Detection of Time-Varying Flows in Wireless Networks

Jinsub Kim and Lang Tong
School of ECE, Cornell University, Ithaca, NY 14853. Email: {jk752, lt35}@cornell.edu

***Abstract*—The problem of detecting the presence of time-varying flows in multi-hop wireless networks is considered. In particular, from transmission timing measurements, a test is constructed to determine whether there is a flow of data packets between a pair of nodes. It is assumed that the packet flows may have time-varying (piecewise constant) flow rates.**

First, a timing-based detector is proposed to detect a flow in the given measurements, and its performance analysis follows. Then, based on the detector, a sliding window technique is proposed for continuous monitoring. The techniques are tested using the MSN Voice over IP (VoIP) traffic and the synthetic Poisson traffic.

I. Introduction

This paper considers the problem of detecting the presence of time-varying flows in a multi-hop wireless network. In a wireless network, suppose that we record transmission timings (epochs) of nodes R_1 and R_2, and R_1 and R_2 may have time-varying (piecewise constant) transmission rates. The transmission epochs of R_1 and R_2 may correspond to different scenarios: Some of these epochs may correspond to a packet flow[1] from R_1 to R_2, or vice versa. The flows between R_1 and R_2 may be bidirectional. It is also possible that there is no flow between R_1 and R_2, and the transmission epochs at these two nodes represent independent transmissions to their corresponding neighboring nodes. Our objective is to detect the presence of a flow between R_1 and R_2.

This problem has a number of practical applications. In intrusion detection, the interactive stepping stone attack has the property that a sequence of nodes (stepping stones) in the attack path relay packets back and forth. For surveillance applications, using simple monitoring devices, one may be able to figure out the networking configurations, routes, and possibly the roots of multicasting trees. Fig. 1 illustrates a specific application to network security, where transmission epochs of a wireless device (R_1) and an access point (R_2) are recorded. By detecting the flow between R_1 and R_2, one can see whether R_1 is injecting packets into the area covered by the access point R_2.

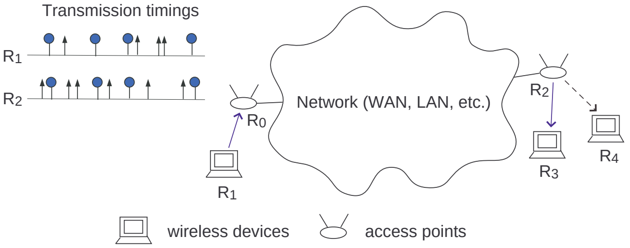

Fig. 1. R_1 is sending packets to R_3 that is in the area covered by R_2, thereby forming a packet flow from R_1 to R_2; the epochs of the packets are marked with circles. Besides the flow, R_1 and R_2 may have other transmissions (marked with arrows): control/management packets, and other data packets.

Using timing for flow detection is nontrivial, partly because we do not assume any information from packet headers; only the timing of transmission is used. Of course, header information may be available in many cases. Such information should then be incorporated into the detection scheme, which is beyond the scope of this paper. In addition, when there exists a flow between R_1 and R_2, some of their transmission epochs might not correspond to the flow. Such epochs are referred to as *chaff* epochs. Chaff epochs can originate from various sources. A node might multiplex the transmissions of intersecting flows, and it can also add dummy transmissions to confuse the detection system.

In the absence of any header information, we need to impose certain constraint on how nodes relay packets that belong to a certain flow. A practical constraint is that, if a node forwards a flow packet, it must forward the packet within a deadline Δ. Such a delay constraint is essential for time-sensitive applications such as VoIP, video streaming, etc..

A. Related Work

Our work was motivated by a series of previous works on timing-based detection of two-hop unidirectional flows, which has been actively studied in the context of stepping-stone detection [1]. To deal with encrypted traffic, researchers restricted the observations to the timing measurements. Donoho *et al.* [2] were the first to employ the flow model with a maximum delay constraint. Their multiscale analysis was shown to be able to detect a flow if the flow lasts for a sufficiently long time. Following their seminal work, many practical algorithms were proposed to detect flows with a maximum delay constraint (see references in [3]). Donoho *et al.* [2] also mentioned about the chaff insertion with the claim that their algorithm can detect a flow if the chaff portion is independent of the flow. The independent chaff insertion was also considered by Zhang *et al.* [4] with the assumption that only one node is allowed to insert chaff transmissions.

The flow detection becomes more challenging if arbitrary chaff insertion is allowed. For arbitrary chaff insertion, Blum *et*

Work in this paper was sponsored by Army Research Office MURI Program under award W911NF-08-1-0238. The first author was partially supported by Samsung Scholarship.

[1] If R_2 is relaying some packets received from R_1 to its neighboring node, the transmission epochs of those packets at R_1 and R_2 correspond to the flow from R_1 to R_2.

al. [5] proposed the counting-based algorithm, and analyzed the tradeoff between the sample size and the error probabilities. He and Tong [6] also considered arbitrary chaff insertion and proposed a matching-based algorithm. Under the Poisson traffic assumption, a threshold τ was shown to exist such that if the fraction of chaff is less than τ, the flow is detectable; otherwise, the flow can be hidden by proper chaff insertion.

However, since the aforementioned studies on unidirectional flow detection were done in the context of stepping-stone detection, they excluded the possibility of the presence of bidirectional flows which are common in wireless networks[2]. Hence, their algorithms need to be adjusted for use in wireless networks. Kim and Tong [7] modified the algorithm in [6] to detect a flow in wireless networks. In this paper, we improve the algorithm in [7] so that it can deal with the traffic with varying rates.

B. Summary of Contributions and Organization

First, to detect a flow in the traffic with varying rates, we improve Bidirectional Flow Detector (BFD)[3], the flow detector presented in [7]. Our detection algorithm has several advantages over BFD: (i) Our algorithm can detect a flow even though it is contained in the traffic with varying rates. (ii) BFD needs an accurate threshold that heavily depends on the traffic characteristic, but our algorithm does not require it. (iii) Our algorithm can be used as a heuristic to detect a flow in the traffic with unknown characteristics.

For continuous monitoring of flows, we propose a sliding window technique in which we repeatedly run our detection algorithm over the fixed number of most recent samples, while removing old samples as new samples are collected. We present numerical performance analysis for our techniques, using the MSN VoIP traffic and the synthetic Poisson traffic. Overall, the numerical results are promising, and the monitoring algorithm was able to detect a flow with a reasonably small detection delay and a low false alarm frequency.

The rest of the paper is organized as follows. In Section II, we introduce notations employed throughout the paper, and formulate the flow detection problem. Section III and Section IV present the flow detection algorithm and the monitoring algorithm, respectively. Then, supporting numerical results follow in Section V. Finally, Section VI concludes the paper with remarks.

II. Mathematical Formulation

We model the transmission timings of each node as a point process. Uppercase bold letters (*e.g.,* $\mathbf{S}$) denote point processes, and lowercase bold letters (*e.g.,* $\mathbf{s}$) denote their realizations. $S(i)$ is a random variable representing the ith transmission epoch, and $s(i)$ is its realization. In addition, $\mathcal{S}$ denotes the set of all epochs in the realization $\mathbf{s}$. We define a *superposition operator* $\bigoplus$ for a pair of increasing sequences: given $(a_1, a_2, \ldots)$ and $(b_1, b_2, \ldots)$, $(a_i)_{i=1}^{\infty} \oplus (b_i)_{i=1}^{\infty} = (c_i)_{i=1}^{\infty}$, where c_i is the ith smallest element among the elements of two sequences[4]. Then, we mathematically define a flow between a pair of nodes as follows.

Definition 2.1: A pair of processes $(\mathbf{F}_1, \mathbf{F}_2)$ forms a *flow* if for every realization $\mathbf{f}_1$ and $\mathbf{f}_2$, $\mathbf{f}_i$ can be partitioned into $\mathbf{f}_i^{12}$ and $\mathbf{f}_i^{21}$ ($\mathbf{f}_i = \mathbf{f}_i^{12} \oplus \mathbf{f}_i^{21}$), such that there exist bijections $g_1 : \mathcal{F}_1^{12} \rightarrow \mathcal{F}_2^{12}$ and $g_2 : \mathcal{F}_2^{21} \rightarrow \mathcal{F}_1^{21}$ satisfying $0 \leq g_1(s) - s \leq \Delta, \forall s \in \mathcal{F}_1^{12}$, and $0 \leq g_2(s) - s \leq \Delta, \forall s \in \mathcal{F}_2^{21}$.

$(\mathbf{f}_1^{12}, \mathbf{f}_2^{12})$ and $(\mathbf{f}_2^{21}, \mathbf{f}_1^{21})$ correspond to packet flows in $\mathbf{F}_1 \rightarrow \mathbf{F}_2$ and $\mathbf{F}_2 \rightarrow \mathbf{F}_1$ directions, respectively. The bijection condition means packet conservation, and $g_i(s) - s \in [0, \Delta]$ ensures that every transmission satisfies causality and the delay bound Δ. We define that a pair of point processes $\mathbf{S}_1$ and $\mathbf{S}_2$ *contain a flow* if they can be partitioned into the flow part $(\mathbf{F}_i)$ and the chaff part $(\mathbf{W}_i)$ such that $(\mathbf{F}_1, \mathbf{F}_2)$ is a flow and $\mathbf{S}_i = \mathbf{F}_i \oplus \mathbf{W}_i$.

The flow detection is formulated as follow. Let $\mathbf{S}_1$ and $\mathbf{S}_2$ denote the transmission processes of R_1 and R_2, respectively. Given the measurements $(\mathbf{s}_i)_{i=1}^2$ in the time interval $[0, t]$, we test the following hypotheses:

$$\begin{array}{ll} \mathcal{H}_0 : & \mathbf{S}_1 \text{ and } \mathbf{S}_2 \text{ are independent} \\ \mathcal{H}_1 : & \mathbf{S}_1 \text{ and } \mathbf{S}_2 \text{ contain a flow} \end{array} \tag{1}$$

III. Flow Detection: Traffic with varying rates

A. Fundamental Limit of Timing-based Detection

Under $\mathcal{H}_0$, intuitively, any pair of $\mathbf{S}_1$ and $\mathbf{S}_2$ can be partitioned into the flow part and the chaff part, if the flow rate is sufficiently low. This implies that if a flow rate is low and a large amount of chaff transmissions are allowed, then R_1 and R_2 can hide a flow between them by mimicking $\mathcal{H}_0$. Hence, a flow is detectable only if its strength is strong enough compared to the chaff portion. Under $\mathcal{H}_1$, the flow strength can be measured by the *relative flow rate* defined as below.

Definition 3.1: Let $(\mathbf{s}_i)_{i=1}^2$ be the realization of $(\mathbf{S}_i)_{i=1}^2$ under $\mathcal{H}_1$, and $(\mathbf{f}_i)_{i=1}^2$ and $(\mathbf{w}_i)_{i=1}^2$ denote the realizations of the flow part and the chaff part, respectively. Then, the *relative flow rate* is defined as

$$\mathrm{R}_{\mathrm{f}}(t) \triangleq \frac{\displaystyle\sum_{i=1}^{2} |\mathcal{F}_i \cap [0, t]|}{\displaystyle\sum_{i=1}^{2} |(\mathcal{F}_i \cup \mathcal{W}_i) \cap [0, t]|}, \tag{2}$$

$$\mathrm{R}_{\mathrm{f}} \triangleq \liminf_{t \to \infty} \mathrm{R}_{\mathrm{f}}(t)$$

Therefore, $\mathrm{R}_{\mathrm{f}}(t)$ is the fraction of flow epochs in the observations up to time t, and high $\mathrm{R}_{\mathrm{f}}(t)$ means that the flow strength is strong compared to the chaff portion.

[2] Most studies on stepping-stone detection observe timings of a pair of incoming and outgoing streams at a point. Hence, a flow cannot exist in the direction of from the outgoing stream to the incoming stream.

[3] The original name of the detector is Packet-Forward-Detect, but we rename it to better describe its purpose.

[4] If the same element appears multiple times (total n) in $(a_1, a_2, \ldots)$ and $(b_1, b_2, \ldots)$, then it also appears n times in $(c_1, c_2, \ldots)$.

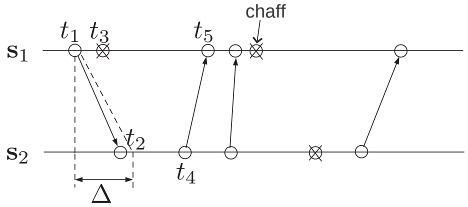

Fig. 2. The illustration of BiBGM operation.

B. Background: Bidirectional Flow Detector

This section introduces an existing timing-based algorithm for flow detection, Bidirectional Flow Detector (BFD)[5] proposed in [7]. BFD calculates an upper bound on the actual $\mathrm{R}_{\mathrm{f}}(t)$, denoted by $\bar{\mathrm{R}}_{\mathrm{f}}(t)$, and compares it to a predetermined threshold τ to make a decision. Specifically, BFD takes the following form.

$$\begin{cases} \text{declare } \mathcal{H}_0 & \text{if } \bar{\mathrm{R}}_{\mathrm{f}}(t) < \tau \\ \text{declare } \mathcal{H}_1 & \text{if } \bar{\mathrm{R}}_{\mathrm{f}}(t) \geq \tau \end{cases} \tag{3}$$

Given $(\mathbf{s}_i)_{i=1}^2$, $\bar{\mathrm{R}}_{\mathrm{f}}(t)$ is obtained by the below optimization.

$$\max_{\substack{\mathbf{f}_i,\, \mathbf{w}_i\,: \\ \mathbf{s}_i = \mathbf{f}_i \oplus \mathbf{w}_i \sim \mathcal{H}_1}} \quad \frac{\sum_{i=1}^{2} |\mathcal{F}_i \cap [0,\, t]|}{\sum_{i=1}^{2} |(\mathcal{F}_i \cup \mathcal{W}_i) \cap [0,\, t]|}$$

where $\mathbf{s}_i = \mathbf{f}_i \oplus \mathbf{w}_i \sim \mathcal{H}_1$ means the constraint that $(\mathbf{f}_1,\, \mathbf{f}_2)$ is a realization of a flow with a maximum delay constraint Δ.

Hence, $\bar{\mathrm{R}}_{\mathrm{f}}(t)$ is obtained by optimally partitioning $(\mathbf{s}_i)_{i=1}^2$ into the flow part and the chaff part such that the flow part is maximized. In [7], a matching algorithm called Bidirectional-Bounded-Greedy-Match (BiBGM) was proposed and proved to achieve this optimal partitioning by finding a maximum number of valid matches. Given $(\mathbf{s}_i)_{i=1}^2$, BiBGM works as follows.

1) Initially, all the epochs in $\mathcal{S}_1 \cup \mathcal{S}_2$ are unmatched.
2) Let s be the earliest epoch in $\mathcal{S}_1 \cup \mathcal{S}_2$. Match s with the first unmatched epoch in $[s,\, s+\Delta]$ in the other node.
3) Move to the next unmatched epoch t in $\mathcal{S}_1 \cup \mathcal{S}_2$. Match t with the first unmatched epoch in $[t,\, t+\Delta]$ in the other node. Keep moving to the next unmatched epoch in $\mathcal{S}_1 \cup \mathcal{S}_2$ and finding its match based on the same rule.
4) After the trial to match the last epoch in $\mathcal{S}_1 \cup \mathcal{S}_2$, label all the unmatched epochs as chaff and terminate.

Fig. 2 illustrates the operation of BiBGM. BiBGM first tries to find a match for t_1, which is the earliest epoch in $\mathcal{S}_1 \cup \mathcal{S}_2$. Since t_2 is the first unmatched epoch in $[t_1,\, t_1+\Delta] \cap \mathcal{S}_2$, t_1 is matched with t_2. Next, BiBGM looks for a match for t_3. However, there is no unmatched epoch in $[t_3,\, t_3+\Delta] \cap \mathcal{S}_2$. Thus, BiBGM marks t_3 as chaff. Then, BiBGM moves to t_4 and searches for an unmatched epoch in $[t_4,\, t_4+\Delta] \cap \mathcal{S}_1$.

The implementation of BiBGM is given in Table. I. Its computational complexity is $O(|\mathcal{S}_1| + |\mathcal{S}_2|)$, which is linear with respect to the sample size.

TABLE I
BIDIRECTIONAL-BOUNDED-GREEDY-MATCH (BIBGM) [7]

```
BiBGM($\mathbf{s}_1,\, \mathbf{s}_2,\, \Delta$):
$m = n = 1$;
while $m \leq |\mathcal{S}_1|$ and $n \leq |\mathcal{S}_2|$
    if $s_2(n) < s_1(m) - \Delta$
        $s_2(n)$ is chaff; $n \leftarrow n+1$;
    else if $s_2(n) > s_1(m) + \Delta$
        $s_1(m)$ is chaff; $m \leftarrow m+1$;
    else
        match $s_1(m)$ with $s_2(n)$;
        $m \leftarrow m+1$; $n \leftarrow n+1$;
    end
end
mark $s_1(i),\, s_2(j)$ with $m \leq i,\, n \leq j$ as chaff;
$\bar{\mathrm{R}}_{\mathrm{f}} \leftarrow \frac{\text{the number of matched epochs}}{|\mathcal{S}_1|+|\mathcal{S}_2|}$;
return $\bar{\mathrm{R}}_{\mathrm{f}}$
```

Under the Poisson traffic assumption, the performance of BFD was analyzed in [7]. Under $\mathcal{H}_0$, $\bar{\mathrm{R}}_{\mathrm{f}}(t)$ converges almost surely to a constant τ_0, which depends on the rates of $\mathbf{S}_1$ and $\mathbf{S}_2$. For any positive number ϵ ($\epsilon < \tau_0$), BFD with the threshold $\tau_0 + \epsilon$ is shown to be consistent[6] if R_{f} under $\mathcal{H}_1$ is greater than $\tau_0 + \epsilon$. In addition, when the chaff parts of $\mathbf{S}_1$ and $\mathbf{S}_2$ are independent Poisson processes, a flow can be detected by BFD consistently, regardless of its strength.

C. Adaptive Flow Detector

In this section, we present Adaptive Flow Detector (AFD), a detection algorithm aimed at detecting a flow in the traffic with varying rates.

The motivation for AFD stems from the following limitations of BFD. To set a proper threshold τ of BFD, we need to know the details of the traffic characteristics, including the interarrival distribution and the traffic rates. However, estimation of such information generally requires a long time and a large number of samples. Furthermore, some traffic characteristics (*e.g.*, traffic rates) might vary in the middle of the observation interval. Hence, we need an adaptive scheme that can detect a flow even though the traffic characteristics are unknown and time-varying.

Instead of a predetermined threshold, AFD employs an adaptive threshold that is obtained based on the measurements as follows. As a first step, AFD assumes temporal independence to approximate the $\mathcal{H}_0$ traffic using the measurements. Fig. 3 describes the approximation procedure, referred to as Independent-Traffic Approximation (ITA). ITA has two parameters, the synthesis window width W_S and the gap α ($\alpha \geq \Delta$) between subsequent windows. The intuition behind ITA is that if α is large enough, then the epochs of $\mathbf{S}_1$ in $A1$ and the epochs of $\mathbf{S}_2$ in $B1$ will tend to be uncorrelated, even when a flow exists. Given the measurements $(\mathbf{s}_i)_{i=1}^2$ in $[0,\, t]$, ITA works as follows:

1) $(\bar{\mathbf{s}}_i)_{i=1}^2$ denotes the resulting data. Initially, $\bar{\mathbf{s}}_1$ and $\bar{\mathbf{s}}_2$ contain no epoch.
2) Take the epochs of $\mathbf{s}_1$ in $[0,\, W_S]$, and add them to $\bar{\mathbf{s}}_1$.

[5] For better description, we rename Packet-Forward-Detect [7] to BFD.

[6] A detector is said to be *consistent* if both the miss detection and the false alarm probabilities vanish as the sample size grows.

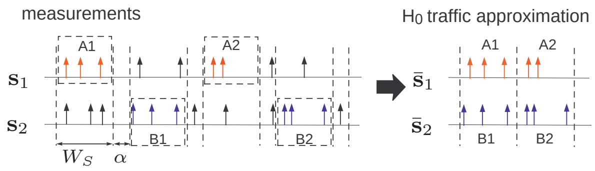

Fig. 3. ITA: The W_S-second intervals $A1$, $A2$, $B1$, and $B2$ are cut from the measurements and assembled to approximate the $\mathcal{H}_0$ traffic.

TABLE II
ADAPTIVE FLOW DETECTOR (AFD)

```
AFD($\mathbf{s}_1$, $\mathbf{s}_2$, $\Delta$, $t$, $W_S$, $\alpha$, $\epsilon$):
1:  $\bar{\mathrm{R}}_\mathrm{f} \leftarrow$ BiBGM($\mathbf{s}_1$, $\mathbf{s}_2$, $\Delta$)
2:  $\bar{\mathbf{s}}_1 \leftarrow ()$, $\bar{\mathbf{s}}_2 \leftarrow ()$
3:  for $i = 0 : 1 : \lfloor \frac{t}{W_S} \rfloor - 1$
4:      $a_1 \leftarrow \mathbf{s}_1 \cap [2i(W_S + \alpha),\ 2i(W_S + \alpha) + W_S]$
5:      $a_2 \leftarrow a_1 - i(W_S + 2\alpha)$
6:      $\bar{\mathbf{s}}_1 \leftarrow \bar{\mathbf{s}}_1 \oplus a_2$
7:      $a_1 \leftarrow \mathbf{s}_2 \cap [(2i+1)(W_S + \alpha),\ (2i+1)(W_S + \alpha) + W_S]$
8:      $a_2 \leftarrow a_1 - i(W_S + 2\alpha) - (W_S + \alpha)$
9:      $\bar{\mathbf{s}}_2 \leftarrow \bar{\mathbf{s}}_2 \oplus a_2$
10: end
11: $\bar{\tau} \leftarrow$ BiBGM($\bar{\mathbf{s}}_1$, $\bar{\mathbf{s}}_2$, $\Delta$)
12: return $\begin{cases} \mathcal{H}_1 & \text{if } \bar{\mathrm{R}}_\mathrm{f} \geq \bar{\tau} + \epsilon \\ \mathcal{H}_0 & \text{o.w.;} \end{cases}$
```

$*$ $\mathbf{s}_i \cap [t_1,\ t_2]$ is a subsequence of $\mathbf{s}_i$ consisting of the epochs in $[t_1,\ t_2]$.
$*$ For a sequence $(x_i)_{i=1}^{\infty}$ and a real number r, $(x_i)_{i=1}^{\infty} - r \triangleq (y_i)_{i=1}^{\infty}$ where $y_i = x_i - r,\ \forall i$.

3) Take the epochs of $\mathbf{s}_2$ in $[W_S + \alpha,\ 2W_S + \alpha]$, subtract $W_S + \alpha$ from the epochs, and add them to $\bar{\mathbf{s}}_2$.
4) For $i = 1,\ 2, \ldots, \lfloor \frac{t}{2(W_S+\alpha)} \rfloor - 1$:
    a) Take the epochs of $\mathbf{s}_1$ in $[2i(W_S + \alpha),\ 2i(W_S + \alpha) + W_S]$, subtract $i(W_S + 2\alpha)$ from the epochs, and add them to $\bar{\mathbf{s}}_1$.
    b) Take the epochs of $\mathbf{s}_2$ in $[(2i+1)(W_S+\alpha),\ (2i+1)(W_S+\alpha)+W_S]$, subtract $i(W_S+2\alpha)+(W_S+\alpha)$ from the epochs, and add them to $\bar{\mathbf{s}}_2$.

Given $(\mathbf{s}_i)_{i=1}^2$ in $[0,\ t]$, AFD employs ITA and operates as follows:

1) Run BiBGM on $(\mathbf{s}_i)_{i=1}^2$, and let $\bar{\mathrm{R}}_\mathrm{f}(t)$ denote the resulting $\bar{\mathrm{R}}_\mathrm{f}$.
2) Run ITA on $(\mathbf{s}_i)_{i=1}^2$ to generate $(\bar{\mathbf{s}}_i)_{i=1}^2$, and run BiBGM on $(\bar{\mathbf{s}}_i)_{i=1}^2$. Let $\bar{\tau}(t)$ denote the resulting $\bar{\mathrm{R}}_\mathrm{f}$.
3) If $\bar{\mathrm{R}}_\mathrm{f}(t) \geq \bar{\tau}(t) + \epsilon$, declare $\mathcal{H}_1$; otherwise, declare $\mathcal{H}_0$.

If $\mathcal{H}_0$ is true, $\bar{\mathrm{R}}_\mathrm{f}(t)$ and $\bar{\tau}(t)$ are expected to be close. Instead of a predetermined threshold τ, AFD uses an adaptive threshold $\bar{\tau}(t)+\epsilon$, where ϵ is added to allow a small difference between $\bar{\mathrm{R}}_\mathrm{f}(t)$ and $\bar{\tau}(t)$ under $\mathcal{H}_0$. If $\mathcal{H}_1$ is true and the relative flow rate is high enough, $\bar{\mathrm{R}}_\mathrm{f}(t)$ is expected to be greater then $\bar{\tau}(t)$. Implementation of AFD is given in Table. II. It contains ITA in the lines 2-10. The computational complexity of AFD is $O(|\mathbb{S}_1| + |\mathbb{S}_2|)$.

In ITA, the number of epochs in $(\bar{\mathbf{s}}_i)_{i=1}^2$ is at most a half of that of the original measurements. Fig. 4 describes a heuristic (which we refer to as ITAh) to double the number of epochs in $(\bar{\mathbf{s}}_i)_{i=1}^2$. Although no further analysis is provided, numerical results in Section V show that this heuristic leads to a better performance of AFD. In the upcoming analysis and the simulations in Section V, AFD employs ITA, not ITAh, unless otherwise specified.

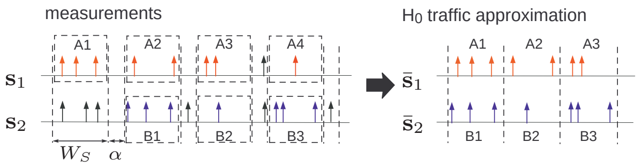

Fig. 4. ITAh: Unlike ITA, we do not throw away $A2$, $A4, \ldots$ and $B2$, $B4, \ldots$. Here, $A1$, $A2$, $A3, \ldots$ and $B1$, $B2$, $B3, \ldots$ are cut from the measurements and assembled to approximate $\mathcal{H}_0$ traffic.

In the rest of this section, we present a theorem stating the performance of AFD for a sufficiently large t. Suppose that $\mathbf{S}_1$ and $\mathbf{S}_2$ are Poisson processes. Under $\mathcal{H}_1$, let $\mathbf{F}_i$ and $\mathbf{W}_i$ denote the flow and chaff part of $\mathbf{S}_i$ respectively, and suppose the following[7] are true:

1) $\mathbf{F}_1$ and $\mathbf{F}_2$ are built by two independent Poisson processes ($\mathbf{F}_1^{12}$, $\mathbf{F}_2^{21}$) and two sequences of delay[8] variables ($(\alpha_i)_{i=1}^{\infty}$, $(\beta_i)_{i=1}^{\infty}$), where $\mathbf{F}_1 = \mathbf{F}_1^{12} \oplus sort(\{\mathbf{F}_2^{21}(i) + \beta_i,\ i = 1, 2, \ldots\})$ and $\mathbf{F}_2 = \mathbf{F}_2^{21} \oplus sort(\{\mathbf{F}_1^{12}(i) + \alpha_i,\ i = 1, 2, \ldots\})$.
2) $\mathbf{W}_1$, $\mathbf{W}_2$, $\mathbf{F}_1^{12}$, and $\mathbf{F}_2^{21}$ are independent.
3) $(\alpha_i)_{i=1}^{\infty}$ and $\mathbf{W}_1$ are independent, $(\beta_i)_{i=1}^{\infty}$ and $\mathbf{W}_2$ are independent, and $(\alpha_i)_{i=1}^{\infty}$, $(\beta_i)_{i=1}^{\infty}$, $\mathbf{F}_1^{12}$, and $\mathbf{F}_2^{21}$ are independent.

Under these assumptions, the below theorem states the consistency of AFD when there are rate changes during the observation interval.

Theorem 3.1: Let $\rho_1, \ldots, \rho_m \in (0, 1)$ be m fixed constants satisfying $\sum_{i=1}^m \rho_i = 1$. Suppose that in $[(\sum_{i=1}^{k-1} \rho_i)t,\ (\sum_{i=1}^{k} \rho_i)t]$, $\mathbf{S}_1$ and $\mathbf{S}_2$ have the rates $\lambda_1^{(k)}$ and $\lambda_2^{(k)}$ respectively, and $\lambda^{(k)} \triangleq \frac{\lambda_1^{(k)}+\lambda_2^{(k)}}{2}$. Suppose that, under $\mathcal{H}_1$, R_f is greater than $\sigma + \epsilon$ a.s., where

$$\sigma \triangleq \frac{\sum_{i=1}^m \lambda^{(i)} \rho_i \gamma_i}{\sum_{i=1}^m \lambda^{(i)} \rho_i}$$

and

$$\gamma_i \triangleq \begin{cases} \frac{2\lambda_1^{(i)}\lambda_2^{(i)}(e^{2\Delta\lambda_2^{(i)}} - e^{2\Delta\lambda_1^{(i)}})}{(\lambda_2^{(i)}+\lambda_1^{(i)})(\lambda_2^{(i)} e^{2\Delta\lambda_2^{(i)}} - \lambda_1^{(i)} e^{2\Delta\lambda_1^{(i)}})} & \text{if } \lambda_1^{(i)} \neq \lambda_2^{(i)} \\ \frac{2\lambda\Delta}{1+2\lambda\Delta} & \text{if } \lambda_1^{(i)} = \lambda_2^{(i)} = \lambda. \end{cases}$$

Then, if t goes to infinity, the miss detection probability of AFD with ϵ vanishes and its false alarm probability decays exponentially fast.

Sketch of Proof: Let $\tilde{t}$ denote the time length of $(\bar{\mathrm{s}}_i)_{i=1}^2$. Then, $\tilde{t} = \lfloor \frac{t}{2(W_S+\alpha)} \rfloor W_S$. Due to the traffic assumptions, $\bar{\mathbf{S}}_1$ and $\bar{\mathbf{S}}_2$ are independent nonhomogeneous Poisson processes, regardless of the true hypothesis. In addition, for sufficiently large t, the rates of $\bar{\mathbf{S}}_1$ and $\bar{\mathbf{S}}_2$ in $[(\sum_{i=1}^{k-1} \rho_i)\tilde{t},\ (\sum_{i=1}^{k} \rho_i)\tilde{t}]$

[7] The performance of AFD depends on how well $(\bar{\mathbf{S}}_i)_{i=1}^2$ approximate $\mathcal{H}_0$ traffic. However, under $\mathcal{H}_1$, the fact that a flow exists does not give enough detail about the correlation between $\mathbf{S}_1$ and $\mathbf{S}_2$, which affects the quality of the $\mathcal{H}_0$ approximation. Hence, to assess the performance of AFD, we impose more assumptions to further specify the correlation.

[8] $\alpha_i, \beta_i \in [0,\ \Delta]$ a.s. $, \forall i$.

are $\lambda_1^{(k)}$ and $\lambda_2^{(k)}$ respectively[9]. Therefore, corollary 4.1 in [7] implies[10] that $\bar{\tau}(t)$ converges to σ a.s..

(i) False alarm probability: The false alarm probability is[11]

$$\begin{aligned} & P_F(t) = P_0(\bar{\mathrm{R}}_{\mathrm{f}}(t) \geq \bar{\tau}(t) + \epsilon) \\ & \leq P_0(\bar{\mathrm{R}}_{\mathrm{f}}(t) \geq \sigma + \tfrac{\epsilon}{2}) + P_0(\bar{\mathrm{R}}_{\mathrm{f}}(t) < \sigma + \tfrac{\epsilon}{2}, \, \bar{\mathrm{R}}_{\mathrm{f}}(t) - \epsilon \geq \bar{\tau}(t)) \\ & \leq P_0(\bar{\mathrm{R}}_{\mathrm{f}}(t) \geq \sigma + \tfrac{\epsilon}{2}) + P_0(\sigma - \tfrac{\epsilon}{2} > \bar{\tau}(t))) \end{aligned}$$

By following the proof procedure of theorem 6.4 in [3], Sanov's theorem [8] can be used to show that both terms in the last line decay exponentially fast.

(ii) Miss detection probability: Under $\mathcal{H}_1$, the optimality of BiBGM implies that $\bar{\mathrm{R}}_{\mathrm{f}}(t) \geq \mathrm{R}_{\mathrm{f}}(t)$ a.s.. Therefore, $\liminf_{t\to\infty} \bar{\mathrm{R}}_{\mathrm{f}}(t) \geq \mathrm{R}_{\mathrm{f}} > \sigma + \epsilon$ a.s.. In addition, $\bar{\tau}(t)$ converges to σ a.s.. Hence, the miss detection probability

$$P_M(t) = P_1(\bar{\mathrm{R}}_{\mathrm{f}}(t) < \bar{\tau}(t) + \epsilon)$$

vanishes as t goes to infinity. ■

Corollary 4.1 in [7] implies that if $\mathbf{S}_1$ and $\mathbf{S}_2$ are independent Poisson processes with rates $\lambda_1^{(i)}$ and $\lambda_2^{(i)}$ respectively, then $\lim_{t\to\infty} \bar{\mathrm{R}}_{\mathrm{f}}(t) = \gamma_i$. Note that σ is the weighted mean of γ_is, where the weight of γ_i is the ratio of the number of epochs in the ith interval to the number of total epochs.

IV. Monitoring Algorithm

For continuous monitoring, we propose a sliding window technique utilizing AFD as its building block, which we refer to as Adaptive Flow Monitor (AFM). AFM has two integer parameters, the sliding window size (W) and the test period (β). AFM repeatedly executes AFD over the W most recent samples, which we refer to as the *observation window*, whenever new β samples arrive. AFM is aimed at detecting the presence of a flow in the observation window.

At every β sample arrivals, AFM executes the following:

1) Update $(\mathbf{s}_i)_{i=1}^2$ by adding new β epochs, and update $(\bar{\mathbf{s}}_i)_{i=1}^2$ accordingly using ITA.
2) Run BiBGM over the updated portion of $(\mathbf{s}_i)_{i=1}^2$ and $(\bar{\mathbf{s}}_i)_{i=1}^2$ to find new matches.
3) Based on matches, calculate $\bar{\mathrm{R}}_{\mathrm{f}}$ using only the epochs in the observation interval. Calculate $\bar{\tau}$ using only the epochs in $(\bar{\mathbf{s}}_i)_{i=1}^2$ whose original epochs before ITA are in the observation interval.
4) Declare $\mathcal{H}_1$ if $\mathrm{R}_{\mathrm{f}} \geq \bar{\tau} + \epsilon$; otherwise, declare $\mathcal{H}_0$.

The computational complexity of AFM is linear with respect to the number of all the monitored samples.

V. Numerical Results

A. Numerical Results: AFD

We first use the synthetic Poisson traffic with varying rates to test the performance of AFD. In the first half of samples, $\mathbf{S}_1$ and $\mathbf{S}_2$ have the rates $\lambda_1^{(1)}$ and $\lambda_2^{(1)}$, and in the other half, the rates are $\lambda_1^{(2)}$ and $\lambda_2^{(2)}$. For $\mathcal{H}_0$ traffic, we generated the realizations of two independent Poisson processes. For $\mathcal{H}_1$ traffic, $\mathbf{S}_i = \mathbf{F}_i^{12} \oplus \mathbf{W}_i$: $\mathbf{W}_1$, $\mathbf{W}_2$, and $\mathbf{F}_1^{12}$ are independent Poisson processes, and $\mathbf{F}_2^{12} = sort(\{\mathbf{F}_1^{12}(i) + \alpha_i, \, i = 1, 2, \ldots\})$ where delays (α_i) are i.i.d. and uniformly distributed over $[0, \Delta]$. The fraction of chaff[12] (f_c) is 0.4 and 0.7 for the first half samples and the second half samples, respectively.

TABLE III
AFD on MSN VoIP traffic: $W_S = 2$, $\alpha = \Delta = 0.15$, $\epsilon = 0.05$.
Number of experiments: 160, 80, and 40 for sample size 5000, 10000, and 20000, respectively.
Total traffic rates: $\lambda_1 = 26.80$, $\lambda_2 = 34.93$. FTP data rate: 11.11.

sample size	P_F (ITA)	P_M (ITA)	P_F (ITAh)	P_M (ITAh)
5000	0.1000	0.1500	0.0875	0.1063
10000	0.0375	0.0625	0.0375	0.075
20000	0	0.075	0	0.025

Fig. 5 shows the ROC curves of AFD. The ROC curves are obtained by plotting the false alarm probability (x axis) and the detection probability (y axis) of AFD with ϵ, while increasing ϵ from 0 to 1 by 0.01. We tested two different approximation procedures, ITA (solid line) and ITAh (dashed line). The ROC curves imply that ITAh results in a better performance. In AFD, $\bar{\tau}(t)$ plays a role of an estimate of the threshold τ in BFD, and it is obtained by running BiBGM over $(\bar{\mathbf{s}}_i)_{i=1}^2$. Compared to ITA, ITAh uses twice more samples to obtain $\bar{\tau}(t)$, so it is natural to expect that ITAh would give a better performance. As the sample size increases, the ROC curves moves to the upper left corner implying a better detection performance.

We also test AFD using the MSN VoIP traffic, which is a representative example of traffic with a delay constraint. As described in Fig. 1, we located one laptop (R_1) in one room and two laptops (R_3, R_4) in another room. R_1 is connected to a wireless LAN via the access point R_0, and R_3 and R_4 are connected to a wireless LAN via the access point R_2, where two access points are using different channels. Under $\mathcal{H}_1$, R_1 has an MSN VoIP call with R_3, and R_4 downloads a file (with 20kB/s rate limit) from an FTP server in our laboratory. Under $\mathcal{H}_0$, R_1 and R_3 make independent VoIP calls, and R_4 downloads a file from the same server. We recorded[13] the transmission epochs of R_1 ($\mathbf{s}_1$) and those of the access point R_2 ($\mathbf{s}_2$). $\mathbf{s}_1$ consists of MSN VoIP packets and control/management packets, and $\mathbf{s}_2$ consists of MSN VoIP packets for R_3, FTP data packets for R_4, and control/management packets (except beacon packets). Table III shows the result of the experiment. The result implies that AFD works reasonably well for the MSN VoIP traffic, and it works well as a heuristic to detect a flow in traffic with unknown characteristics.

[9] In general, there exists small intervals (with length less than W_S) around $(\sum_{i=1}^{(k)} \rho_i)\tilde{t}$, $k = 1, \ldots, m-1$, in which the rates will disagree with this statement. However, their effect vanishes as t increases.

[10] Note that $\widehat{\mathrm{CTR}}(t)$ in [7] is equivalent to $1 - \bar{\mathrm{R}}_{\mathrm{f}}(t)$.

[11] P_i denotes the probability measure conditioning on that $\mathcal{H}_i$ is true.

[12] $f_c \triangleq \frac{\text{chaff transmission rate}}{\text{total traffic rate}}$.

[13] Window Live Messenger 2009 (14.0.8089.726) was used for MSN VoIP calls, and Wireshark (ver 1.2.6) network protocol analyzer was used to collect the timing measurements.

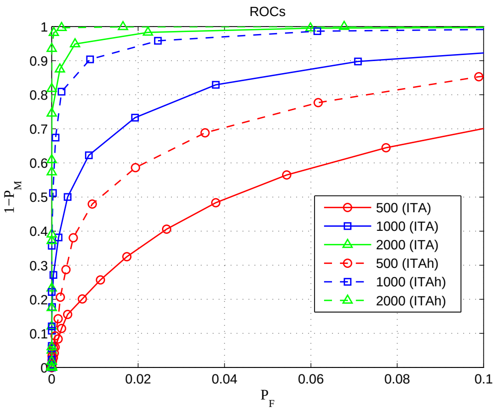

Fig. 5. ROC curves of AFD. $W_S = 2$, $\alpha = \Delta = 0.1$, $\lambda_1^{(1)} = \lambda_2^{(1)} = 10$, $\lambda_1^{(2)} = \lambda_2^{(2)} = 20$, $f_c^{(1)} = 0.4$, $f_c^{(2)} = 0.7$, 10000 Monte Carlo runs.

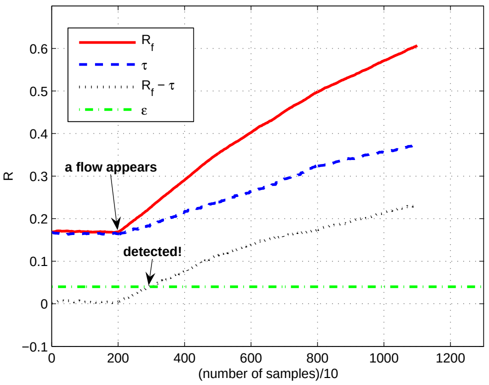

Fig. 6. AFM on Poisson traffic: $W = 12000$, $\beta = 10$, $\alpha = \Delta = 0.1$, $W_S = 2$, $\epsilon = 0.04$. Until the 2000th sample, $\lambda_1 = \lambda_2 = 2$. At the 2000th sample, chaff rates of both nodes decrease by 1, and a flow with rate 9 appears ($(\lambda_1, \lambda_2) = (10, 10)$, $f_c = 0.1$). At the 5000th sample, R_1 increases the chaff transmission rate by 5 ($(\lambda_1, \lambda_2) = (15, 10)$, $f_c = 0.28$). At the 8000th sample, the flow rate decreases by 5 ($(\lambda_1, \lambda_2) = (10, 5)$, $f_c = 0.47$).

B. Numerical Results: AFM

In Fig. 6, the sample paths of $\bar{\mathrm{R}}_{\mathrm{f}}$, $\bar{\tau}$, and $\bar{\mathrm{R}}_{\mathrm{f}} - \bar{\tau}$ are given to visualize how these statistics of AFM change when the traffic characteristics change dynamically. When there is no flow, $\bar{\mathrm{R}}_{\mathrm{f}}$ and $\bar{\tau}$ are almost same. However, once a flow appears, they begin to diverge from each other.

We looked at two metrics to measure the performance of AFM. First, we consider the average number of samples (T_F) that AFM observes until it generates the first false alarm when it is run over the $\mathcal{H}_0$ traffic. T_F can tell us how often AFM would generate false alarms. Second, we look at the average of the detection delay (T_D) which is defined as the number of samples that AFM observes to detect a flow[14]. To numerically obtain T_F, we generated the realizations of independent Poisson processes $\mathbf{S}_1$ and $\mathbf{S}_2$ with rates λ_1 and λ_2, and ran AFM on them. With a period of 1000 sample arrivals, (λ_1, λ_2) rotates among $(10,\ 10)$, $(10,\ 20)$, $(20,\ 20)$, and $(20,\ 10)$. For T_D, we first generated the realizations of independent Poisson processes with $\lambda_1 = \lambda_2 = 12$. Then, we made a flow to appear at a certain time ($\lambda_1 = \lambda_2 = 20$, $f_c = 0.2$ or 0.6), and measured the detection delay.

TABLE IV
AFM on Poisson traffic: $W_S = 2$, $\alpha = \Delta = 0.1$, $\epsilon = 0.04$, $\beta = 10$.
Monte Carlo runs: 10000 for T_D, 1000 for T_F.

W	T_F	T_D $(f_c = 0.2)$	T_D $(f_c = 0.6)$
2000	5217.4	532.2	1180.5
4000	37111.0	1156.9	2778.7
8000	923330	2476.2	5961.9

Table IV contains the result. The increase in f_c or W results in longer T_D. This is reasonable because such changes makes AFD, the building block of AFM, less sensitive to the appearance of a flow. As W increases, T_F also increases, and it increases much faster than T_D. Such fast increasement of T_F seems to agree with the exponential dacay of AFD's false alarm probability (as the sample size grows). When $W = 8000$, T_F is 923330, and it means that AFM takes 30778 seconds (8.55 hours) to generate the first false alarm on average.

VI. Conclusion

This paper studied timing-based detection of time-varying flows in wireless networks. We proposed a practical algorithm to detect a flow contained in the traffic with varying rates and unknown characteristics. Then, we presented a sliding window technique for continuous monitoring. Our algorithms require only the transmission timings of nodes, which are easily available in wireless networks. We tested the algorithms using the MSN VoIP traffic and the synthetic Poisson traffic, and the results are encouraging.

References

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

[2] 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 *5th International Symposium on Recent Advances in Intrusion Detection, Lecture Notes in Computer Science 2516*, 2002.

[3] T. He and L. Tong, "Detection of Information Flows," *IEEE Trans. Inf. Theory*, vol. 54, pp. 4925–4945, Nov. 2008.

[4] L. Zhang, A. Persaud, A. Johnson, and Y. Guan, "Detection of Stepping Stone Attack under Delay and Chaff Perturbations," in *Proc. of The 25th IEEE International Performance Computing and Communications Conference*, Phoenix, AZ, Apr. 2006.

[5] A. Blum, D. Song, and S. Venkataraman, "Detection of Interactive Stepping Stones: Algorithms and Confidence Bounds," in *Conference of Recent Advance in Intrusion Detection (RAID)*, Sophia Antipolis, French Riviera, France, September 2004.

[6] T. He and L. Tong, "Detecting Information Flows: Improving Chaff Tolerance by Joint Detection," in *Proc. 2007 Conference on Information Sciences and Systems*, Baltimore, MD, March 2007.

[7] J. Kim and L. Tong, "Timing-based Detection of Packet Forwarding in MANETs," *Accepted to the 11th International Workshop on Signal Processing Advances in Wireless Communications*, Marrakech, Morocco, June 2010, http://acsp.ece.cornell.edu/pubC.html.

[8] T. Cover and J. Thomas, *Elements of Information Theory, 2nd Edition*. Wiley, 2006.

[14] If the flow appears at the ith sample, and AFM detects it by observing until $\bar{i}$th sample, then $\bar{i} - i$ is a detection delay.