{s}_1$, label all the unmatched packets as chaff.

For example, consider the three-hop information flow illustrated in Fig. 8. To match $t_1 \in \mathcal{S}_1$, MBDR first tries to find a match for t_2. Because t_2 can be matched with $t_3 \in \mathcal{S}_3$, t_1 is matched with t_2. If t_2 does not have a match in $\mathbf{s}_3$, MBDR will try to match t_1 with the next unmatched packet in $[t_1, t_1+\Delta]$ in $\mathbf{s}_2$. If there is no more packet left, MBDR will label t_1 as chaff.

A direct implementation of MBDR has complexity $O((\lambda\Delta)^n|\mathcal{S}_1|)$, where λ is the maximum rate of $\mathbf{S}_1, \ldots, \mathbf{S}_n$. The complexity can be reduced to $O\left(n^2|\mathcal{S}_1|\right)$ by expanding the recursions (see Appendix II). Note that MBDR is reduced to BGM when $n = 2$.

It is easy to verify that if we transmit information-carrying packets according to the matching found by MBDR, the transmissions will satisfy the bounded delay constraint at every hop. Moreover, such a transmission schedule preserves the order of incoming packets. The following proposition states that MBDR is optimal.

Proposition 5.1: For any realization $(\mathbf{s}_i)_{i=1}^n$, MBDR inserts the minimum number of chaff packets in transmitting an n-hop information flow with bounded delay Δ.

Proof: See Appendix I. □

By arguments similar to those in the proof of Theorem 4.2, one can show that the CTR of MBDR converges a.s. It is difficult to compute the exact limit.[6] Instead, we give the following bound.

Theorem 5.2: If $\mathbf{S}_i$ $(i = 1, \ldots, n)$ are independent Poisson processes of maximum rate λ, then

$$\lim_{t\to\infty} \mathrm{CTR}_{\mathrm{MBDR}}(t) \geq 1 - \kappa_n \text{ a.s.} \tag{11}$$

where

$$\kappa_n = \min\left((\lambda\Delta)^{n-2}(1-e^{-\lambda\Delta}), \prod_{i=1}^{n-1}(1-e^{-i\lambda\Delta})\right). \tag{12}$$

Proof: See Appendix I. □

By Theorem 5.2, we see that the CTR of MBDR goes to one exponentially with the increase of n if $\lambda\Delta < 1$. It can be shown that if we repeatedly apply BGM, then the CTR is lower bounded by $1-(1-e^{-\lambda\Delta})^{n-1}$ a.s., which always converges to one exponentially.[7]

Although in Definition 2.1 we have assumed identical delay bounds at all the relay nodes, MBDR can be easily extended to different delay bounds, and κ_n in Theorem 5.2 becomes

$$\min\left((1-e^{-\lambda\Delta_{n-1}})\prod_{i=1}^{n-2}(\lambda\Delta_i), \prod_{i=1}^{n-1}(1-e^{-i\lambda\Delta_i})\right)$$

where Δ_i is the maximum delay at the ith relay node.

The optimality of MBDR allows us to have the following result.

Corollary 5.3: If under $\mathcal{H}_0$, $\mathbf{S}_1, \ldots, \mathbf{S}_n$ are independent Poisson processes of rates bounded by λ, then $\beta_n^\Delta \geq 1-\kappa_n$.

By this result, we see that for sufficiently light traffic or small delay bound (i.e., $\lambda\Delta < 1$), β_n^Δ converges to one exponentially fast as n increases. Numerical calculation shows that β_n^Δ still converges to one for $\lambda\Delta > 1$, but the convergence is slower than exponential. If we calculate the maximum rate of the information flow by $\lambda(1-\beta_n^\Delta)$, then this rate will go to zero with the increase of n, implying that it is almost impossible to hide

[6]For example, for independent Poisson processes, computing the CTR of MBDR involves computing the limiting distribution of an $(n-1)$-dimensional continuous state space Markov process.

[7]It was claimed in [19] and [13] that the CTR of MBDR is lower bounded by $1-(1-e^{-\lambda\Delta})^{n-1}$ a.s. The claim is not correct, and the lower bound only holds for repeat application of BGM.

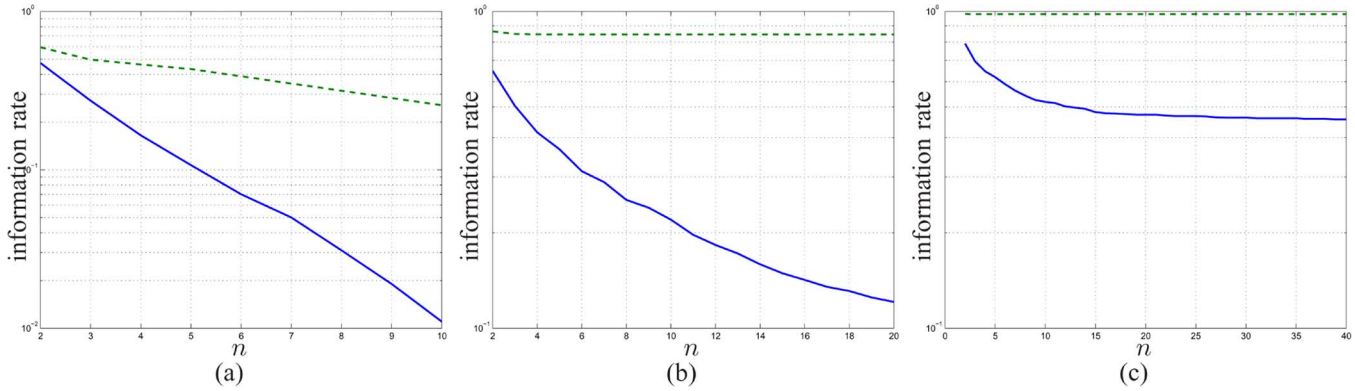

Fig. 9. Normalized rate of information flow $1-\beta_n^{\Delta}$ with respect to n ($\Delta=1$, solid line: $1-\beta_n^{\Delta}$ computed for 1000 packets per process; dashed line: κ_n): (a) $\lambda=0.9$; (b) $\lambda=2$; and (c) $\lambda=4$.

information flows over arbitrarily long paths. See Fig. 9 for numerically computed normalized information rate $1-\beta_n^{\Delta}$ as a function of n. From the plots, it is clear that the information rate decays exponentially at $\lambda\Delta<1$ [Fig. 9(a)] and subexponentially at $\lambda\Delta>1$ [Fig. 9(b) and (c)]. Moreover, the plot shows that κ_n is a loose upper bound on the normalized information rate, which is as expected because it is derived by union bounds (see the proof for details).

B. Multihop Flows With Bounded Memory

Suppose that we want to transmit an n-hop information flow with bounded memory M according to certain processes. We generalize BMR to an algorithm called multibounded memory relay (MBMR) to insert chaff noise in this case.

Algorithm MBMR borrows the idea of monitoring memory usage in BMR. Specifically, let $M_i(k)$ $(i=1,\ldots,n-1)$ denote the number of stored packets at R_{i+1} after the kth packet in the total traffic. Algorithm MBMR keeps updating $(M_i(k))_{i=1}^{n-1}$ for $k=1,2,\ldots$ and assigns chaff packets if memory underflow or overflow occurs. Given a realization $(\mathbf{s}_i)_{i=1}^{n}$ and $(s_k)_{k=1}^{\infty}\triangleq \mathbf{s}_1\oplus\cdots\oplus\mathbf{s}_n$, MBMR works as follows. For $k=1,2,\ldots$

1) label $s_k\in\mathcal{S}_i$ as chaff if and only if $M_{i-1}(k-1)=0$ or $M_i(k-1)=M$ (initially, $M_j(0)=0$ for $j=1,\ldots,n-1$; $M_0(0)=\infty$; $M_n(0)=-\infty$);
2) compute $(M_j(k))_{j=1}^{n-1}$ by

$$M_{i-1}(k)=\begin{cases}M_{i-1}(k-1)-1, & \text{if } s_k\neq\text{chaff}\\ M_{i-1}(k-1), & \text{o.w.}\end{cases}$$

$$M_i(k)=\begin{cases}M_i(k-1)+1, & \text{if } s_k\neq\text{chaff}\\ M_i(k-1), & \text{o.w.}\end{cases}$$

and $M_j(k)=M_j(k-1)$ for $j=1,\ldots,i-2,i+1,\ldots,n-1$.

See Fig. 10 for an example of MBMR.

Algorithm MBMR has complexity $O\left(\sum_{i=1}^{n}|\mathcal{S}_i|\right)$. See Appendix II for its implementation. Note that MBMR is reduced to BMR when $n=2$. If we sequentially match the nonchaff packets found by MBMR, then we will have a transmission schedule that satisfies the bounded memory constraint. The optimality of MBMR is provided by the following proposition.

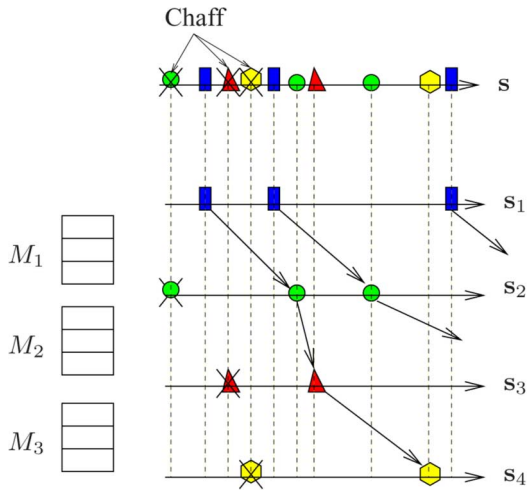

Fig. 10. MBMR for $n=4$ and $M=3$ ($\mathbf{s}=\mathbf{s}_1\oplus\cdots\oplus\mathbf{s}_4$): monitor the memory usage of the relay nodes and assign a chaff packet if the memory of any node will be underflowed or overflowed. Initially, $M_i(0)=0$ $(i=1,2,3)$; at the end of this realization (after the tenth packet), $(M_1(10),M_2(10),M_3(10))=(1,1,0)$.

Proposition 5.4: For any realization $(\mathbf{s}_i)_{i=1}^{n}$, MBMR inserts the minimum number of chaff packets to schedule the transmission of an n-hop information flow with bounded memory M.

Proof: The proof follows the same arguments as in the proof of Proposition 4.4. □

We can now characterize β_n^M by the CTR of MBMR. If $\mathbf{S}_1,\ldots,\mathbf{S}_n$ are independent Poisson processes, then the CTR of MBMR converges almost surely, and the limit can be calculated by the limiting distribution of a Markov chain, as shown in Appendix I-G.

It is difficult to give a closed-form expression for the exact CTR of MBMR. Alternatively, we derive the following upper and lower bounds. Let

$$\mathcal{A}\triangleq\{(\mathbf{S}_i)_{i=1}^{n}:\mathbf{S}_1,\ldots,\mathbf{S}_n \text{ are independent Poisson processes}\}.$$

We have the following theorem.

Theorem 5.5: For any $(\mathbf{S}_i)_{i=1}^{n}\in\mathcal{A}$

$$\lim_{t\to\infty}\text{CTR}_{\text{MBMR}}(t)\geq 1-u_n \text{ a.s.} \tag{13}$$

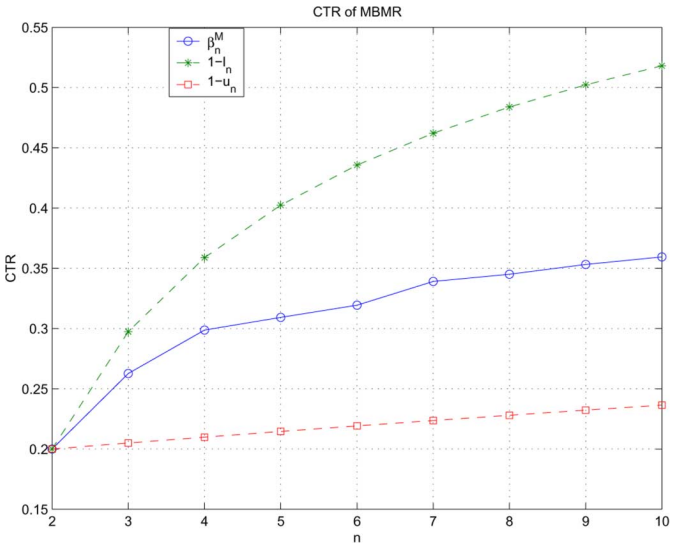

Fig. 11. Level of undetectability β_n^M and its bounds as functions of n: $M = 4$; compute β_n^M on 10 000 packets.

Furthermore

$$\inf_{\mathcal{A}} \lim_{t\to\infty} \mathrm{CTR}_{\mathrm{MBMR}}(t) \le 1 - l_n \text{ a.s.} \tag{14}$$

Here l_n and u_n are given by

$$l_{n+1} = \frac{l_n\left(1 - l_n^M\right)}{1 - l_n^{M+1}} \tag{15}$$

and

$$u_{n+1} = u_n\left(1 - \frac{1}{M+1}2^{-M/u_n}\right) \tag{16}$$

for $n \ge 2$, and $l_2 = u_2 = M/(M+1)$.

Proof: See Appendix I. □

Although identical memory constraints have been assumed in Definition 2.1, MBMR can be easily modified to allow different memory constraints, and it can be shown that the CTR is bounded between $1 - u_n'$ and $1 - l_n'$, where

$$l_{n+1}' = \frac{l_n'\left(1 - {l_n'}^{K_n}\right)}{1 - {l_n'}^{K_n+1}}$$

and

$$u_{n+1}' = u_n'\left(1 - \frac{1}{K_n+1}2^{-K_n/u_n'}\right)$$

for $n \ge 2$, and $l_2' = u_2' = K_1/(K_1+1)$. Here K_i $(i = 1,\ldots,n-1)$ is the memory constraint at the ith relay node.

Based on Theorem 5.5 and the optimality of MBMR, we have the following result.

Corollary 5.6: If under $\mathcal{H}_0$, $\mathbf{S}_1,\ldots,\mathbf{S}_n$ are independent Poisson processes, then $1 - u_n \le \beta_n^M \le 1 - l_n$.

The bounds in Corollary 5.6 are not far from the actual value of β_n^M at small n; see the numerical results in Fig. 11.

Another interpretation of Corollary 5.6 is that the normalized maximum rate of information flow calculated by $1 - \beta_n^M$ is bounded between l_n and u_n. Numerical calculation shows that l_n and u_n both decay polynomially. Specifically, l_n decays at approximately $\Theta(n^{-1/M})$ and u_n at $\Theta(n^{-1/(2M-2)})$. Furthermore, numerical comparison shows that if $\lambda\Delta = M$, β_n^M increases slower than β_n^Δ as $n \to \infty$, suggesting that it is relatively easier to hide information flows with bounded memory.

TABLE I
LEVELS OF UNDETECTABILITIES (POISSON NULL HYPOTHESIS)

β_2^Δ	$=$	$\frac{1}{1+\lambda\Delta}$
β_2^M	$=$	$\frac{1}{1+M}$
β_n^Δ	$\ge$	$1-\kappa_n$
$1-u_n \le$	β_n^M	$\le 1-l_n$

VI. DETECTOR

In Sections IV and V, we have characterized the levels of undetectabilities for information flows with bounded delay or bounded memory. The results are summarized in Table I. These results provide upper bounds on the level of detectability.

In this section, we will present an explicit detector whose consistency can approximate β_n^j $(j = \Delta, M)$ arbitrarily. Our main theorem is stated as follows.

Theorem 6.1: For any $\epsilon > 0$, there exists a detector such that its consistency is no smaller than $\beta_n^j - \epsilon$ $(j = \Delta, M)$.

Remark: The theorem states that as $\epsilon \to 0$, there exists a sequence of detectors with consistency approaching β_n^j $(j = \Delta, M)$. Therefore, the level of strong detectability is no smaller than β_n^j, i.e., $\underline{\alpha}_n^j \ge \beta_n^j$ $(j = \Delta, M)$.

The rest of this section is dedicated to the proof of Theorem 6.1. The proof is by constructing a detector and showing that its consistency approximates β_n^j $(j = \Delta, M)$ arbitrarily. Ideally, we would like to know what strategy is used to perturb timing and insert chaff noise so that we can design a detector accordingly. The difficulty here is that we do not know what strategy is going to be used when information flows are transmitted, and therefore, our goal is to design a single detector that has good performance for a wide variety of information flows.

The key idea is to design the detector based on the amount of chaff noise needed by the optimal chaff-inserting algorithms. If the detector is designed to guarantee that even the optimal algorithms need a sufficiently large amount of chaff to evade detection, then any other chaff-inserting algorithm would have to insert no less chaff noise to evade detection. Therefore, we can make sure that the detector is r-consistent against fractions of chaff up to a certain level. Specifically, we propose the following detector.

Definition 6.2: Given observations[8] $(\mathbf{s}_i)_{i=1}^n$ $(n \ge 2)$, the detector is defined as

$$\delta_t\left((\mathbf{s}_i)_{i=1}^n \tau_n\right) = \begin{cases} 1, & \text{if } \widehat{\mathrm{CTR}}(t) \le \tau_n \\ 0, & \text{o.w.} \end{cases} \tag{17}$$

where τ_n is a predetermined threshold, and $\widehat{\mathrm{CTR}}(t)$ is the minimum fraction of chaff in the measurements.

[8]To be precise, the detector is only given the part of $\mathbf{s}_i$ $(i = 1,\ldots,n)$ that falls into the length-t observation interval.

Remark: The statistic $\widehat{\mathrm{CTR}}(t)$ is computed by the optimal chaff-inserting algorithm followed by certain adjustments. Specifically, it is calculated by the following procedure.

1) Compute $\mathcal{C}$, the set of chaff packets found by the optimal chaff-inserting algorithm (MBDR for bounded delay flows or MBMR for bounded memory flows).
2) Calculate a number C by

$$C = \left| \mathcal{C} \setminus \left(\bigcup_{i=1}^{n} \mathcal{S}_i \cap [0, (i-1)\Delta) \right) \right| \tag{18}$$

for bounded delay flows, or

$$C = |\mathcal{C}| + \min_{0 \le k \le w^*} d(k) \tag{19}$$

for bounded memory flows, where $d(k)$ is the cumulative difference defined as

$$d(k) \triangleq \sum_{j=1}^{k} \left(I_{\{s_j \in \mathcal{S}_1\}} - I_{\{s_j \in \mathcal{S}_2\}} \right) \tag{20}$$

$d(0) = 0$, and w^* is the first time that $d(k)$ varies by M, i.e., $w^* \triangleq \inf\{w : \max_{0 \le k \le w} d(k) - \min_{0 \le k \le w} d(k) = M\}$.

3) Compute $\widehat{\mathrm{CTR}}(t) = C/N$, where $N = \sum_{i=1}^{n} |\mathcal{S}_i|$.

For implementation details, we refer to Appendix II. We point out that for large N, the influence of the adjustment in step (2) on $\widehat{\mathrm{CTR}}(t)$ is negligible.

The reason for $\widehat{\mathrm{CTR}}(t)$ to be the minimum fraction of chaff in the measurements hinges on two facts. The first is the optimality of the chaff-inserting algorithm used to find $\mathcal{C}$. The second is the adjustment in step (2). The adjustment is needed because the detector may not observe the beginning of the information flow. At the time the detector starts, there may have been packets stored at the relay nodes, and when these packets are relayed, the relay packets may appear to be chaff noise from the detector's perspective because they do not correspond to any observed packets. We solve this problem by ignoring certain chaff packets found at the beginning of the measurements. For bounded delay flows, these are the packets in $[0, (i-1)\Delta)$ in $\mathbf{s}_i$ $(i = 1, \ldots, n)$. For bounded memory flows, these are the packets that may be relays of packets stored in the memory initially. Detailed explanations can be found in Appendix II.

Now that $\widehat{\mathrm{CTR}}(t)$ is the minimum CTR in the measurements, we can guarantee detection as follows.

Theorem 6.3: The detector in Definition 6.2 has vanishing miss probability for all the information flows with CTR bounded by τ_n a.s.

Theorem 6.3 is a direct implication of the fact that $\widehat{\mathrm{CTR}}(t)$ is the minimum fraction of chaff packets in the measurements. Actually, a stronger statement holds, which is that the detector has no miss detection for all realizations of information flows with no more than τ_n fraction of chaff packets.

The threshold value needs to be carefully chosen such that the detector satisfies certain false alarm constraint. Specifically, under the assumption that $\mathbf{S}_i$'s are independent Poisson processes of maximum rate λ under $\mathcal{H}_0$, we have the following theorems on the false alarm probabilities.

Theorem 6.4: If $\tau_n < \beta_n^j$ $(j = \Delta, M)$, then the false alarm probability satisfies

$$\lim_{N \to \infty} \frac{1}{N} \log P_F(\delta_t) \le -\Gamma_n(\tau_n; \lambda, \Delta) < 0 \tag{21}$$

for bounded delay flows, and

$$\lim_{N \to \infty} \frac{1}{N} \log P_F(\delta_t) \le -\Gamma_n(\tau_n; M) < 0 \tag{22}$$

for bounded memory flows, where $N = \sum_{i=1}^{n} |\mathcal{S}_i|$.

Proof: See Appendix I. □

The theorem states that the false alarm probability of the proposed detector decays exponentially as long as the threshold is less than β_n^j $(j = \Delta, M)$. The functions $\Gamma_n(\tau_n; \lambda, \Delta)$ and $\Gamma_n(\tau_n; M)$ give lower bounds on the error exponents; see the proof for their definitions. We point out that $\Gamma_n(\tau_n; \lambda, \Delta)$ and $\Gamma_n(\tau_n; M)$ are positive for all $\tau_n < \beta_n^j$ $(j = \Delta, M)$, and they are both decreasing functions of τ_n.

Combining Theorems 6.3 and 6.4 yields the following result.

Corollary 6.5: If $\tau_n < \beta_n^j$ $(j = \Delta, M)$, then τ_n is the consistency of the proposed detector.

Remark: As $\tau_n \to \beta_n^j$, the consistency of the proposed detector converges to β_n^j, which proves that the level of strong detectability is lower bounded by β_n^j. From Corollary 6.5, we see that the proposed detector is optimal in terms of consistency. In particular, because $\beta_n^j \to 1$ as n increases, the proposed detector can detect almost all the long-lasting information flows with sufficiently long paths.

The threshold τ_n represents a tradeoff between the consistency and the false alarm probability. A larger τ_n enables consistent detection against more chaff noise at the cost of a higher false alarm probability, whereas a smaller τ_n leads to a smaller false alarm probability but less consistency against chaff noise.

VII. Simulations

In this section, we simulate the proposed detector on synthetic Poisson traffic. Under $\mathcal{H}_0$, $(\mathbf{S}_1, \ldots, \mathbf{S}_n)$ is a sequence of independent Poisson processes of rate λ. Under $\mathcal{H}_1$, it is the mixture of an information flow $(\mathbf{F}_1, \ldots, \mathbf{F}_n)$ of rate $(1 - f_c)\lambda$ [for some $f_c \in (0, 1)$] and chaff traffic $(\mathbf{W}_1, \ldots, \mathbf{W}_n)$, where $\mathbf{W}_i$ $(i = 1, \ldots, n)$ are independent Poisson processes of rate $f_c\lambda$. Here the parameter f_c is the CTR.

The process $\mathbf{F}_1$ is a Poisson process of rate $(1 - f_c)\lambda$, and its relays $\mathbf{F}_i$ $(i > 1)$ are generated as follows. For information flows with bounded delay

$$\mathbf{F}_i = \mathrm{sort}(F_{i-1}(1) + D_1, F_{i-1}(2) + D_2, \ldots), \qquad i > 1$$

where $\mathbf{F}_{i-1} = (F_{i-1}(1), F_{i-1}(2), \ldots)$, and $D_1, D_2, \ldots$ are independent identically distributed (i.i.d.) delays uniformly distributed in $[0, \Delta]$. For information flows with bounded memory, we partition the timestamps of $\mathbf{F}_{i-1}$ into groups of size $\lfloor M/2 \rfloor$, where the jth group is

$$(F_{i-1}((j-1)\lfloor M/2 \rfloor), \ldots, F_{i-1}(j\lfloor M/2 \rfloor - 1)).$$

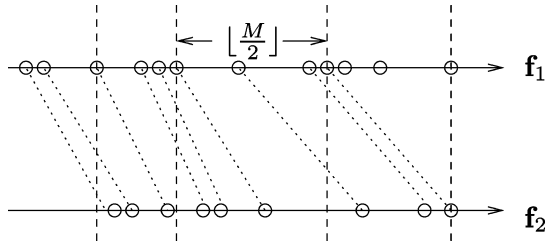

Fig. 12. Generating information flows with bounded memory $\left(\lfloor \frac{M}{2} \rfloor = 3\right)$: $\mathbf{f}_2$ is generated by storing $\lfloor \frac{M}{2} \rfloor$ packets from $\mathbf{f}_1$ and randomly releasing these packets during the arrival of the next $\lfloor \frac{M}{2} \rfloor$ packets.

TABLE II
SIMULATION PARAMETERS

n	the number of processes
λ	the rate of $\mathbf{S}_i$ $(i = 1, \ldots, n)$
Δ	maximum delay
M	maximum memory size
f_c	CTR

Then, $\mathbf{F}_i$ is generated by selecting $\lfloor M/2 \rfloor$ timestamps independently and uniformly from the interval $[F_{i-1}((j-1)\lfloor M/2 \rfloor), F_{i-1}(j\lfloor M/2 \rfloor))$ for each $j \geq 2$. As illustrated in Fig. 12, if we match timestamps in the generated realizations $\mathbf{f}_{i-1}$ and $\mathbf{f}_i$ $(i \geq 2)$ sequentially, then the matching satisfies the bounded memory constraint.

Explanations of the parameters used in the simulation are summarized in Table II. We are mainly interested in the influence of changing n on the detection performance. Because it can be shown that increasing n has opposite effects on the false alarm and the miss probabilities, we plot the receiver operating characteristics (ROCs) [20] for different n.

We first fix the sample size per process and vary the threshold to plot the ROCs for bounded delay flows and bounded memory flows; see Figs. 13 and 14. The threshold varies in $[0, 0.4]$ in Fig. 13 and $[0, 0.25]$ in Fig. 14. From the plots, we see that the ROCs approach the upper left corner as n increases, implying that the detector has better performance as the number of processes increases. This is as expected because as n increases, the detector has more observations, and thus the detection performance should be improved.

We then fix the total sample size and plot the ROCs for bounded delay flows and bounded memory flows; see Figs. 15 and 16. The threshold varies in $[0, 0.3]$ in Fig. 15 and $[0, 0.15]$ in Fig. 16. The question we want to answer is whether given the total sample size, one should take samples at two nodes or multiple nodes. As illustrated in Fig. 15, given a total sample size of 200 packets, the ROC for taking samples at four nodes is an outer bound of the ROCs for two and six nodes. Similar observation can be obtained from Fig. 16. Intuitively, this is because as n increases, the sample size per process decreases, but the constraints on information flows become tighter. These contradictory effects suggest that if the total sample size is constrained, there is an optimal n such that taking samples at n nodes optimizes the detection performance. Note that here we have assumed the flow path to be sufficiently long. Moreover, note that the level of detectability at $n = 2$ is equal to 0.2 for both bounded delay and bounded memory flows.

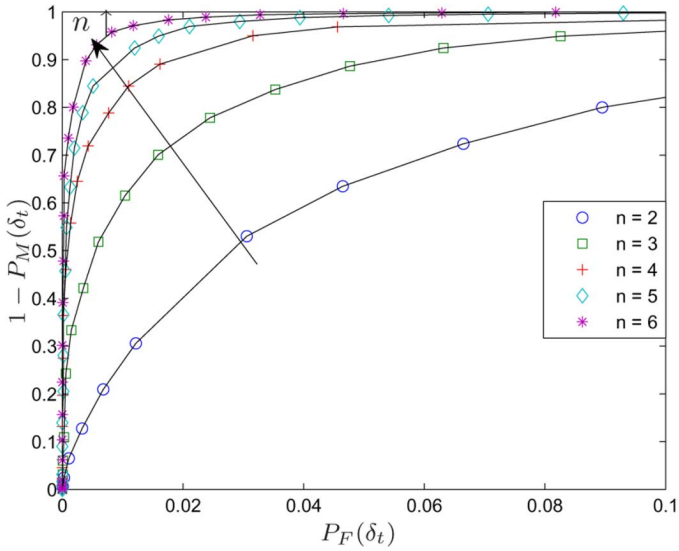

Fig. 13. ROCs for detecting bounded delay flows: $\lambda = 4$, $\Delta = 1$, $f_c = 0.2$, $n = 2, \ldots, 6$, 100 packets per process, 10 000 Monte Carlo runs.

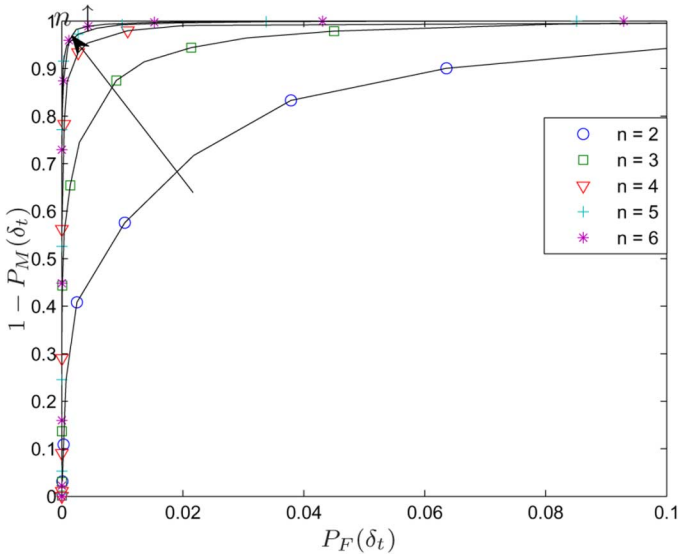

Fig. 14. ROCs for detecting bounded memory flows: $\lambda = 4$, $M = 4$, $f_c = 0.2$, $n = 2, \ldots, 6$, 40 packets per process, 10 000 Monte Carlo runs.

The discussions following Corollaries 4.3 and 4.6 state that there are information flows, which make the detection no better than random guessing, whereas in the simulation, the detector clearly performs much better than random guessing. This observation shows that the chaff-inserting methods used in the simulation are not optimal.

If we compare the ROCs for bounded delay flows with those for bounded memory flows (i.e., Fig. 13 versus Fig. 14 and Fig. 15 versus Fig. 16), we see that the detector of bounded memory flows can achieve similar performance as the detector of bounded delay flows at smaller sample sizes. Note that in our simulations, we have made $M = \lambda\Delta$ to facilitate comparison. There is a natural correlation between the bounded delay model and the bounded memory model by Little's theorem [21]. In the bounded delay model, the maximum delay is bounded by Δ, and the average number of stored packets is bounded by M;

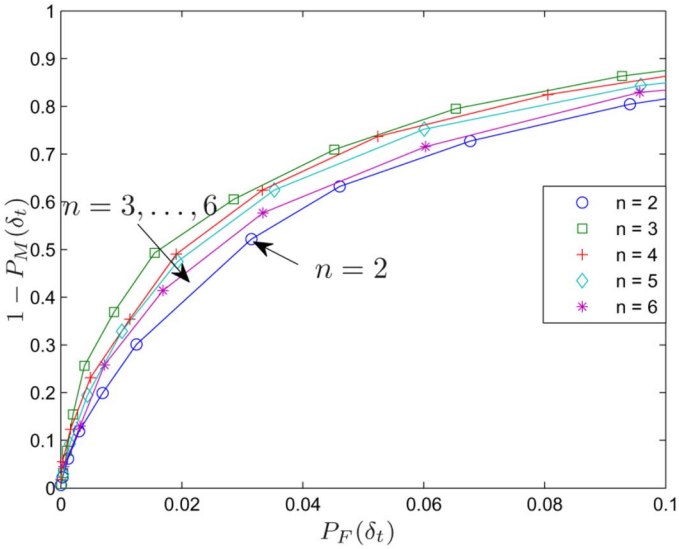

Fig. 15. ROCs for detecting bounded delay flows: $\lambda = 4$, $\Delta = 1$, $f_c = 0.2$, $n = 2, \ldots, 6$, totally 200 packets over all processes, 10 000 Monte Carlo runs.

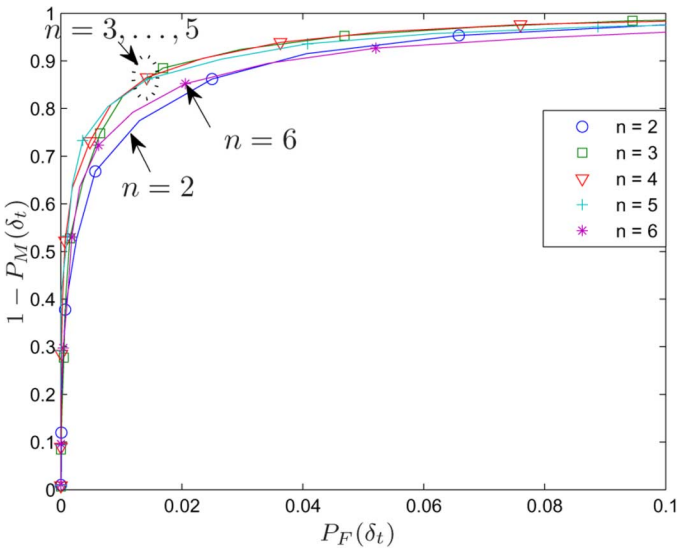

Fig. 16. ROCs for detecting bounded memory flows: $\lambda = 4$, $M = 4$, $f_c = 0.2$, $n = 2, \ldots, 6$, totally 100 packets over all processes, 10 000 Monte Carlo runs.

in the bounded memory model, the maximum number of stored packets is bounded by M, whereas the average delay is bounded by Δ.

VIII. Discussions on Non-Poisson Traffic

We have assumed that the node transmission timestamps can be modeled as independent Poisson processes under $\mathcal{H}_0$. Poisson assumption allows us to obtain clean analytical results, but it is known that wide-area traffic such as Internet traffic does not fit the Poisson model. In this section, we discuss the detection of information flows when the normal traffic is non-Poisson.

First, we point out that the concept of detectability and undetectability, the chaff-inserting algorithms, and the threshold detector based on these algorithms are generally applicable regardless of the traffic model. As the traffic model changes, however, the specific levels of detectability and undetectability, and the threshold of the detector for consistent detection will change accordingly, and it is not guaranteed that the levels of detectability and undetectablity will be equal. Preliminary results on two-hop information flows with bounded delays under renewal traffic (i.e., the normal traffic can be modeled as independent renewal processes) have been presented in [22]; generalization to other types of flows and multiple hops is possible and will be reported separately.

Next, we argue that Poisson processes are less bursty than real-world traffic, and therefore, our results provide lower bounds on the level of detectability of actual information flows. Specifically, it was shown in [23] that renewal processes with the Pareto interarrival distribution[9] of shape parameter close to one fit network traffic over many time scales. For such traffic, we have shown in [22] that it requires much more chaff noise to hide an information flow with bounded delay in independent traffic than it does for Poisson traffic of the same rate. Therefore, the level of undetectability (which can be shown to be equal to the level of detectability) under the Pareto interarrival distribution is higher than that of Poisson traffic. Similar comparison has also been observed for bounded memory flows.

To verify the claim that Poisson assumption provides lower bounds on the actual detection performance, we simulate BGM and BMR on the traces LBL-PKT-4, which contains an hour's worth of all wide-area traffic between the Lawrence Berkeley National Laboratory (Berkeley, CA) and the rest of the world.[10] We compute the CTR of pairs of different traces,[11] and then compare the empirical cumulative distribution function (c.d.f.) of the computed CTR with the c.d.f. of the CTR predicted by Theorems 4.2 and 4.5 for independent Poisson processes of the same rates as the empirical rates of the traces; see Figs. 17 and 18. From these plots, it is clear that at the same threshold, the traces have much lower false alarm probabilities than Poisson processes.

Note that it is not true that the level of detectability for Poisson traffic is always lower than any other type of traffic. In fact, even traffic with the Pareto interarrival distribution can have either a lower or a higher level of detectability depending on the shape parameter; see [22] for details. In this sense, we have analyzed a popular traffic model, Poisson traffic, and our results should be viewed as a benchmark for other types of traffic.

IX. Conclusion

This paper addresses timing-based detection of information flows in the presence of active perturbations and chaff noise. It characterizes the detectability of information flows in terms of the maximum amount of chaff noise that allows consistent detection and shows how to design the detector to achieve consistent detection based on knowledge of the normal traffic. The Poisson assumption under the null hypothesis makes our results

[9]The Pareto distribution with shape parameter β ($\beta \geq 0$) and location parameter α ($\alpha \geq 0$) has the probability density function $p(x) = \beta \alpha^{\beta} x^{-\beta-1}$ ($x \geq \alpha$).

[10]The traces were made by Paxson and were first used in his paper [23].

[11]We extract 134 TCP traces from the data, each of which is truncated to 1000 packets.

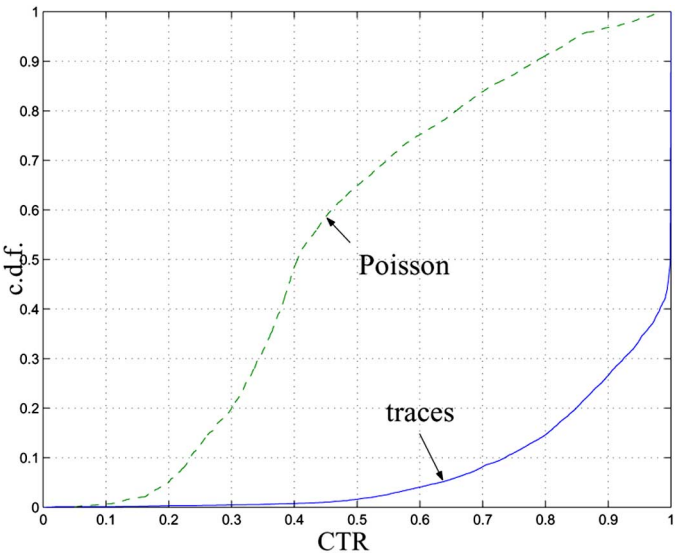

Fig. 17. The c.d.f. of the CTR of BGM for $\Delta = 5$: CTR on traces versus CTR on Poisson processes.

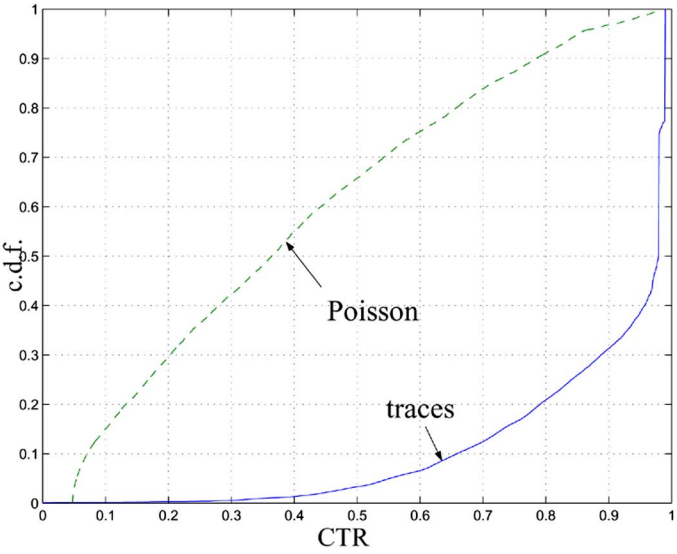

Fig. 18. The c.d.f. of the CTR of BMR for $M = 20$: CTR on traces versus CTR on Poisson processes.

lower bounds on the detection performance of practical information flows. The proposed detector coupled with capacity constraints between neighbor nodes can capture all the long-lived information flows with positive rates and sufficiently long paths.

APPENDIX I

A. Proof of Theorem 4.2

Let Y_j be the jth packet delay, i.e., $Y_j = S_2(j) - S_1(j)$. Define

$$\begin{aligned} Z_j &\triangleq Y_j - Y_{j-1} \\ &= (S_2(j) - S_2(j-1)) - (S_1(j) - S_1(j-1)). \end{aligned}$$

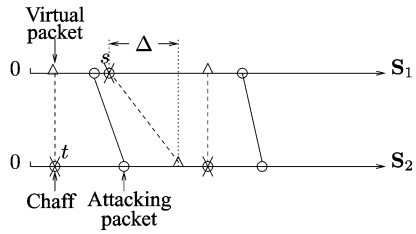

Fig. 19. Inserting virtual packets to calculate the delays of chaff packets.

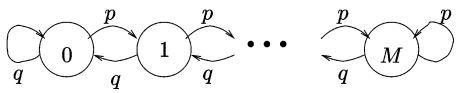

Fig. 20. Markov chain formed by $\{d'(w)\}; p = \frac{\lambda_1}{\lambda_1+\lambda_2}, q = 1-p$.

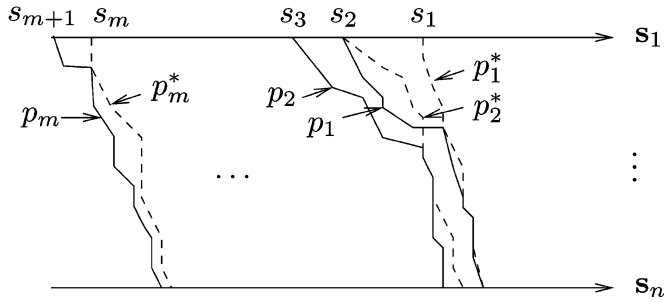

Fig. 21. Every relay sequence in $\mathcal{P}^*$ corresponds to a relay sequence in $\mathcal{P}$; solid line: sequences in $\mathcal{P}$; dashed line: sequences in $\mathcal{P}^*$.

We see that Z_j's are i.i.d. random variables; each Z_j is the difference between two independent exponential random variables with mean $1/\lambda_2$ and $1/\lambda_1$, respectively. The process $\{Y_j\}_{j=1}^{\infty}$ is a general random walk with step Z_j. Define $Y_0 = 0$.

Now for every chaff packet inserted at t in $\mathbf{S}_2$, we insert a virtual packet at t in $\mathbf{S}_1$; for every chaff packet at s in $\mathbf{S}_1$, we insert a virtual packet at $s + \Delta$ in $\mathbf{S}_2$, as illustrated in Fig. 19. Let the new packet delays after the insertion of virtual packets be $\{Y'_j\}_{j=0}^{\infty}$. It can be shown that $\{Y'_j\}_{j=0}^{\infty}$ is also a random walk with step Z_j, but it has two reflecting barriers at 0 and Δ, i.e.,

$$Y'_j = \min\left(\max\left(Y'_{j-1} + Z_j, 0\right), \Delta\right).$$

Because it is almost surely impossible for $Y'_{j-1} + Z_j$ to be exactly equal to 0 or Δ, each time $Y'_j = 0$ corresponds to a chaff packet in $\mathbf{S}_2$, and $Y'_j = \Delta$ corresponds to a chaff packet in $\mathbf{S}_1$. Thus, the limiting probability for a packet to be chaff is $h_\Delta/(1-h_0)$ in $\mathbf{S}_1$, and $h_0/(1-h_\Delta)$ in $\mathbf{S}_2$, where $h_0 = \lim_{j\to\infty} \Pr\{Y'_j = 0\}$, and $h_\Delta = \lim_{j\to\infty} \Pr\{Y'_j = \Delta\}$. The overall probability for a packet in $\mathbf{S}_1 \oplus \mathbf{S}_2$ to be chaff is a weighted sum

$$\frac{\lambda_1 h_\Delta}{(\lambda_1+\lambda_2)(1-h_0)} + \frac{\lambda_2 h_0}{(\lambda_1+\lambda_2)(1-h_\Delta)}. \tag{23}$$

By ergodicity of $\{Y'_j\}_{j=0}^{\infty}$, the CTR of BGM converges to the limiting probability in (23) almost surely.

Now we calculate h_0 and h_Δ. Let the equilibrium distribution function of Y_j' be $H(x)$, i.e., $H(x) = \lim_{j\to\infty} \Pr\{Y_j' \le x\}$. It is shown in [24, Example 2.16] that

$$h_0 = H(0) = \begin{cases} \frac{1-\frac{\lambda_1}{\lambda_2}}{1-\left(\frac{\lambda_1}{\lambda_2}\right)^2 e^{\Delta(\lambda_1-\lambda_2)}}, & \text{if } \lambda_1 \neq \lambda_2 \\ \frac{1}{2+\lambda_1\Delta}, & \text{o.w.} \end{cases}$$

and[12]

$$h_\Delta = 1 - H(\Delta-) = \begin{cases} \frac{\left(\frac{\lambda_1}{\lambda_2}\right)e^{\Delta(\lambda_1-\lambda_2)}\left(1-\frac{\lambda_1}{\lambda_2}\right)}{1-\left(\frac{\lambda_1}{\lambda_2}\right)^2 e^{\Delta(\lambda_1-\lambda_2)}}, & \text{if } \lambda_1 \neq \lambda_2 \\ \frac{1}{2+\lambda_1\Delta}, & \text{o.w.} \end{cases}$$

Therefore, by (23), we have that the CTR of BGM satisfies

$$\lim_{t\to\infty} \mathrm{CTR}_{\mathrm{BGM}}(t) = \begin{cases} \frac{(\lambda_2-\lambda_1)\left(1+\left(\frac{\lambda_1}{\lambda_2}\right)e^{\Delta(\lambda_1-\lambda_2)}\right)}{(\lambda_1+\lambda_2)\left(1-\left(\frac{\lambda_1}{\lambda_2}\right)e^{\Delta(\lambda_1-\lambda_2)}\right)}, & \text{if } \lambda_1 \neq \lambda_2 \\ \frac{1}{1+\lambda_1\Delta}, & \text{if } \lambda_1 = \lambda_2 \end{cases}$$

almost surely. □

B. Proof of Proposition 4.4

Algorithm BMR is feasible because the nonchaff part $(\mathbf{f}_1, \mathbf{f}_2)$ satisfies the bounded memory constraint. It remains to show the optimality.

Assume that C^* is an optimal chaff-inserting algorithm. If $M_1(k-1) = M$ and $s_k \in \mathcal{S}_1$, then node R_2 has an arrival when the memory is full, and C^* has to drop at least one arriving packet at or before s_k to prevent memory overflow. If $M_1(k-1) = 0$ and $s_k \in \mathcal{S}_2$, then R_2 has a departure when the memory is empty, so C^* has to insert at least one dummy packet at or before s_k in $\mathbf{s}_2$ to prevent memory underflow. Therefore, BMR inserts no more chaff than C^*. □

C. Proof of Theorem 4.5

If $\mathbf{S}_1$ and $\mathbf{S}_2$ are independent Poisson processes of rates λ_1 and λ_2, respectively, then it is known that the cumulative differences $\{d(w)\}$ defined in (20) form a simple random walk. Algorithm BMR assigns chaff such that the cumulative differences $\{d'(w)\}$ of the processes $\mathbf{F}_1$ and $\mathbf{F}_2$ satisfy $0 \le d'(w) \le M$ for all w. By the memoryless property of exponential interarrival times, it is easy to see that $\{d'(w)\}$ is a random walk with reflecting barriers at 0 and M (i.e., a Markov chain with state space $\{0, \ldots, M\}$). Its transition probabilities are shown in Fig. 20.

It is easy to see that $\{d'(w)\}$ is an irreducible, aperiodic, and positive recurrent Markov chain, and thus has a limit distribution $(\pi_0, \ldots, \pi_M)$. Because the limit distribution satisfies

$$\pi_i = \frac{\lambda_1}{\lambda_2}\pi_{i-1}, \qquad i = 1, \ldots, M$$

we have

$$\pi_0 = \begin{cases} \frac{1-\frac{\lambda_1}{\lambda_2}}{1-\left(\frac{\lambda_1}{\lambda_2}\right)^{M+1}}, & \text{if } \lambda_1 \neq \lambda_2 \\ \frac{1}{1+M}, & \text{o.w.} \end{cases}$$

$$\pi_M = \begin{cases} \frac{\left(\frac{\lambda_1}{\lambda_2}\right)^M\left(1-\frac{\lambda_1}{\lambda_2}\right)}{1-\left(\frac{\lambda_1}{\lambda_2}\right)^{M+1}}, & \text{if } \lambda_1 \neq \lambda_2 \\ \frac{1}{1+M}, & \text{o.w.} \end{cases}$$

The physical meaning of $d'(w)$ is the number of stored packets after the transmission of the wth packets in $\mathbf{S}_1 \oplus \mathbf{S}_2$. The self-loop at state 0 corresponds to chaff packets in $\mathbf{S}_2$ because these transmissions occur when the memory is empty (so they have to be dummy packets); the self-loop at state M corresponds to chaff in $\mathbf{S}_1$ because the transmissions occur when the memory is full (so the packets will be dropped).

By ergodicity of $\{d'(w)\}$, as $w \to \infty$, the CTR of BMR converges to the limiting probability of self-loops almost surely. The limiting probability is a weighted sum $\pi_0 q + \pi_M p$, which is equal to

$$\begin{cases} \frac{(\lambda_2-\lambda_1)\left(1+\left(\frac{\lambda_1}{\lambda_2}\right)^{M+1}\right)}{(\lambda_1+\lambda_2)\left(1-\left(\frac{\lambda_1}{\lambda_2}\right)^{M+1}\right)}, & \text{if } \lambda_1 \neq \lambda_2 \\ \frac{1}{1+M}, & \text{if } \lambda_1 = \lambda_2. \end{cases}$$

□

D. Proof of Proposition 5.1

By expanding the recursions of MBDR, it can be shown that MBDR is equivalent to an algorithm, which finds the earliest sequence of relay timestamps for each packet in $\mathbf{s}_1$. That is, for $s \in \mathcal{S}_1$, if $p = (s, t_2, \ldots, t_n)$ $(t_i \in \mathcal{S}_i)$ is a sequence of relay timestamps for s, then MBDR finds the sequence $\tilde{p} = (s, \tilde{t}_2, \ldots, \tilde{t}_n)$ such that:

1) $\tilde{p}$ satisfies the causality and the bounded delay constraints;
2) $\tilde{t}_i \le t_i$ $(i = 2, \ldots, n)$ for any other sequence of relay timestamps that satisfies these constraints.

We will refer to a sequence of relay timestamps as a *relay sequence*.

A set of relay sequences preserves the order of packets if for any two sequences $(t_i)_{i=1}^n$ and $(t_i')_{i=1}^n$ in the set, $t_1 \le t_1'$ implies $t_i \le t_i'$ for all $i = 2, \ldots, n$. We will use the following result.

Lemma 10.1: Among all sets of relay sequences satisfying the constraints of causality, packet conservation, and bounded delay, there always exists a set which has the largest size and preserves the order of packets.

By this lemma, it suffices to search among order-preserving sets of relay sequences. It remains to show that it is optimal to find the earliest relay sequences.

Let $\mathcal{P}$ be the set of relay sequences found by MBDR, and $\mathcal{P}^*$ be the largest and order-preserving set of relay sequences. Suppose $s_1 \in \mathcal{S}_1$ has a relay sequence $p_1^* \in \mathcal{P}^*$ but not in $\mathcal{P}$, as illustrated in Fig. 21. Then, there must be relay sequences in $\mathcal{P}$, which start earlier than s_1 and partly overlap with p_1^* (otherwise, MBDR would have chosen p_1^* or a sequence earlier than p_1^* for

[12]Because $H(\cdot)$ is not continuous at Δ, we use $H(\Delta-)$ to denote the left limit of $H(x)$ as x approaches Δ from the left (note that $H(\Delta) = 1$).

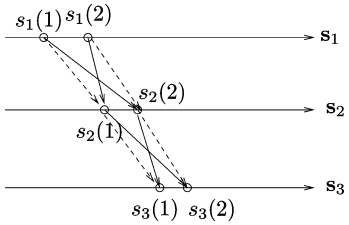

Fig. 22. Solid lines denote the original relay sequences; dashed lines denote the reorganized relay sequences that preserve the order of packets.

s_1). Let the earliest of these sequences be p_1, with starting timestamp $s_2 \in \mathcal{S}_1$. For $j = 2, 3, \ldots$, do the following.

i) If s_j does not have a relay sequence in $\mathcal{P}^*$, we stop searching; otherwise, suppose that s_j has a relay sequence $p_j^* \in \mathcal{P}^*$.

ii) The sequence p_j^* is at least partly earlier than p_{j-1} because p_j^* is earlier than p_{j-1}^* and p_{j-1}^* partly overlaps with p_{j-1}. Because MBDR has not chosen the earlier part of p_j^*, it implies that there must be sequences in $\mathcal{P}$ earlier than p_{j-1}, which partly overlap with p_j^*. Let the earliest of these sequences be p_j, with starting timestamp $s_{j+1} \in \mathcal{S}_1$. Continue with i).

When we stop searching, we will either find a timestamp in $\mathbf{s}_1$, which has a relay sequence in $\mathcal{P}$, but not in $\mathcal{P}^*$, or reach a relay sequence $p_m \in \mathcal{P}$, which starts before all the relay sequences in $\mathcal{P}^*$. Therefore, for every relay sequence in $\mathcal{P}^*$, we can find a different sequence in $\mathcal{P}$, which implies that the size of $\mathcal{P}$ is no smaller than that of $\mathcal{P}^*$. □

E. Proof of Lemma 10.1

The proof is by induction. As illustrated in Fig. 22, suppose that $(s_1(1), s_2(2), s_3(1))$ and $(s_1(2), s_2(1), s_3(2))$ are relay sequences satisfying causality, packet conservation, and bounded delay. By switching the intersected part, we obtain two sequences $(s_1(1), s_2(1), s_3(1))$ and $(s_1(2), s_2(2), s_3(2))$, which satisfy these constraints and also preserve the packet order. By repeatedly applying such switching, we can reorganize any set of relay sequences into an order-preserving set and maintain satisfaction of the constraints. □

F. Proof of Theorem 5.2

We bound the CTR of MBDR by deriving upper bounds on the probability for an arbitrary packet in $\mathbf{S}_1$ to have a match. Then, the result of Theorem 5.2 holds by ergodicity. Compared with the first packet, subsequent packets are more difficult to match because some of their relay timestamps may have been used to relay previous packets. Thus, it suffices to upper bound the probability for the first packet to have a match. Denote this probability by P_n.

First, note that a necessary condition for the first packet at time t to have a match is that the corresponding intervals $[t, t + (i-1)\Delta]$ in $\mathbf{S}_i$ $(i = 2, \ldots, n)$ in which the packet can be relayed are all nonempty. The probability for this event is at most $\prod_{i=1}^{n-1}(1 - e^{-i\lambda\Delta})$ (achievable if all the processes have rate λ). Thus, $P_n \le \prod_{i=1}^{n-1}(1 - e^{-i\lambda\Delta})$.

Next, we prove by induction that P_n is also upper bounded by $(\lambda\Delta)^{n-2}(1 - e^{-\lambda\Delta})$. For $n = 2$, this bound is the same as the upper bound derived above. Assume that the result holds for P_{n-1} $(n \ge 3)$. By writing P_n in parts with respect to the number of timestamps within delay Δ in $\mathbf{S}_2$, we have

$$\begin{aligned} P_n &\le \sum_{k=1}^{\infty} \frac{(\lambda\Delta)^k}{k!} e^{-\lambda\Delta} \\ &\quad \cdot \Pr\{\text{at least one of the } k \text{ timestamps has a match}\} \\ &\le \sum_{k=1}^{\infty} \frac{(\lambda\Delta)^k}{k!} e^{-\lambda\Delta} k P_{n-1} \\ &= \lambda\Delta P_{n-1} \end{aligned} \tag{24}$$

where union bound is used to obtain (24). Hence, we have shown that $P_n \le (\lambda\Delta)^{n-2}(1 - e^{-\lambda\Delta})$.

Combining these two bounds, we have that the CTR of MBDR is lower bounded by

$$1 - \min\left(\prod_{i=1}^{n-1}(1 - e^{-i\lambda\Delta}), (\lambda\Delta)^{n-2}(1 - e^{-\lambda\Delta})\right) \text{ a.s.}$$

G. Asymptotic CTR of MBMR

Here we show how to calculate the CTR of MBMR by a Markov chain. In particular, we are interested in computing β_n^M $(n \ge 2)$. Assume the processes are independent and Poisson under $\mathcal{H}_0$.

If $\mathbf{S}_1, \ldots, \mathbf{S}_n$ are independent Poisson processes, then the vectors $\left\{(M_i(k))_{i=1}^{n-1}\right\}_{k=0}^{\infty}$ computed by MBMR form an $(n-1)$-dimensional homogeneous Markov chain. By arguments similar to that in the proof of Theorem 4.5, it can be shown that the CTR is minimized when all $\mathbf{S}_i$'s have equal rate, in which case the CTR of MBMR is β_n^M. We will focus on the equal rate case although the method is easily generalizable to arbitrary rates.

If $\mathbf{S}_i$ $(i = 1, \ldots, n)$ have equal rate, then the transition probabilities of $\left\{(M_i(k))_{i=1}^{n-1}\right\}$ are as follows. Denote the transition probability by $\Pr\left\{\mathbf{m}_1^{n-1} | \mathbf{m}'^{n-1}_1\right\}$, where $\mathbf{m}_1^{n-1}, \mathbf{m}'^{n-1}_1 \in \{0, \ldots, M\}^{n-1}$, and $(m_i, \ldots, m_j)$ $(i \le j)$ by $\mathbf{m}_i^j$. For $2 \le i \le n-1$, $m_{i-1} > 0$, and $m_i < M$

$$\Pr\left(\mathbf{m}_1^{i-2}, m_{i-1} - 1, m_i + 1, \mathbf{m}_{i+1}^{n-1} | \mathbf{m}_1^{n-1}\right) = \frac{1}{n}$$

for $m_1 < M$

$$\Pr\left(m_1 + 1, \mathbf{m}_2^{n-1} | \mathbf{m}_1^{n-1}\right) = \frac{1}{n}$$

for $m_{n-1} > 0$

$$\Pr\left(\mathbf{m}_1^{n-2}, m_{n-1} - 1 | \mathbf{m}_1^{n-1}\right) = \frac{1}{n}$$

moreover

$$\Pr(\mathbf{m}_1^{n-1} | \mathbf{m}_1^{n-1}) = \frac{1}{n} \cdot \left(I_{\{m_1 = M\}} + \sum_{i=2}^{n-1} I_{\{m_{i-1}=0 \vee m_i = M\}} + I_{\{m_{n-1}=0\}}\right).$$

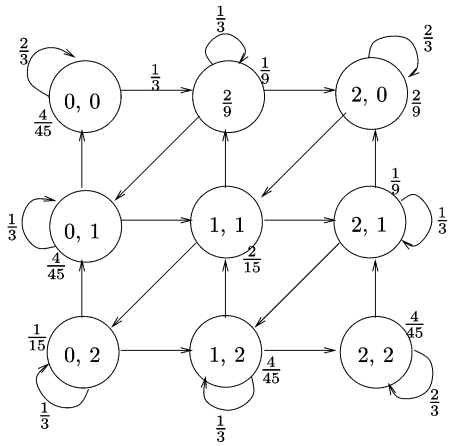

Fig. 23. Markov chain of $\{(M_1(k), M_2(k))\}_{k=0}^{\infty}$. All straight lines have transition probability 1/3. All the states are marked with their limiting probabilities, e.g., $\pi(0,2) = 1/15$.

According to MBMR, each self-loop corresponds to a chaff packet, and therefore, the CTR is equal to the probability of self-loops in the equilibrium distribution. That is, if π is the equilibrium distribution of $\{(M_i(k))_{i=1}^{n-1}\}$, then the CTR of MBMR converges to the limiting probability of self-loops, denoted by η_n, almost surely, where

$$\eta_n = \sum_{\mathbf{m}_1^{n-1} \in \{0,\ldots,M\}^{n-1}} \pi\left(\mathbf{m}_1^{n-1}\right) \Pr\left(\mathbf{m}_1^{n-1} | \mathbf{m}_1^{n-1}\right).$$

For example, for $n = 3$ and $M = 2$, $(M_1(k), M_2(k))$ $(k \geq 0)$ follows the Markov chain in Fig. 23. Here $\eta_3 = \frac{1}{3}(\frac{1}{15} + \frac{2\times4}{45} + \frac{2\times1}{9}) + \frac{2}{3}(\frac{2\times4}{45} + \frac{2}{9}) = \frac{19}{45}$. This is the CTR of MBMR for three-hop information flows with memory size 2, i.e., $\beta_3^2 = \frac{19}{45}$.

H. Proof of Theorem 5.5

We prove the theorem by induction.

For $n = 2$, we have seen from Theorem 4.5 that the minimum CTR of MBMR is $1/(1+M)$.

Assume the result holds up to n $(n \geq 2)$. For $(n+1)$-hop flows, it suffices to show that $1 - u_{n+1} \leq \lim_{t\to\infty} \mathrm{CTR}_{\mathrm{MBMR}}(t) \leq 1 - l_{n+1}$ a.s. when $\mathbf{S}_i$'s have equal rate. This is because equal rate is the case that minimizes CTR (which can be shown by arguments similar to Theorem 4.5). We prove the result by showing that the asymptotic fraction of nonchaff packets (i.e., 1− CTR) is bounded between l_{n+1} and u_{n+1}.

Note that the output of a relay node is no longer a Poisson process. This is because the probability of finding another information-carrying packet after an information-carrying packet is greater than the probability of finding an information-carrying packet after a chaff packet. The precise model to decide whether a packet is chaff is the Markov chain shown in Appendix I-G. As a result, the arrival process at node R_{n+1} is more regular than a Poisson process of the same rate.

For the lower bound, assuming $\mathbf{S}_i$'s all have rate λ, we substitute the arrival process at node R_{n+1} with a Poisson process of rate λl_n. Because we destroy the regularity and may also reduce the rate (because λl_n is a lower bound on the rate), this substitution gives us a lower bound on the fraction of nonchaff packets. For this arrival process and an independent Poisson process of rate λ, which is the departing process of R_{n+1}, we know from the proof of Theorem 4.5 that the asymptotic fraction of chaff packets in the departing process is

$$\pi_0 = \frac{1 - \frac{\lambda_1}{\lambda_2}}{1 - \left(\frac{\lambda_1}{\lambda_2}\right)^{M+1}}$$

where $\lambda_1 = \lambda l_n$ and $\lambda_2 = \lambda$. Therefore, we have that the asymptotic fraction of nonchaff packets is lower bounded by

$$1 - \pi_0 = 1 - \frac{1 - l_n}{1 - l_n^{M+1}}$$

which is equal to l_{n+1}.

For the upper bound, we consider the following arrival process at node R_{n+1}. The process is generated by dividing points in a Poisson process of rate λ into consecutive groups of size M/u_n and selecting M consecutive points from the beginning of each group. Analogous to conventional batched processes, we refer to the group size M/u_n as the *period*, and M as the *batch size*. A realization of such a process is drawn in Fig. 24.

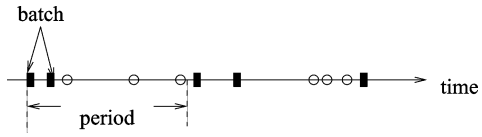

Fig. 24. A "batched" arrival process generated from a Poisson process. ■: arrival timestamps; ○: points in the underlying Poisson process; $M = 2$, period $= 5$.

We consider such a batched process because it maximizes the time between the (kM)th arrival and the $(kM+1)$th arrival $(k \in \mathbb{N})$ so that it is least likely for the memory to be overflowed. Moreover, we choose the period to make the arrival rate equal to λu_n (which may be higher than the actual rate). Therefore, using this arrival process allows us to obtain an upper bound on the fraction of nonchaff packets.

Consider such an arrival process and an independent Poisson process of rate λ. After the first arrival in a period, with probability $2^{-M/u_n}$, there will be no departure until the first arrival in the next period. In this case, there are $M+1$ consecutive arrivals, and thus, at least one packet will be dropped. Hence, the fraction of dropped packets at node R_{n+1} is lower bounded by $2^{-M/u_n}/(M+1)$, i.e., at most $1 - 2^{-M/u_n}/(M+1)$ fraction of the information-carrying packets arriving at R_{n+1} can be successfully relayed. Because at most u_n fraction of the incoming packets of R_{n+1} is carrying information, the overall fraction of information-carrying packets relayed by R_{n+1} is upper bounded by

$$u_n\left(1 - \frac{1}{M+1}2^{-M/u_n}\right)$$

which is equal to u_{n+1}. □

I. Proof of Theorem 6.4

We prove the theorem for bounded delay flows and bounded memory flows separately. Here we present the proof for $n = 2$; the proof for $n \geq 2$ is analogous.

1) Proof for Bounded Memory Flows: By Theorem 4.5, we know that the false alarm probability is maximized when $\lambda_1 = \lambda_2$, where λ_i $(i = 1, 2)$ is the rate of $\mathbf{S}_i$. Consider this equal rate case.

Define T_1 to be the number of packets in $\mathbf{S}_1 \oplus \mathbf{S}_2$ until the first chaff packet, including the first chaff packet, and T_i $(i > 1)$ the number of packets between the $(i-1)$th and ith chaff packets, excluding the $(i-1)$th chaff packet but including the ith. Let C be the number of chaff packets found by BMR. Then, the false alarm probability can be written as

$$\begin{aligned} P_F(\delta_t) &= \Pr\{C \leq \tau_2 N\} \\ &= \Pr\left\{\sum_{i=1}^{\tau_2 N} T_i \geq N\right\} \\ &= \Pr\left\{\frac{1}{\tau_2 N}\sum_{i=1}^{\tau_2 N} T_i \geq \frac{1}{\tau_2}\right\}. \end{aligned} \tag{25}$$

It is know that for Poisson processes, the cumulative differences $\{d(k)\}_{k=1,2,\ldots}$ defined in (20) form a simple random walk with $\Pr\{d(k) = d+1 | d(k-1) = d\} = 1/2$. The Markovian property implies that $T_1, T_2, \ldots$ are independent, and for $i \geq 2$, T_i has the same distribution as $N_{-1,M+1}$ defined by

$$N_{-1,M+1} \triangleq \inf\{k : d(k) = -1 \text{ or } M+1 | d(0) = 0\}. \tag{26}$$

By Theorem 4.5, we know that the ratio C/N will almost surely converge to $1/(1+M)$ as $N \to \infty$, i.e., $\lim_{c\to\infty} c/\left(\sum_{i=1}^{c} T_i\right) = 1/(1+M)$ almost surely. It implies that $\lim_{c\to\infty} \frac{1}{c}\sum_{i=1}^{c} T_i = 1 + M$ almost surely, and thus $\mathbb{E}[T_i] = 1 + M$ $(i \geq 2)$.

Now that T_i's $(i \geq 2)$ are i.i.d., by Sanov's theorem in [25], we have

$$\lim_{N'\to\infty} \frac{1}{N'} \log \Pr\left\{\frac{1}{N'}\sum_{i=1}^{N'} T_i \geq 1/\tau_2\right\} = -\min_{W:\mathbb{E}[W]\geq 1/\tau_2} D(W||T_2)$$

where $N' = \tau_2 N$. By (25), we obtain that

$$\begin{aligned} \lim_{N\to\infty} \frac{1}{N} \log P_F(\delta_t) &= -\tau_2 \min_{W:\mathbb{E}[W]\geq 1/\tau_2} D(W||T_2) \\ &\triangleq -\Gamma_2(\tau_2; M). \end{aligned}$$

It is difficult to compute $\Gamma_2(\tau_2; M)$ directly, but the computation can be reduced to an optimization over a single variable by Cramer's theorem [26]. Nevertheless, as long as $1/\tau_2 > 1 + M$, we have that $\mathbb{E}[W] > \mathbb{E}[T_2]$, and thus $\Gamma_2(\tau_2; M)$ is positive. By the definition of $\Gamma_2(\tau_2; M)$, it is easy to see that it is a decreasing function of τ_2.

2) Proof for Bounded Delay Flows: The proof for bounded delay flows is similar to that for bounded memory flows. By Theorem 4.2, we see that the false alarm probability is maximized when $\mathbf{S}_1$ and $\mathbf{S}_2$ both have the maximum rate λ. Consider this case.

TABLE III
BOUNDED GREEDY MATCH (BGM)

```
Bounded-Greedy-Match(s_1, s_2, Δ):
  m = n = 1;
  while m ≤ |S_1| and n ≤ |S_2|
    if s_2(n) − s_1(m) < 0
      s_2(n) = chaff; n = n + 1;
    else if s_2(n) − s_1(m) > Δ
        s_1(m) = chaff; m = m + 1;
      else
        match s_1(m) with s_2(n);
        m = m + 1; n = n + 1;
      end
    end
  end
```

Let T_i $(i \geq 1)$ be defined the same as in the proof for bounded memory flows. Then, the false alarm probability can be written as

$$P_F(\delta_t) = \Pr\left\{\frac{1}{N'}\sum_{i=1}^{N'} T_i \geq 1/\tau_2\right\} \tag{27}$$

where $N' = \tau_2 N$.

Let Y_j be defined as in the proof of Theorem 4.2. We have shown that the process $\{Y_j\}_{j=1,2,\ldots}$ is a general random walk. For $i \geq 2$, T_i's are i.i.d. with the same distribution as

$$2 \cdot \inf\{j : Y_j < 0 \text{ or } Y_j > \Delta \mid Y_0 = 0\} - 1. \tag{28}$$

Let C be the number of chaff packets found by BGM. By Theorem 4.2, we have $\lim_{N\to\infty} C/N = 1/(1+\lambda\Delta)$ almost surely. Thus, $\lim_{c\to\infty} \frac{1}{c}\sum_{i=1}^{c} T_i = 1 + \lambda\Delta$ almost surely, which implies that $\mathbb{E}[T_2] = 1 + \lambda\Delta$.

By Sanov's theorem [25], we have

$$\lim_{N'\to\infty} \frac{1}{N'} \log \Pr\left\{\frac{1}{N'}\sum_{i=1}^{N'} T_i \geq 1/\tau_2\right\} = -\min_{W:\mathbb{E}[W]\geq 1/\tau_2} D(W||T_2).$$

Plugging in (27) yields that

$$\begin{aligned} \lim_{N\to\infty} \frac{1}{N} \log P_F(\delta_t) &= -\tau_2 \min_{\mathbb{E}[W]\geq 1/\tau_2} D(W||T_2) \\ &\triangleq -\Gamma_2(\tau_2; \lambda, \Delta). \end{aligned}$$

For $1/\tau_2 > 1 + \lambda\Delta$, we have that $\mathbb{E}[W] > \mathbb{E}[T_2]$, and therefore, $\Gamma_2(\tau_2; \lambda, \Delta) > 0$. As τ_2 increases, the minimization is over a larger set, and thus $\Gamma_2(\tau_2; \lambda, \Delta)$ decreases. This completes the proof. □

APPENDIX II

A. Chaff-Inserting Algorithm for Two-Hop Bounded Delay Flows

For the algorithm BGM presented in Section IV-A, we combine the insertion of chaff and the matching of information-carrying packets into the implementation presented in Table III.

TABLE IV
MULTIBOUNDED DELAY RELAY (MBDR)

```
Multi-Bounded-Delay-Relay(s_1, ..., s_n, Δ):
  for k = 1 : |S_1|
    match of s_1(k) = MBDR1(s_1(k), 1, s_1, ..., s_n, Δ);
    if match of s_1(k) = ∅
      s_1(k) = chaff;
    end
  end

MBDR1(s, i, s_1, ..., s_n, Δ):
  for t ∈ S_{i+1} ∩ [s, s + Δ]
    match of t = MBDR1(t, i + 1, s_1, ..., s_n, Δ);
    if match of t = ∅
      t = chaff;
    else
      return t;
    end
  end
  return ∅;
```

TABLE V
EXPANDED MULTIBOUNDED DELAY RELAY (E-MBDR)

```
Expanded-Multi-Bounded-Delay-Relay(s_1, ..., s_n, Δ):
  (p_i)_{i=1}^n = (0, ..., 0);
  for j = 1 : |S_1|
    C_{1,j} = {s_1(j)};
    for i = 1 : n − 1
      for all s ∈ C_{i,j} in increasing order
        for all t ∈ S_{i+1} ∩ [s, s + Δ], t > p_{i+1}, and t ∉ C_{i+1,j}
          t.predecessor = s;
          add t to the set C_{i+1,j};
        end
      end
    end
    if C_{n,j} ≠ ∅
      t_n = min(C_{n,j});
      for i = n − 1 : −1 : 1
        t_i = t_{i+1}.predecessor;
      end
      (t_i)_{i=1}^n is the sequence of relay timestamps for s_1(j);
      (p_i)_{i=1}^n = (t_i)_{i=1}^n;
    end
  end
  for all s ∈ ⋃_{i=1}^n S_i and s ≠ any selected relay timestamp
    s = chaff;
  end
```

This implementation of BGM uses two pointers m and n to record the current timestamps examined in $\mathbf{s}_1$ and $\mathbf{s}_2$, and keeps updating m and n depending on whether the match is successful. Its complexity is $O(|\mathcal{S}_1| + |\mathcal{S}_2|)$.

B. Chaff-Inserting Algorithm for Multihop Bounded Delay Flows

Implementation of the algorithm MBDR presented in Section V-A is presented in Table IV. The complexity of such an direct implementation is $O((\lambda\Delta)^n|\mathcal{S}_1|)$ (λ is the maximum rate of $\mathbf{S}_1, \ldots, \mathbf{S}_n$).

Performance of recursive algorithms can often be improved by expanding recursions. An implementation of expanded MBDR is shown in Table V. The complexity of this implementation is[13] $O\left(n^2|\mathcal{S}_1|\right)$.

TABLE VI
BOUNDED MEMORY RELAY (BMR)

```
Bounded-Memory-Relay(s_1, s_2, M):
  s = s_1 ⊕ s_2;
  d = 0;
  for w = 1 : |S|
    if (d = M and s(w) ∈ S_1) or (d = 0 and s(w) ∈ S_2)
      s(w) = chaff;
    else if s(w) ∈ S_1
        d = d + 1;
      else
        d = d − 1;
      end
    end
  end
```

TABLE VII
MULTIBOUNDED MEMORY RELAY (MBMR)

```
Multi-Bounded-Memory-Relay(s_1, ..., s_n, M):
  s = s_1 ⊕ ··· ⊕ s_n;
  (M_1, ..., M_{n−1}) = (0, ..., 0);
  for w = 1 : |S|
    if s(w) ∈ S_1
      if M_1 < M
        M_1 = M_1 + 1;
      else
        s(w) = chaff;
      end
    else if s(w) ∈ S_n
        if M_{n−1} > 0
          M_{n−1} = M_{n−1} − 1;
        else
          s(w) = chaff;
        end
      else
        let i (1 < i < n) be such that s(w) ∈ S_i;
        if M_{i−1} > 0 & M_i < M
          M_{i−1} = M_{i−1} − 1;
          M_i = M_i + 1;
        else
          s(w) = chaff;
        end
      end
    end
  end
```

C. Chaff-Inserting Algorithm for Two-Hop Bounded Memory Flows

A pseudocode implementation of BMR presented in Section IV-B is given in Table VI.

Note that once BMR marks out the chaff packets, the order in which information-carrying packets are transmitted is irrelevant as far as the memory constraint is concerned. The complexity of BMR is only $O\left(|\mathcal{S}_1| + |\mathcal{S}_2|\right)$.

D. Chaff-Inserting Algorithm for Multi-Hop Bounded Memory Flows

Algorithm MBMR is a direct generalization of BMR. Its implementation is given in Table VII.

[13]The dominating step is the recursive computation of $C_{i,j}$'s. Suppose that the maximum rate of $\mathbf{S}_1, \ldots, \mathbf{S}_n$ is λ, and thus there are at most $(i-1)\lambda\Delta$ points in $C_{i,j}$ on the average. The selection of these points takes $(2i-3)\lambda\Delta$ steps. The total complexity can be calculated by $|\mathcal{S}_1|\sum_{i=2}^{n}(2i-3)\lambda\Delta = \lambda\Delta(n-1)^2|\mathcal{S}_1|$.

TABLE VIII
DETECT BOUNDED DELAY (DBD)

```
Detect-Bounded-Delay(s_1, s_2, Δ, N, λ'):
  i = j = 1;
  C = 0;
  while i + j ≤ N
    if s_2(j) − s_1(i) < 0
      if s_2(j) ≥ Δ
        C = C + 1;
      end
      j = j + 1;
    else if s_2(j) − s_1(i) > Δ
        C = C + 1; i = i + 1;
    else
        i = i + 1; j = j + 1;
    end
  end
  return { H_1   if C/N ≤ 1/(1+λ'Δ),
         { H_0   o.w.;
```

Algorithm MBMR has complexity $O(\sum_{i=1}^{n}|\mathcal{S}_i|)$. It uses $M_i\ (i=1,\ldots,n-1)$ to record the number of packets stored in node R_{i+1}. The algorithm keeps updating M_i's and guarantees that each M_i is always between 0 and M, which implies that the scheduling found by MBMR satisfies the bounded memory constraint.

E. Detection Algorithm for Two-Hop Bounded Delay Flows

Algorithm detect bounded delay (DBD) is derived to detect two-hop information flows with bounded delay. It does detection with the help of the optimal chaff-inserting algorithm BGM.

Given measurements $(\mathbf{s}_1, \mathbf{s}_2)$, DBD

1) calculates C, the number of chaff packets assigned by BGM in $\mathbf{s}_1 \oplus \mathbf{s}_2$ but excluding chaff in $\mathcal{S}_2 \cap [0, \Delta)$;
2) returns $\mathcal{H}_1$ if the ratio of C and the total sample size is less than or equal to $1/(1+\lambda'\Delta)$ (λ' is a design parameter), and otherwise returns $\mathcal{H}_0$.

Implementation of DBD is presented in Table VIII. The complexity of DBD is $O(N)$, where N is the joint sample size, i.e., the total number of examined packets in $\mathbf{s}_1 \oplus \mathbf{s}_2$.

Suppose $\mathcal{H}_1$ is true. Then, the actual number of chaff packets in $\mathbf{s}_1 \oplus \mathbf{s}_2$ has to be no smaller than C because BGM is optimal, and chaff packets in $[0, \Delta)$ in $\mathbf{s}_2$ have been ignored (because they may be the relay packets of packets arriving before the detector starts). It means that the actual CTR has to be more than $1/(1+\lambda'\Delta)$ to evade DBD. Therefore, DBD has no miss detection for $\mathrm{CTR} \le 1/(1+\lambda'\Delta)$.

F. Detection Algorithm for Multihop Bounded Delay Flows

We extend DBD to multiple hops by utilizing the multihop chaff-inserting algorithm MBDR. The algorithm, called detect multibounded delay (DMBD), works as follows.

Given measurements $(\mathbf{s}_1, \ldots, \mathbf{s}_n)$, DMBD

1) calculates C, the number of chaff packets found by MBDR, excluding chaff packets in the beginning $(i-1)\Delta$ period of $\mathbf{s}_i$ for $i = 1, \ldots, n$;
2) returns $\mathcal{H}_1$ if the ratio between C and the total sample size is bounded by τ_n (τ_n is a design parameter); otherwise, returns $\mathcal{H}_0$.

TABLE IX
DETECT MULTIBOUNDED DELAY (DMBD)

```
Detect-Multi-Bounded-Delay(s_1, ..., s_n, Δ, N, τ_n):
  C = 0;
  (J_1, ..., J_n) = (I_1, ..., I_n) = (0, ..., 0);
  K_1 = 0;
  for i = 2 : n
    K_i = sup{k : s_i(k) < (i − 1)Δ};
  end
  j = 1;
  while Σ_{i=1}^{n} J_i < N & j ≤ |S_1|
    C_{1,j} = {s_1(j)};
    for i = 1 : n − 1
      for all s ∈ C_{i,j} in increasing order
        for all t ∈ S_{i+1} ∩ [s, s + Δ], t > s_{i+1}(J_{i+1}), and t ∉ C_{i+1,j}
          t.predecessor = s;
          add t to the set C_{i+1,j};
        end
      end
    end
    if C_{n,j} ≠ ∅
      I_n = min{k : s_n(k) ∈ C_{n,j}};
      for i = n − 1 : −1 : 1
        I_i is such that s_i(I_i) = s_{i+1}(I_{i+1}).predecessor;
      end
      C = C + Σ_{i=1}^{n} (I_i − max(J_i, K_i) − 1);
      (J_1, ..., J_n) = (I_1, ..., I_n);
    end
    j = j + 1;
  end
  C = C + max(N − Σ_{i=1}^{n} max(J_i, K_i), 0);
  Ñ = max(Σ_{i=1}^{n} J_i, N);
  return { H_1   if C/Ñ ≤ τ_n
         { H_0   o.w.;
```

See Table IX for an implementation of DMBD based on the extended version of MBDR. The complexity of this implementation is $O(nN)$.

Because MBDR inserts the minimum number of chaff packets, and chaff packets at the beginning of $\mathbf{s}_i$ are ignored (because they may be the relay packets of information-carrying packets sent before the detector starts), C is always a lower bound on the actual number of chaff packets, which means that the actual CTR has to be larger than τ_n to evade DMBD. Therefore, DMBD has no miss detection for $\mathrm{CTR} \le \tau_n$.

G. Detection Algorithm for Two-Hop Bounded Memory Flows

Algorithm detect bounded memory (DBM) detects two-hop information flows with bounded memory based on the chaff-inserting algorithm BMR.

Given measurements $(\mathbf{s}_1, \mathbf{s}_2)$, DBM

1) calculates $d(w)$ $(w = 1, 2, \ldots)$, the cumulative difference between $\mathbf{s}_1$ and $\mathbf{s}_2$ defined in (20) ($d(0) = 0$);
2) if $v(w) \triangleq \max_{0\le k\le w} d(k) - \min_{0\le k\le w} d(k)$ is less than M for all w, returns $\mathcal{H}_1$; otherwise, computes the smallest index w^* such that $v(w^*) = M$; let $d_u = \max_{0\le k\le w^*} d(k)$ and $d_l = \min_{0\le k\le w^*} d(k)$;
3) calculates C, the number of chaff packets assigned by BMR to keep the variable d between d_l and d_u (the original BMR keeps d between 0 and M);

TABLE X
DETECT BOUNDED MEMORY (DBM)

Detect-Bounded-Memory($\mathbf{s}_1$, $\mathbf{s}_2$, M, N, M'):

$\mathbf{s} = \mathbf{s}_1 \oplus \mathbf{s}_2$;
$d = d_{\max} = d_{\min} = 0$;
$C = 0$;
for $w = 1 : N$
  if ($s(w) \in \mathcal{S}_1$, $d - d_{\min} = M$) or ($s(w) \in \mathcal{S}_2$, $d_{\max} - d = M$)
    $C = C + 1$;
  else
    $d = \begin{cases} d+1 & \text{if } s(w) \in \mathcal{S}_1, \\ d-1 & \text{if } s(w) \in \mathcal{S}_2; \end{cases}$
    $d_{\max} = \max(d_{\max}, d)$;
    $d_{\min} = \min(d_{\min}, d)$;
  end
end
return $\begin{cases} \mathcal{H}_1 & \text{if } \frac{C}{N} \le \frac{1}{1+M'}, \\ \mathcal{H}_0 & \text{o.w.;} \end{cases}$

TABLE XI
DETECT MULTIBOUNDED MEMORY (DMBM)

Detect-Multi-Bounded-Memory($\mathbf{s}_1, \ldots, \mathbf{s}_n$, M, N, τ_n):

$\mathbf{s} = \mathbf{s}_1 \oplus \cdots \oplus \mathbf{s}_n$;
$(M_1, \ldots, M_{n-1}) = (U_1, \ldots, U_{n-1}) = (L_1, \ldots, L_{n-1}) = (0, \ldots, 0)$;
$C = 0$;
for $w = 1 : N$
  i is such that $s(w) \in \mathcal{S}_i$;
  if $i = 1$
    if $M_1 - L_1 < M$
      $M_1 = M_1 + 1$;
      $U_1 = max(U_1, M_1)$;
    else
      $C = C + 1$;
    end
  else if $i = n$
      if $U_{n-1} - M_{n-1} < M$
        $M_{n-1} = M_{n-1} - 1$;
        $L_{n-1} = min(L_{n-1}, M_{n-1})$;
      else
        $C = C + 1$;
      end
    else if $U_{i-1} - M_{i-1} < M$ & $M_i - L_i < M$
        $M_{i-1} = M_{i-1} - 1$;
        $M_i = M_i + 1$;
        $L_{i-1} = min(L_{i-1}, M_{i-1})$;
        $U_i = max(U_i, M_i)$;
      else
        $C = C + 1$;
      end
    end
  end
end
return $\begin{cases} \mathcal{H}_1 & \text{if } \frac{C}{N} \le \tau_n, \\ \mathcal{H}_0 & \text{o.w.;} \end{cases}$

4) returns $\mathcal{H}_1$ if the ratio of C and the total sample size is bounded by $1/(1+M')$ (M' is a design parameter); otherwise, returns $\mathcal{H}_0$.

Implementation of DBM is given in Table X. Algorithm DBM has complexity $O(N)$.

It is shown in [8] that the actual number of chaff packets in $\mathbf{s}_1 \oplus \mathbf{s}_2$ is lower bounded by C. It implies that DBM has no miss detection for realizations of information flows with CTR up to $1/(1+M')$.

H. Detection Algorithm for Multihop Bounded Memory Flows

We extend DBM to a joint detection algorithms called detect multibounded memory (DMBM) based on the chaff-inserting algorithm MBMR.

Given measurements $(\mathbf{s}_1, \ldots, \mathbf{s}_n)$, DMBM

1) for $i = 1, \ldots, n-1$, calculates $v_i(w) \triangleq \max_{0 \le k \le w} d_i(k) - \min_{0 \le k \le w} d_i(k)$ for $w = 1, 2, \ldots$, where $d_i(k)$ is the cumulative difference between $\mathbf{s}_i$ and $\mathbf{s}_{i+1}$;
2) if $v_i(w) < M$ for all w, $U_i = \infty$ and $L_i = -\infty$; otherwise, let $U_i = \max_{0 \le k \le w^*} d_i(k)$, and $L_i = \min_{0 \le k \le w^*} d_i(k)$, where $w^* = \inf\{w : v_i(w) \ge M\}$;
3) calculates C, the number of chaff packets assigned by MBMR to keep the variable M_i $(i = 1, \ldots, n-1)$ between L_i and U_i (originally, MBMR keeps M_i between 0 and M);
4) returns $\mathcal{H}_1$ if the ratio of C and the total sample size is bounded by τ_n (τ_n is a design parameter); otherwise, returns $\mathcal{H}_0$.

Implementation of DMBM is presented in Table XI. The algorithm has complexity $O(N)$.

The value of C is the number of times that memory overflow or underflow would have occurred if chaff packets had not been inserted. For an information flow with bounded memory M, the actual number of chaff packets is at least C, and thus the actual CTR has to be larger than τ_n to evade DMBM. Therefore, DMBM has no miss detection for $\mathrm{CTR} \le \tau_n$.

REFERENCES

[1] D. Donoho, A. Flesia, U. Shankar, V. Paxson, J. Coit, and S. Staniford, "Multiscale stepping-stone detection: Detecting pairs of jittered interactive streams by exploiting maximum tolerable delay," in *Lecture Notes in Computer Science 2516*. Berlin, Germany: Springer-Verlag, 2002.

[2] P. Venkitasubramaniam, T. He, and L. Tong, "Anonymous networking amidst eavesdroppers," *IEEE Trans. Inf. Theory*, vol. 54, Special Issue on Information-Thoeretic Security, no. 6, pp. 2770–2784, Jun. 2008.

[3] S. Staniford-Chen and L. Heberlein, "Holding intruders accountable on the internet," in *Proc. IEEE Symp. Security Privacy*, Oakland, CA, May 1995, pp. 39–49.

[4] X. Wang, D. Reeves, S. Wu, and J. Yuill, "Sleepy watermark tracing: An active network-based intrusion response framework," in *Proc. 16th Int. Inf. Security Conf.*, 2001, pp. 369–384.

[5] Y. Zhang and V. Paxson, "Detecting stepping stones," in *Proc. 9th USENIX Security Symp.*, Aug. 2000, pp. 171–184.

[6] K. Yoda and H. Etoh, "Finding a connection chain for tracing intruders," in *Lecture Notes in Computer Science 1895*. Berlin, Germany: Springer-Verlag, Oct. 2000.

[7] X. Wang, D. Reeves, and S. Wu, "Inter-packet delay-based correlation for tracing encrypted connections through stepping stones," in *Lecture Notes in Computer Science 2502*. Berlin, Germany: Springer-Verlag, 2002, pp. 244–263.

[8] T. He and L. Tong, "Detecting encrypted stepping-stone connections," *IEEE Trans. Signal Process.*, vol. 55, no. 5, pp. 1612–1623, May 2007.

[9] X. Wang and D. Reeves, "Robust correlation of encrypted attack traffic through stepping stones by manipulation of inter-packet delays," in *Proc. ACM Conf. Comput. Commun. Security*, 2003, pp. 20–29.

[10] A. Blum, D. Song, and S. Venkataraman, "Detection of interactive stepping stones: Algorithms and confidence bounds," in *Proc. Conf. Recent Advance Intrusion Detection*, Sophia Antipolis, French Riviera, France, Sep. 2004, pp. 258–277.

[11] P. Peng, P. Ning, D. Reeves, and X. Wang, "Active timing-based correlation of perturbed traffic flows with chaff packets," in *Proc. 25th IEEE Int. Conf. Distributed Comput. Syst. Workshops*, Columbus, OH, Jun. 2005, pp. 107–113.

[12] L. Zhang, A. Persaud, A. Johson, and Y. Guan, "Detection of stepping stone attack under delay and Chaff perturbations," in *Proc. 25th IEEE Int. Performance Comput. Commun. Conf.*, Phoenix, AZ, Apr. 2006, pp. 36–45.

[13] T. He and L. Tong, "Detecting information flows: Improving chaff tolerance by joint detection," in *Proc. Conf. Inf. Sci. Syst.*, Baltimore, MD, Mar. 2007, pp. 51–56.

[14] T. He and L. Tong, "Distributed detection of information flows," *IEEE Trans. Inf. Forensics Security*, vol. 3, Special Issue on Statistical Methods for Network Security and Forensics, no. 3, pp. 390–403, Sep. 2008.
[15] P. Venkitasubramaniam, T. He, L. Tong, and S. Wicker, "Toward an analytical approach to anonymous wireless networking," *IEEE Commun. Mag.*, vol. 46, no. 2, pp. 140–146, Feb. 2008.
[16] X. Hong, P. Wang, J. Kong, Q. Zheng, and J. Liu, "Effective probabilistic approach protecting sensor traffic," in *Proc. Military Commun. Conf.*, Atlantic City, NJ, Oct. 2005, pp. 1–7.
[17] Y. Zhu, X. Fu, B. Graham, R. Bettati, and W. Zhao, "On flow correlation attacks and countermeasures in mix networks," in *Proc. Privacy Enhancing Technol. Workshop*, May 26–28, 2004, pp. 207–225.
[18] J. Shao, *Mathematical Statistics*. New York: Springer-Verlag, 1999.
[19] T. He, P. Venkitasubramaniam, and L. Tong, "Packet scheduling against stepping-stone attacks with chaff," in *Proc. IEEE Military Commun. Conf.*, Washington, DC, Oct. 2006, pp. 1–7.
[20] H. V. Poor, *An Introduction to Signal Detection and Estimation*. New York: Springer-Verlag, 1994.
[21] D. P. Bertsekas and R. Gallager, *Data Networks*. Englewood Cliffs, NJ: Prentice-Hall, 1992.
[22] T. He, A. Agaskar, and L. Tong, "On security-aware transmission scheduling," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, Las Vegas, NV, Mar. 2008, pp. 1681–1684.
[23] V. Paxson and S. Floyd, "Wide-area traffic: The failure of Poisson modeling," *IEEE/ACM Trans. Netw.*, vol. 3, no. 3, pp. 226–244, Jun. 1995.
[24] D. Cox and H. Miller, *The Theory of Stochastic Processes*. New York: Wiley, 1965.
[25] T. Cover and J. Thomas, *Elements of Information Theory*. New York: Wiley, 1991.
[26] F. D. Hollander, *Large Deviations (Fields Institute Monographs, 14)*. Providence, RI: American Mathematical Society, 2000.