Identifying 802.11 Traffic From Passive Measurements Using Iterative Bayesian Inference

Wei Wei, *Member, IEEE*, Sharad Jaiswal, Jim Kurose, *Fellow, IEEE*, Don Towsley, *Fellow, IEEE, ACM*, Kyoungwon Suh, *Member, IEEE*, and Bing Wang, *Member, IEEE*

***Abstract*—In this paper, we propose a classification scheme that differentiates Ethernet and WLAN TCP flows based on measurements collected passively at the edge of a network. This scheme computes two quantities, the fraction of wireless TCP flows and the degree of belief that a TCP flow traverses a WLAN inside the network, using an iterative Bayesian inference algorithm that we developed. We prove that this iterative Bayesian inference algorithm converges to the unique maximum likelihood estimate (MLE) of these two quantities. Furthermore, it has the advantage that it can handle any general K-classification problem given the marginal distributions of these classes. Numerical and experimental evaluations demonstrate that our classification scheme obtains accurate results. We apply this scheme to two sets of traces collected from two campus networks: one set collected from UMass in mid 2005 and the other collected from UConn in late 2010. Our technique infers that 4%–7% and 52%–55% of incoming TCP flows traverse an IEEE 802.11 wireless link in these two networks, respectively.**

***Index Terms*—IEEE 802.11 wireless LAN, iterative Bayesian inference, TCP ACK-pairs, wireless traffic detection.**

I. Introduction

THE DEPLOYMENT and use of IEEE 802.11 wireless LANs (WLANs) has grown dramatically over the past several years. The presence of a wireless infrastructure within a network, however, raises issues such as the placement and management of wireless access points, maintaining network security, and monitoring the end-to-end performance of wireless users. Identifying wireless traffic has several practical applications for network administrators. It is useful to know the extent of wireless usage in order to allocate resources (such as access points) within the network. Detecting wireless usage at previously unknown locations in the network can detect unauthorized wireless networks that are potential security holes (since they may allow unauthorized access within the network). Finally, by monitoring the identified wireless traffic, one can infer the end-to-end performance of flows, thus keeping tabs on the performance of the wireless network.

Identifying wireless traffic within a network is, however, not an easy task. Wireless access points are invisible to topology discovery tools such as *traceroute* (since they do not reduce the TTL of a packet). Moreover, the IP address of a host may not provide any useful information about the type of its access network. This is because a network administrator may not allocate separate IP address pools for wired and wireless hosts. Even if there were separate pools, a host with an address from the wired address pool may act as a NAT box for a set of wireless hosts or install a wireless router and become a wireless host.

One approach toward monitoring WLAN traffic and estimating the extent of WLAN traffic is to monitor and compute statistics from all access points in the network. This approach, however, requires access to perhaps hundreds of such access points within a large network, many of them unknown, thus making this technique infeasible and impractical. In this paper, we present a novel methodology to detect TCP flows that have traversed an 802.11 WLAN using measurements collected *passively* at the *edge* of a network. This methodology only requires a single monitor, and hence incurs little deployment cost, and is easy to manage and maintain.

The contributions of our work are as follows. We propose a classification scheme to differentiate Ethernet and WLAN TCP flows based on measurements collected passively at the edge of a network. This scheme takes the interarrival times of TCP *ACK-pairs*[1] as input and computes the fraction of wireless TCP flows. Furthermore, for each TCP flow, it determines the belief that this flow traverses a WLAN inside the network. The core of this classification scheme is an iterative Bayesian inference algorithm that we develop to obtain the maximum likelihood estimate (MLE) of these quantities. This inference algorithm can handle any general K-classification problem $(K \geq 2)$ given the marginal distributions of these classes. Numerical and experimental evaluations demonstrate that our classification scheme obtains accurate results.

We apply the classification scheme to various traces collected between April and May 2005 by a monitoring device placed at

Manuscript received August 22, 2009; revised January 10, 2011 and June 01, 2011; accepted June 07, 2011; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor M. Allman. Date of publication July 07, 2011; date of current version April 12, 2012. This work was supported in part by the National Science Foundation under NSF Grants ANI-0325868, ANI-0240487, ANI-0085848, EIA-0080119, and NSF CAREER Award 0746841, ARO W911NF-08-1-0233, and a 2009 Caterpillar Scholar Award. A preliminary version of this paper appeared in the IEEE Conference on Computer Communications (INFOCOM), Barcelona, Spain, April 23–29, 2006.

W. Wei, J. Kurose, and D. Towsley are with the University of Massachusetts, Amherst, Amherst, MA 01003 USA (e-mail: weiwei@cs.umass.edu; kurose@cs.umass.edu; towsley@cs.umass.edu).

S. Jaiswal is with Alcatel-Lucent Bell Labs, Bangalore 560 045, India (e-mail: jsharad@alcatel-lucent.com).

K. Suh is with Illinois State University, Normal, IL 61790 USA (e-mail: kwsuh@ilstu.edu).

B. Wang is with the University of Connecticut, Storrs, CT 06269 USA (e-mail: bing@engr.uconn.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNET.2011.2159990

[1] Informally, an ACK-pair refers to two ACKs generated in response to data packets that arrive close in time at the measurement point. A more precise definition will be given in Section III.

the gateway router of the University of Massachusetts, Amherst (UMass) campus network. Our scheme infers that between 4%–7% of all TCP flows entering the UMass campus traverse an 802.11 wireless link within the campus. It also detects wireless usage (through the use of private routers and access points) in areas that are not covered by the official wireless infrastructure. We further apply this scheme to various traces collected in December 2010 from the University of Connecticut, Storrs (UConn) campus network and infer a prevalent amount of incoming TCP flows (between 52%–55%) traverses a WLAN inside the UConn campus.

The rest of the paper is organized as follows. Section II describes related work. Section III presents the problem setting and a high-level description of our approach. Section IV presents the analytical foundation of our classification scheme. Sections V and VI present our iterative Bayesian inference algorithm and classification scheme, respectively. Numerical and experimental evaluations of our classification scheme are presented in Section VII. Section VIII describes the inference results using data gathered from the two campus networks. Finally, Section IX briefly discusses using our scheme in future networks, and Section X concludes the paper.

II. Related Work

The study [35] is most closely related to our work. Both studies utilize measurements collected passively at the edge of a network to differentiate Ethernet and WLAN traffic. They, however, differ in important aspects: The study [35] makes a *deterministic* decision on whether a flow traverses a WLAN or not (after obtaining a sufficient number of observations), while our current study makes a *probabilistic* decision on the belief that a flow traverses a WLAN and infers the fraction of flows that traverse a WLAN (more generally, it infers the belief that a flow belongs to a certain class and the fraction of each class of traffic). The iterative Bayesian inference algorithm that we develop in this study obtains the unique MLE of the above two quantities, and hence is asymptotically optimal. Although one may use the approach in the study [35] to obtain the fraction of WLAN traffic, it is not MLE and hence does not have the provably optimal properties of MLE.

Detecting wireless traffic has also been studied in several other efforts. Cheng and Marsic classify hosts to be behind either wired or wireless networks based on the round-trip times (RTTs) of TCP connections [12]. Their study relies on certain assumptions about wireless links, such as very low bandwidth and high loss rates, which may not hold in current WLANs. Wei *et al.* propose a simple and efficient end–end scheme to classify an access network into three types: Ethernet, WLAN, or low-bandwidth connections [36]. Different access networks were classified based on cutoff values (derived based on intrinsic properties of these networks) of median and entropy of the interarrival times of the injected UDP packet pairs. Baiamonte *et al.* [5] use entropies to detect wireless traffic based on traffic collected at an aggregation point. Beyah *et al.* [8] use visual inspection to detect wireless hosts. Mano *et al.* [24] propose a technique that segments large packets into smaller ones to detect wireless traffic. All of the above studies detect wireless traffic on a per-host basis. None of them provides the MLE of the extent of wireless traffic, which is one focus of our study.

This study extends our preliminary version [34]. Although packet interarrival times, ACK interarrival times, or TCP probes have been used to detect a shared bottleneck [21], [27], estimate link capacity [10], [15], [16], [18], [22], or estimate bottleneck bandwidth [26], our preliminary version [34] is the first one that defines ACK-pairs and uses the interarrival time of ACK-pairs to differentiate WLAN and Ethernet traffic. The current version considers K-classification that can classify TCP flows into K classes ($K \geq 2$), and hence significantly generalizes the scheme in the preliminary version (which only considers two classes). Furthermore, our current version includes a new set of measurements collected in December 2010, which contains a much richer variety of WLAN traffic (including 802.11a/b/g/n) than that in the preliminary version.

Broadly, our work is related to WLAN measurement and management. Previous work on wireless measurement has used direct measurement techniques and focused on the performance and user behavior in wireless networks (e.g., [6], [7], [17], [29], and [32]). We use an indirect approach to *infer* the percentage of the WLAN traffic and the belief that a flow traverses a WLAN. Most studies on WLAN management rely on distributed monitors that monitor RF airwaves [3], [4], [13], [23], [30], [37], [38]. The rationale is that RF airwave monitoring provides detailed low-level (i.e., PHY and MAC) information that is critical for analyzing the behavior of a network and troubleshooting faults in a network. Our study takes the approach of centralized monitoring at a single aggregation point. The captured information is at higher layers (i.e., IP and transport layers), and hence may not provide sufficient insights for troubleshooting. However, our approach incurs little deployment and maintenance cost, and our study on estimating the extent of wireless traffic and the belief that a flow traverses a WLAN has a number of applications in WLAN management: It is helpful for resource allocation, detecting wireless usage at previously unknown places, and keeping tabs on the performance of wireless network.

Lastly, passive measurement at a single aggregation point falls broadly into "measurement-in-the-middle," i.e., measurements are taken at a single point in the "middle" of the end-to-end connections. The studies of [19] and [20] infer end-to-end properties of a TCP connection through measurement-in-the-middle. Our study differs in that we focus on detecting wireless traffic.

III. Problem Definition and Approach

We now state our inference problem and describe, at a high level, our approach toward solving this problem. Consider a local network, e.g., a university campus or an enterprise network, as illustrated in Fig. 1. End-hosts within this network use either wired Ethernet or 802.11 WLAN to access the Internet. A monitoring point is located at the edge of this local network (e.g., at the gateway router), capturing traffic coming in and going out of the network. Our goal is to determine: 1) what fraction of TCP flows, observed by the passive monitor, pass through a WLAN within the network; and 2) for each TCP flow,

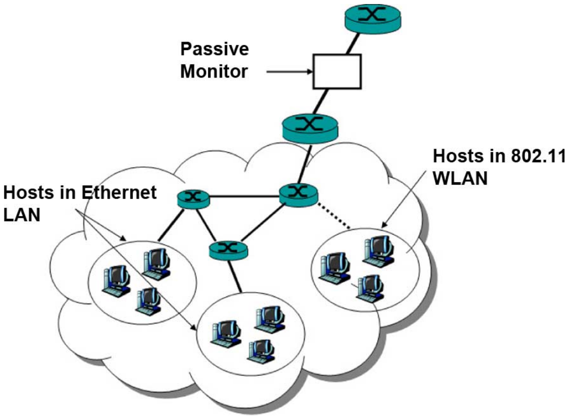
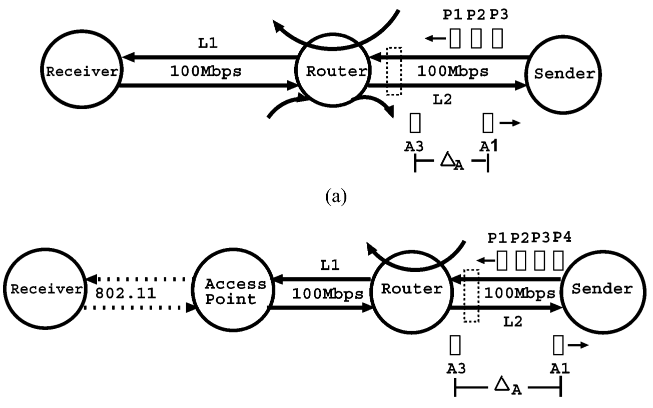


Fig. 1. Problem setting: A monitoring point is located at the gateway router of a local network, capturing traffic coming in and going out of the network. The end-hosts within this network are behind wired Ethernet or 802.11 WLAN.



(a)

(b)

Fig. 2. Settings for the analysis: (a) Ethernet and (b) WLAN (802.11b or 802.11g). The dashed rectangle between the sender and the router represents the monitoring point. The pair of ACKs, A_1 and A_3, form an ACK-pair.

what is the belief (probability) that this TCP flow traverses a WLAN within the network.[2]

This problem is challenging since the monitoring point is at the edge of the network, in the middle of the path between a sender and a receiver, and hence the measurements collected at the monitoring point may not provide accurate information on the two end-hosts. Furthermore, the monitoring point only *passively* collects traffic, which limits the types of techniques that can be utilized to solve the above problem.

Our approach utilizes the intrinsic characteristics of WLAN and Ethernet connections and operates roughly as follows. For each TCP flow, we identify pairs of TCP data packets destined to a receiver (end-host) within the local network and arriving at the monitoring point close in time. A pair of ACKs in response to these data packets (termed as *ACK-pairs*) are generated by the receiver and returned to the sender. As will be shown in Section IV, the interarrival times of the ACK-pairs at the monitoring point (termed as *inter-ACK times*) differ significantly when the data packets and ACK-pairs traverse a wireless hop as compared to a wired Ethernet link. Our scheme exploits this difference to classify WLAN and Ethernet TCP flows. More specifically, it takes inter-ACK times as input and uses an iterative Bayesian inference algorithm (Section V) to infer the extent of WLAN traffic and the belief that a TCP flow traverses a WLAN.

IV. Analytical Basis

In this section, we carry out an analytical study that forms the foundation of our inference algorithm in Section V. The goal of our analytical study is to answer two key questions: 1) What are appropriate statistics to differentiate WLAN and Ethernet traffic? 2) Can WLAN and Ethernet traffic be differentiated *deterministically* using threshold based schemes?

To answer these two questions, we consider an arbitrary TCP flow from an outside server to a receiver residing in the local network, as shown in Fig. 2. The access link of the receiver is either Ethernet [Fig. 2(a)] or WLAN [Fig. 2(b)]. We refer to these two settings as *Ethernet setting* and *WLAN setting*, respectively. In both settings, a router resides between the sender and the receiver; we model the router as two $M/D/1$ queues, for data packets and ACKs, denoted as Q_{D} and Q_{A}, respectively. The utilization of Q_{D} is ρ_{D}, and the utilization of Q_{A} is ρ_{A}. The router is connected to the sender by link L_2 of 100 Mb/s. The monitoring point is between the sender and the router, tapping into link L_2. In the Ethernet setting, the router and the receiver are connected by link L_1 of 100 Mb/s. In the WLAN setting, an access point resides between the router and the receiver; the access point and the router are connected by link L_1 of 100 Mb/s; the receiver is connected to the access point using 11-Mb/s 802.11b or 54-Mb/s 802.11g.

We assume that the receiver implements delayed ACK policy[3] since this policy is commonly used in practice [2], [28]. To accommodate the effects of delayed ACK, we consider four data packets P_1, P_2, P_3, and P_4, each of 1500 B, sent back to back from the sender. Without loss of generality, we assume that packet P_1 is acknowledged. Since we assume delayed ACK, packet P_3 is also acknowledged. Let A_1 and A_3 denote the ACKs corresponding to P_1 and P_3, respectively. Then, A_1 and A_3 form an ACK-pair. Let Δ_A represent the inter-ACK time of A_1 and A_3 at the monitoring point. Let Δ denote the interarrival time of the data packets P_1 and P_3 at the monitoring point. Then, Δ is approximately $120 \times 2 = 240$ μs since each P_i $(i = 1, \ldots, 4)$ is 1500 B and the bandwidth of link L_2 is 100 Mb/s.

We conjecture that inter-ACK times in a WLAN setting are larger than those in an Ethernet setting. Intuitively, this is due to two reasons. First, in a WLAN, even in the absence of contention, the receiver must wait for a random backoff interval after a previous successful transmission to avoid channel capture (see [36] and the references within). This random backoff delay may be inserted between the ACK-pair, leading to a larger inter-ACK time. Second, in a WLAN, the ACKs also contend with the data packets (coming from the opposite direction) for the wireless channel. Therefore, a data packet may be

[2] Since the monitoring point is located at the edge of the network, it does not observe internal flows in the network. Therefore, our approach is not applicable to internal flows.

[3] That is, a receiver releases an ACK after receiving two packets or if the delayed-ACK timer is triggered after the arrival of a single packet.

transmitted between the ACK-pair, and this again increases the inter-ACK time. Our analytical results confirm this conjecture.

Suppose m samples of inter-ACK time are observed in a TCP flow. Let $\xi_{.5}^{(m)}(\Delta_{\mathrm{A}})$ denote the median inter-ACK time given these m samples (we use median instead of mean since it is less sensitive to outliers in the measurements [25]). We have the following results on median inter-ACK times; the proofs are found in Appendix I.

Theorem 1 (Median Inter-ACK Time for Ethernet): In the Ethernet setting, when $0 < \rho_{\mathrm{D}}, \rho_{\mathrm{A}} \leq 1$, we have $P(\xi_{.5}^{(5)}(\Delta_{\mathrm{A}}) \geq 600\ \mu\mathrm{s}) < 0.04$.

Theorem 2 (Median Inter-ACK Time for 802.11b): In the 802.11b WLAN setting, when $0 < \rho_{\mathrm{D}}, \rho_{\mathrm{A}} \leq 1$, we have $P(\xi_{.5}^{(5)}(\Delta_{\mathrm{A}}) \geq 600\ \mu\mathrm{s}) \approx 1$.

Theorem 3 (Median Inter-ACK Time for 802.11g): In the 802.11g WLAN setting, when $0 < \rho_{\mathrm{D}}, \rho_{\mathrm{A}} \leq 1$, we have $P(\xi_{.5}^{(5)}(\Delta_{\mathrm{A}}) \geq 600\ \mu\mathrm{s}) > 0.41$.

Empirical results in the UConn campus network support the above theorems. In particular, our empirical results show that, for 100-Mb/s Ethernet, $P(\xi_{.5}^{(5)}(\Delta_{\mathrm{A}}) \geq 600\ \mu\mathrm{s}) = 0.03$; for 802.11b and 802.11g WLAN, $P(\xi_{.5}^{(5)}(\Delta_{\mathrm{A}}) \geq 600\ \mu\mathrm{s})$ is 0.92 and 0.75, respectively. These empirical results are consistent with Theorems 1–3, respectively. Furthermore, for 802.11n WLAN, we obtain empirically $P(\xi_{.5}^{(5)}(\Delta_{\mathrm{A}}) \geq 600\ \mu\mathrm{s}) = 0.18$.[4] This value is larger than that of 100-Mb/s Ethernet, which is not surprising due to the following two reasons. First, while the raw data rate of an 802.11n connection in the UConn network is up to 144 Mp/s, much lower throughput is achieved in practice (even in idealized settings, the maximum throughput of 802.11n measured in [31] is below 100 Mp/s). Second, despite its higher bit rate, 802.11n still provides half-duplex channels, and hence ACKs still contend with data packets that come from the opposite direction to access a wireless channel, which can lead to large inter-ACK times.

In summary, the above results demonstrate that median inter-ACK times can be used to differentiate Ethernet and WLAN traffics. However, since their distributions have overlap, deterministic classification using a cutoff value will not provide accurate results.

V. Iterative Bayesian Inference Algorithm

In this section, we propose an iterative Bayesian inference algorithm that probabilistically classifies a set of observations into K classes, $K \geq 2$. This algorithm is applicable to any general K-classification problem. We next describe the algorithm in a generic setting; how to apply it to detect WLAN traffic is deferred to Section VI.

Consider K classes, $K \geq 2$. Let O denote the set of observations, $O = \{x_i\}_{i=1}^{N}$, where x_i is the ith observation. Let X_i denote the random variable corresponding to the ith observation, x_i. Let α_k denote the probability that an observation belongs to the kth class. Then, $\alpha_k \in [0,1]$, and $\sum_{k=1}^{K} \alpha_k = 1$. Let $\mathcal{C}_k$ denote the event that an observation belongs to the kth class. Let p_{ki} denote the probability that the ith observation is x_i given that it belongs to the kth class. That is, $p_{ki} = P(X_i = x_i \mid \mathcal{C}_k)$. We assume that p_{ki} is known *a priori* (e.g., obtained from training sets). Let β_{ki} denote the belief that the ith observation belongs to the kth class given that its value is x_i. That is, $\beta_{ki} = P(\mathcal{C}_k \mid X_i = x_i)$. Then, $\beta_{ki} \in [0,1]$, and $\sum_{k=1}^{K} \beta_{ki} = 1$. Using Bayes rule, we have

$$\begin{aligned}\beta_{ki} &= \frac{P(\mathcal{C}_k, X_i = x_i)}{P(X_i = x_i)}\\ &= \frac{\alpha_k p_{ki}}{\sum\limits_{k=1}^{K} \alpha_k p_{ki}}.\end{aligned} \tag{1}$$

Let $L(\alpha_1, \ldots, \alpha_K \mid O)$ denote the likelihood function of observing the set of observations, O. Assuming independent observations, we have

$$\begin{aligned}L(\alpha_1, \ldots, \alpha_K \mid O) &= \prod_{i=1}^{N} P(X_i = x_i \mid \alpha_1, \ldots, \alpha_K)\\ &= \prod_{i=1}^{N} \left(\sum_{k=1}^{K} \alpha_k p_{ki} \right).\end{aligned} \tag{2}$$

Taking logarithm on both sides of (2), we have

$$\begin{aligned}l(\alpha_1, \ldots, \alpha_K \mid O) &:= \log L(\alpha_1, \ldots, \alpha_K \mid O)\\ &= \sum_{i=1}^{N} \log \left(\sum_{k=1}^{K} \alpha_k p_{ki} \right).\end{aligned} \tag{3}$$

Let $\hat{\alpha}_k$ denote the MLE of $\alpha_k,\ k = 1, \ldots, K$. Let $\hat{\beta}_{ki}$ denote the MLE of $\beta_{ki},\ k = 1, \ldots, K,\ i = 1, \ldots, N$. These MLEs are obtained by solving the following optimization problem:

$$\max_{\alpha_1, \ldots, \alpha_K} \quad \sum_{i=1}^{N} \log \left(\sum_{k=1}^{K} \alpha_k p_{ki} \right) \tag{4}$$

$$\text{s.t.} \quad \sum_{k=1}^{K} \alpha_k = 1. \tag{5}$$

Substituting $\alpha_K = 1 - \sum_{k=1}^{K-1} \alpha_k$ into (4), the objective function transforms to

$$\begin{aligned}&f(\alpha_1, \ldots, \alpha_{K-1})\\ &\quad := \sum_{i=1}^{N} \log \left(\sum_{k=1}^{K-1} \alpha_k p_{ki} + (1 - \sum_{k=1}^{K-1} \alpha_k) p_{Ki} \right).\end{aligned} \tag{6}$$

[4] The empirical result for each connection type is obtained from its corresponding training set (i.e., a set of flows that is known to use the connection type). The training set for 100-Mb/s Ethernet is constructed based on IP addresses (see Section VII). The training set for a specific type of WLAN traffic (802.11b, 802.11g, or 802.11n) is constructed from controlled experiments using three laptops (we cannot obtain these training sets through purely passive measurements since all WLAN flows share the same set of IP addresses). Specifically, we connect these three laptops to 802.11b/g/n-capable access points, first using 802.11b, then using 802.11g, and then using 802.11n. In each setting, the experiment lasts for 20 min (the laptops are used to surf the Web, download files, and stream videos during the experiment). The traces of these laptops collected at the monitoring point during these three controlled experiments form the training sets for 802.11b, 802.11g, and 802.11n WLAN, respectively.

The MLE of α_k can then be obtained by solving a system of equations

$$\frac{\partial f(\alpha_1,\ldots,\alpha_{K-1})}{\partial \alpha_k} = \sum_{i=1}^{N} \frac{p_{ki} - p_{Ki}}{g_i} = 0, \qquad k = 1,\ldots,K-1 \tag{7}$$

where $g_i = \sum_{k=1}^{K-1} \alpha_k p_{ki} + (1 - \sum_{k=1}^{K-1} \alpha_k) p_{Ki}$.

We assume that $\sum_{i=1}^{N}(p_{ki} - p_{ji})^2 > 0$ for $k \neq j$ since otherwise we cannot differentiate classes k and j from the observations. Then, we have the following theorem on the uniqueness of the MLE; the proof is found in Appendix II.

Theorem 4: The MLE of α_k is unique, and hence the MLE of β_{ki} is unique, $k = 1,\ldots,K, i = 1,\ldots,N$.

We can solve the system of equations (7) to obtain $\hat{\alpha}_k$. However, directly solving this system of equations is difficult since each equation contains an $(N-1)$th order polynomial term, and hence the equation cannot be solved using only rational operations and finite root extractions when $(N-1) \geq 5$[9]. We next design an iterative algorithm to solve for $\hat{\alpha}_k$ and $\hat{\beta}_{ki}$. This algorithm is summarized in Algorithm 1 .

Algorithm 1: Iterative Bayesian Inference Algorithm

```
Set initial values for $\alpha_k^{(0)},\ k = 1,\ldots,K$
$m = 1$
repeat
    for $k = 1$ to $K$ do
        for $i = 1$ to $N$ do
            $\beta_{ki}^{(m)} = \frac{\alpha_k^{(m)} p_{ki}}{\sum_{j=1}^{K} \alpha_j^{(m)} p_{ji}}$
        end for
    end for
    for $k = 1$ to $K$ do
        $\alpha_k^{(m+1)} = \frac{1}{N} \sum_{i=1}^{N} \beta_{ki}^{(m)}$
    end for
    $m = m + 1$
until $(|\alpha_k^{(m+1)} - \alpha_k^{(m)}| \leq \epsilon,\ k = 1,\ldots,K)$
```

In the algorithm, $\alpha_k^{(m)}$ and $\beta_{ki}^{(m)}$ denote respectively the values of α_k and β_{ki} in the mth iteration, and $\alpha_k^0 \in (0,1)$ denotes the initial value of α_k, $k = 1,\ldots,K, i = 1,\ldots,N$. This algorithm is an Expectation and Maximization (EM) algorithm [14]. Each iteration contains two steps: In the E-step, we update $\beta_{ki}^{(m)}$ from $\alpha_k^{(m)}$ [following (1)] in order to obtain the expected number of observations in each class; in the M-step, we update $\alpha_k^{(m+1)}$ from $\beta_{ki}^{(m)}$ based on the definition of α_k. The iteration continues until the difference of α_k between two iterations is below a convergence threshold, ϵ, for $k = 1,\ldots,K$. The convergence of this algorithm follows from the convergence property of an EM algorithm [14]. We prove that this algorithm converges to the unique MLE of α_k and β_{ki}, as stated in the following theorem; the proof is found in Appendix III.

Theorem 5: Let $\bar{\alpha}_k = \lim_{m \longrightarrow \infty} \alpha_k^{(m)}$ and $\bar{\beta}_{ki} = \lim_{m \longrightarrow \infty} \beta_{ki}^{(m)}$. The iterative Bayesian inference algorithm converges to the unique $\bar{\alpha}_k$ and $\bar{\beta}_{ki}$. Furthermore, $\bar{\alpha}_k$ is the MLE of α_k, and $\bar{\beta}_{ki}$ is the MLE of $\beta_{ki},\ k = 1,\ldots,K,\ i = 1,\ldots,N$.

VI. Classification Scheme

We design a classification scheme that determines, for a given collection of TCP flows, the fraction of WLAN TCP flows and the belief that a TCP flow traverses a WLAN. The core of this classification scheme is the iterative Bayesian inference algorithm presented in Section V. In our context, the classes are the various Ethernet connection types (10-Mb/s, 100-Mb/s, and 1-Gb/s Ethernet) and 802.11 WLAN.[5] The input to the classifier contains a set of observations and the marginal distribution of each class. An observation is the median inter-ACK time of a *qualified* TCP flow, i.e., a TCP flow with no less than M ACK-pairs, where M is a predefined parameter (we set M to 2 or 5 in our experiments). The reason for using median inter-ACK time is based on our analysis that it is useful for differentiating Ethernet and WLAN connections (Theorems 1–3 in Section IV). Next, we first describe how to obtain ACK-pairs and the marginal distribution of each class, and then describe how the classifier operates.

A. Identifying ACK-Pairs

We refer to two successive ACKs as an ACK-pair if the interarrival time of their corresponding data packets at the monitoring point is less than a threshold T. In our experiments, we set T to 250 or 400 μs based on our analysis in Section IV. In addition to the above condition, we also take account of several practical issues when identifying ACK-pairs. First, we exclude all ACKs whose corresponding data packets have been retransmitted or reordered. We also exclude ACKs due to expiration of delayed-ACK timers (if delayed ACK is implemented) since such an ACK is not released immediately after its corresponding data packet, and hence the interarrival time of this ACK and its previous ACK does not reflect the characteristics of the access link. We use the technique in [33] to infer whether delayed ACK is implemented, which further requires that the inter-ACK time of an ACK-pair to be below 200 ms.

B. Obtaining Marginal Distributions

The marginal distribution of a connection type can be obtained from a *training set*, which contains TCP flows known to use this connection type. Given a training set, we obtain the marginal distribution as follows. We first identify a set of qualified TCP flows (i.e., TCP flows with no less than M ACK-pairs). For each qualified TCP flow, we obtain the median inter-ACK time over all ACK-pairs. Suppose that a set of n qualified TCP flows are identified in the training set. Let x_i denote the median inter-ACK time of the ith qualified TCP flow. The value of x_i is discretized as follows: If x_i is smaller than 1 ms, it is discretized to be a multiple of 50 μs; otherwise, it is discretized to be a multiple of 1 ms. Then, the marginal distribution is obtained from the discretized value of $\{x_i\}_{i=1}^{n}$.

[5] We do not differentiate 802.11a/b/g/n since which protocol a host uses depends on many factors. On the other hand, we verify that our scheme can successfully differentiate a specific type of WLAN traffic (802.11b, 802.11g, or 802.11n) and Ethernet traffic (10-Mb/s, 100-Mb/s, or 1-Gb/s Ethernet) given the marginal distributions of these traffic types. The detailed results are omitted; some results are briefly discussed in Section IX.

Constructing training sets for a local network requires knowledge of the network. We detail how we construct training sets for our experiments in Section VII; training sets for other networks can be constructed in a similar manner.

C. Applying the Classification Scheme

Given a collection of TCP flows, our classifier operates as follows. It first identifies a set of ACK-pairs (using threshold T) and obtains the corresponding inter-ACK times for each TCP flow. It then determines whether a TCP flow is qualified (based on whether the number of ACK-pairs is more than the threshold, M) and identifies a set of qualified TCP flows. For each qualified TCP flow, it obtains the median inter-ACK time of this flow. Finally, the set of median inter-ACK times over all qualified TCP flows and the marginal distribution of each class (obtained from its corresponding training set) are fed into the iterative Bayesian inference algorithm to obtain inference results, i.e., α_k, the probability that a TCP flow is in the kth class (for a give set of TCP flows, it represents the fraction of TCP flows in the kth class), and β_{ki}, the belief (probability) that the ith qualified TCP flow belongs to the kth class.

In the above, α_k is the fraction of the kth class considering only qualified TCP flows. In practice, we are also interested in the fraction of a class over *all* the TCP flows (including both qualified and disqualified flows). We next describe how to obtain this quantity. Let $\tilde{\alpha_k}$ denote the fraction of the kth class over all the TCP flows, $k = 1, \ldots, K$. Suppose the number of qualified flows is n, the total number of flows (both qualified and disqualified) is $\tilde{n}$, and the number of flows that belong to the kth class is $\tilde{n_k}$. Let ϕ_k denote the fraction of qualified flows for the kth class, i.e., it is the ratio of the number of qualified flows that belong to the kth class over the total number of flows that belong to the kth class. We do not know ϕ_k beforehand, and hence obtain an estimate of ϕ_k from the training set for the kth class (i.e., the estimate is the ratio of the number of qualified flows over the total number of flows in the training set for the kth class). Summarizing the above, we have

$$\alpha_k = \frac{\phi_k \tilde{n_k}}{n}.$$

Therefore

$$\tilde{n_k} = \frac{\alpha_k n}{\phi_k}.$$

Hence

$$\tilde{\alpha_k} = \frac{\tilde{n_k}}{\sum\limits_{k=1}^{K} \tilde{n_k}} = \frac{\alpha_k/\phi_k}{\sum\limits_{k=1}^{K} \alpha_k/\phi_k}. \tag{8}$$

Observe from (8) that $\sum_{k=1}^{K} \tilde{\alpha_k} = 1$, and we only need α_k and ϕ_k, $k = 1, \ldots, K$, to obtain $\tilde{\alpha_k}$. Furthermore, $\tilde{\alpha_k} = \alpha_k$ when all the classes have the same fraction of qualified flows, i.e., $\phi_1 = \cdots = \phi_K$. In reality, however, different classes often have different fractions of qualified flows, as we shall see in Tables I and II. Therefore, $\tilde{\alpha_k}$ typically differs from α_k and represents an adjusted value of α_k. Henceforth, we refer to α_k and $\tilde{\alpha_k}$ as *inferred fraction* and *adjusted fraction* of class k, respectively.

Lastly, we can obtain the fraction of each class measured in the number of packets [34] and obtain the adjusted fraction when considering all the packets (in both qualified and disqualified flows) following a similar approach as above. In this paper, we only report the results measured in terms of flows (i.e., α_k and $\tilde{\alpha_k}$) in the interest of space.

VII. Performance Evaluation

In this section, we evaluate the performance of our classification scheme. The traffic traces we use are collected from UMass and UConn campus networks. In both networks, we place a measurement system at the gateway router. The measurement system uses a packet capture card (*DAG* card [1]) to copy all packet headers to a disk along with accurate timestamps. The UMass campus network supports three connection types: 10-Mb/s half-duplex Ethernet (henceforth simply referred to as 10-Mb/s Ethernet), 100-Mb/s Ethernet, and 802.11 WLAN. The UConn campus network supports four connection types: 10-Mb/s, 100-Mb/s, and 1-Gb/s Ethernet and 802.11 WLAN. We use the classification scheme with $K = 3$ and 4 for these two networks, respectively.

Next, we first detail how we construct training sets for these two campus networks, and then evaluate the performance of the classification scheme by constructing *testing sets* and applying the scheme to these testing sets. At the end, we obtain the adjusted fraction for each connection type and evaluate its accuracy.

A. Constructing Training Sets

Training sets are required to obtain the marginal distribution of each connection type. For both campus networks, we construct training sets by extracting TCP flows from a group of traces. The traces for the UMass network are collected between February and April 2005, and the traces for the UConn network are collected more recently, on December 1, 2010. In the UMass network, the training sets for 10-Mb/s and 100-Mb/s Ethernet and WLAN contain TCP flows extracted using IP addresses that belong to two academic departments (they use 10-Mb/s LANs and have no access to any wireless network), the Computer Science Department (100-Mb/s Ethernet addresses that are known to us), and a public 802.11 network (it provides wireless access to campus users within certain public places, e.g., the libraries and the campus eateries), respectively. For the UConn network, the training sets for the various connection types are constructed in a similar manner.

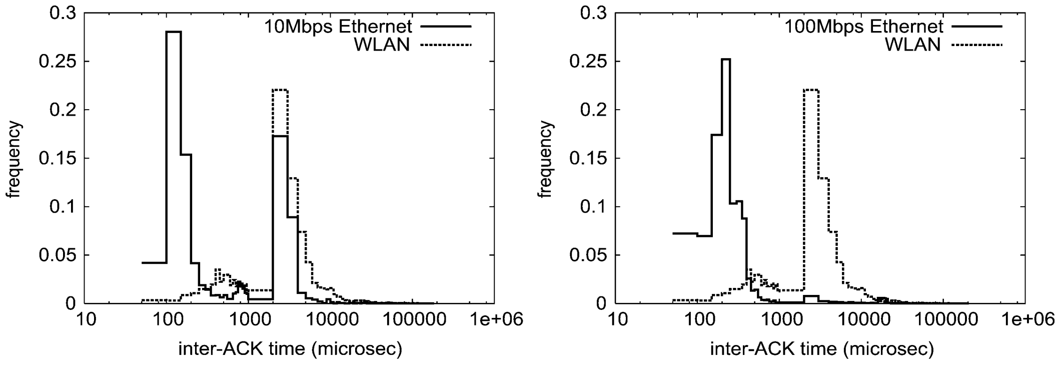
The marginal distribution of a connection type is obtained from its corresponding training set using the approach outlined in Section VI-B. More specifically, in each training set, we set the threshold to identify ACK-pairs, T, to be 250 or 400 μs and set the threshold to identify qualified TCP flows, M, to be 2 or 5. Table I lists the number and percentage (in parentheses) of qualified TCP flows for each connection type in the UMass training sets; Table II lists the results for the UConn training sets. In both tables, as expected, more qualified TCP flows are identified for larger values of T and smaller values of M. We observe that WLAN tends to have a lower percentage of qualified flows than the various Ethernet connections. This is because, as we shall see in Section VIII, the ratio of the number of ACK-pairs

TABLE I
NUMBERS AND PERCENTAGES (IN PARENTHESES) OF QUALIFIED TCP FLOWS IN THE TRAINING SETS FOR UMASS CAMPUS NETWORK

M	WLAN		10Mbps Ethernet		100Mbps Ethernet	
	$T = 250\mu s$	$T = 400\mu s$	$T = 250\mu s$	$T = 400\mu s$	$T = 250\mu s$	$T = 400\mu s$
2	7749 (6.2%)	10873 (8.7%)	2019 (7.5%)	2547 (9.5%)	18527 (8.4%)	21128 (9.5%)
5	2048 (1.6%)	3327 (2.7%)	710 (2.6%)	1002 (3.7%)	9145 (4.1%)	10943 (4.9%)

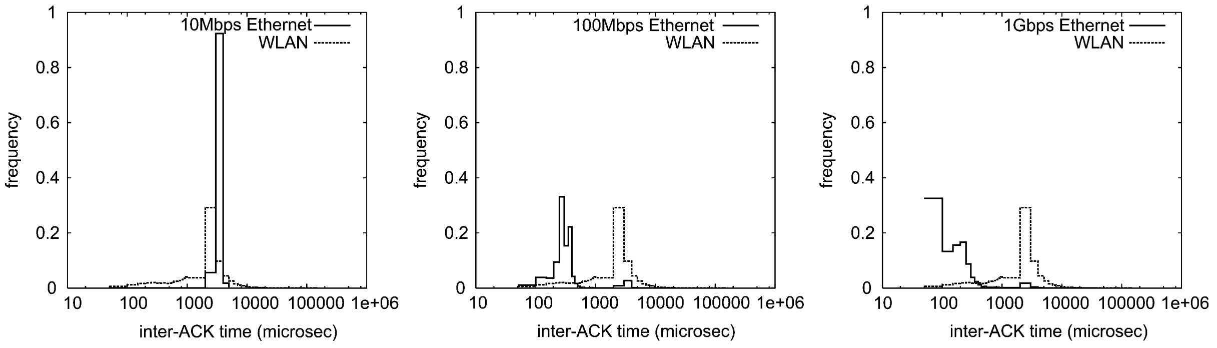
TABLE II
NUMBERS AND PERCENTAGES (IN PARENTHESES) OF QUALIFIED TCP FLOWS IN THE TRAINING SETS FOR UCONN CAMPUS NETWORK

M	WLAN		10Mbps Ethernet		100Mbps Ethernet		1Gbps Ethernet	
	$T = 250\mu s$	$T = 400\mu s$	$T = 250\mu s$	$T = 400\mu s$	$T = 250\mu s$	$T = 400\mu s$	$T = 250\mu s$	$T = 400\mu s$
2	75075 (6.9%)	77186 (7.1%)	272 (6.0%)	285 (6.3%)	3042 (11.2%)	3208 (11.9%)	8153 (8.0%)	8496 (8.3%)
5	36109 (3.3%)	38162 (3.5%)	144 (3.2%)	148 (3.3%)	1574 (5.8%)	1733 (6.4%)	5354 (5.2%)	5649 (5.5%)



(a) (b)

Fig. 3. Marginal distributions of different connection types obtained from the training sets for the UMass network, $T = 400\ \mu$s, $M = 2$. (a) 10-Mb/s half-duplex Ethernet and WLAN. (b) 100-Mb/s Ethernet and WLAN.



(a) (b) (c)

Fig. 4. Marginal distributions of different connection types obtained from the training sets for the UConn network, $T = 400\ \mu$s, $M = 2$. (a) 10-Mb/s Ethernet and WLAN. (b) 100-Mb/s Ethernet and WLAN. (c) 1-Gb/s Ethernet and WLAN.

over the number of packets in a WLAN flow tends to be smaller than that in an Ethernet flow, and hence a short WLAN flow may not have sufficient number of ACK-pairs to be a qualified flow. For each combination of T and M values, we obtain the marginal distribution of each connection type from the qualified TCP flows in its corresponding training set. Figs. 3 and 4 plot the distributions of various Ethernet and WLAN connection types for the UMass and UConn network, respectively, $T = 400\ \mu$s, $M = 2$. In the UMass network, the distribution of 10-Mb/s Ethernet is bimodal [Fig. 3(a)], while the corresponding distribution in the UConn network is not [Fig. 4(a)]. This might be because the 10-Mb/s Ethernet in the UMass network is half-duplex, which can cause ACKs to be queued, leading to small inter-ACK times (the left mode in the distribution). The wireless traffic in the UMass network is predominantly 802.11b (collected in 2005), while the UConn wireless traffic has a richer variety (802.11a/b/g/n) and has smaller inter-ACK times [can be easily observed from their corresponding cumulative distribution functions (cdfs), which are omitted]. In general, consistent with our analytical results in Section IV, the marginal distributions of WLAN and Ethernet differ dramatically while also having significant overlap.

B. Evaluation Results

We conduct both numerical and experimental evaluations. The former uses testing sets that contain observations generated following the marginal distributions of the different classes; the latter uses real network traces. These two types of evaluation are complementary to each other: Numerical evaluation is in a generic setting, while experimental evaluation is in a specific setting that is of interests to us (i.e., classifying TCP flows). We only present the results of experimental evaluation; the results of numerical evaluation are similar. The results are in four scenarios, where the number of classes in a testing set is varied from one to four. The first three scenarios use traces collected from the UMass network; the last one uses traces collected from the UConn network. In each scenario, a testing set contains 100 000 observations. The initial values for α_k when running our iterative inference algorithm are chosen randomly from (0, 1).

1) Single-Class Observations: In this scenario, we construct three testing sets, each containing a single class of TCP flows, i.e., they contain solely 10-Mb/s Ethernet, 100-Mb/s Ethernet, or WLAN flows. For all three testing sets, the inference errors from our classification scheme (i.e., the difference between the inferred fraction and the actual fraction) are zero. Furthermore, for a testing set containing the kth class flows, the belief that a flow belongs to the kth class is 1, and the beliefs that a flow belongs to the other two classes are 0, indicating that our classification scheme leads to a high degree of belief. The high-degree belief results are expected: For the testing set containing solely the kth class flows, since our inferred fraction of this class is 1 and inferred fractions of the other two classes are 0, by (1) in Section V, the beliefs that all the flows belong to the kth class are 1, and the beliefs of being in the other two classes are 0.

2) Two-Class Observations: In this scenario, we construct two types of testing sets: one containing 10-Mb/s Ethernet and WLAN observations, the other containing 100-Mb/s Ethernet and WLAN observations. For both types of testing sets, we vary the fraction of WLAN observations from 0 to 1.

Fig. 5 plots the inference errors versus the fraction of WLAN observations when mixing 10-Mb/s Ethernet and WLAN observations, $T = 400\ \mu$s, $M = 2$. We observe that the inference errors are very small: They are mostly bounded by 0.003 (the maximum absolute error is 0.007), despite the significant overlap of the 10-Mb/s Ethernet and WLAN marginal distributions [see Fig. 3 (a)]. When mixing 100-Mb/s Ethernet and WLAN observations, the inference errors are even smaller (mostly bounded by 0.002, and the maximum absolute error is 0.003, figure omitted).

We next present results on beliefs. Fig. 6(a)–(c) plots the cdf of the beliefs for each class over all the TCP flows when mixing 10-Mb/s Ethernet and WLAN observations, where the fraction of WLAN flows is 0.05, 0.50, and 0.95, respectively (corresponding to low, equal, and high fraction of WLAN flows). Since our inferred fraction of 100-Mb/s Ethernet observations is (close to) zero, as explained in Section VII-B-1, in all three figures, the belief that a flow uses 100-Mb/s Ethernet is (close to) zero. This indicates that our scheme has a high degree of belief that the flows do not use 100-Mb/s Ethernet, which is consistent with the absence of 100-Mb/s Ethernet observations in the testing sets. We also observe from Fig. 6(a)–(c) that as the fraction of WLAN flows increases, more flows have high beliefs of using WLAN. This is because the partial derivative of (1) (in Section V) with respect to α_k is positive, and hence the belief for a flow to be in a class is an increasing function of the fraction of flows in that class. In the extreme case, i.e., the fraction of WLAN flows is zero or one, as shown in Section VII-B-I, for all the flows, the beliefs of using WLAN are zero or one. In Fig. 6(a) and (c), our scheme has a high degree of belief [it concludes that the beliefs for most flows to use WLAN are low in Fig. 6(a), and high in Fig. 6(c)]. In Fig. 6(b), where mixing WLAN and 10-Mb/s Ethernet flows with equal portion, we observe that a significant number of flows have intermediate belief values, indicating that our scheme does not lead to a high degree of belief. This is expected since it is difficult to determine which class a flow belongs to with high confidence in this case (note that our inference results are MLEs, which are already asymptotically optimal). Lastly, we observe similar belief results when mixing WLAN and 100-Mb/s Ethernet flows (figures omitted).

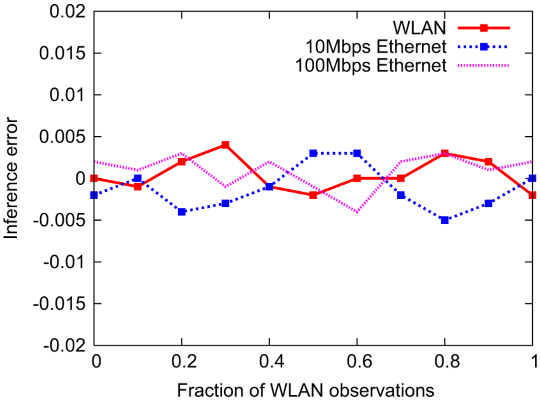
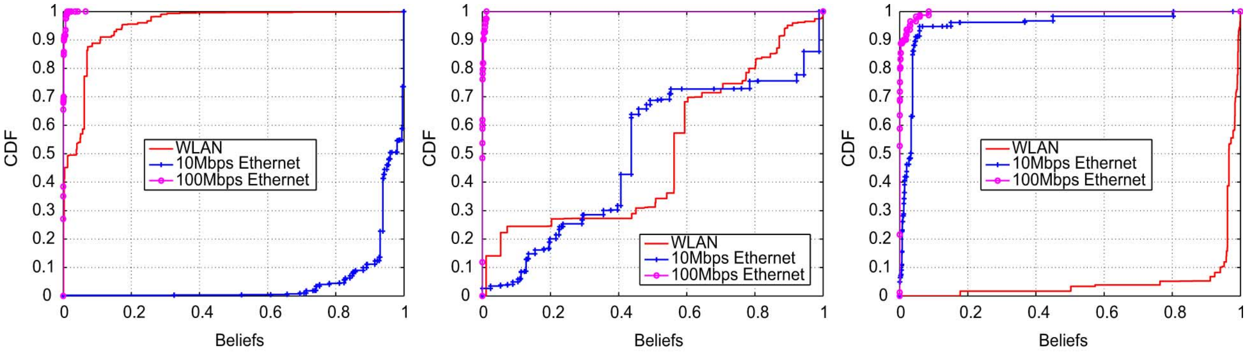


Fig. 5. Inference errors in testing sets containing 10-Mb/s Ethernet and WLAN observations, $T = 400\ \mu$s, $M = 2$.

3) Three-Class Observations: In this scenario, we construct testing sets containing 10-Mb/s, 100-Mb/s Ethernet, and WLAN observations. We again vary the fraction of WLAN flows from 0 to 1, and the fractions of 10-Mb/s and 100-Mb/s Ethernet observations are the same.

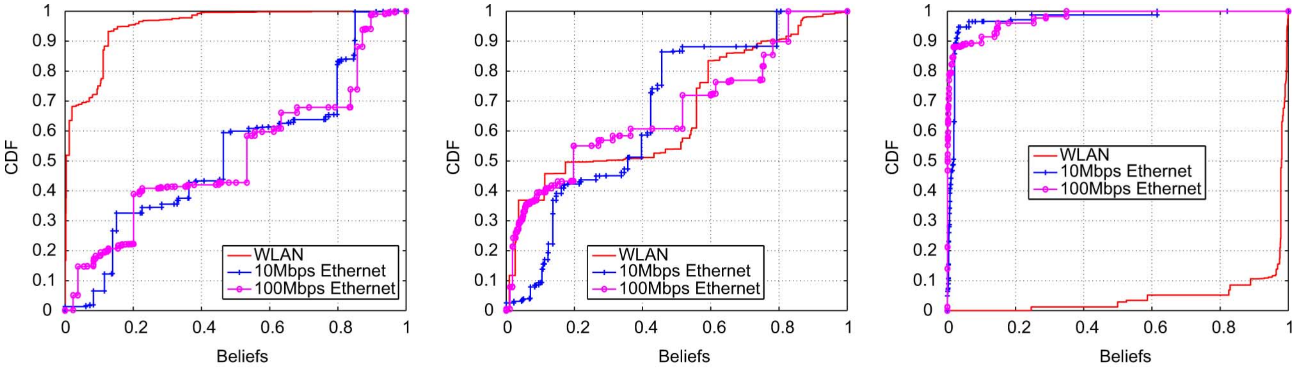
We again observe that our inference results are highly accurate: Most of the inference errors are bounded by 0.003 (the maximum absolute error is 0.005). Fig. 7(a)–(c) plots the cdfs of the beliefs when the fractions of WLAN flows are 0.05, 0.34, and 0.95, respectively. We observe similar results as those when mixing two types of flows. More specifically, most flows have low (high) beliefs of using WLAN when the fraction of WLAN flows is low (high). When the fraction of WLAN flows is 0.34, that is, the testing set has (almost) equal fractions of 10-Mb/s, 100-Mb/s Ethernet, and WLAN flows, for each connection type, a significant number of the flows have intermediate belief values.

4) Four-Class Observations: In this scenario, we construct testing sets containing 10-Mb/s, 100-Mb/s, and 1-Gb/s Ethernet, and WLAN observations. We again vary the fraction of WLAN flows from 0 to 1, and the fractions of the various types of Ethernet observations are the same. We again observe very small inference errors (maximum inference error 0.005). The cdfs of



(a) (b) (c)

Fig. 6. CDFs of the beliefs in testing sets containing 10-Mb/s Ethernet and WLAN observations, $T = 400\ \mu\text{s}$, $M = 2$. Fraction of WLAN flows: (a) 0.05, (b) 0.50, and (c) 0.95.



(a) (b) (c)

Fig. 7. CDFs of the beliefs in testing sets containing 10-Mb/s Ethernet, 100-Mb/s Ethernet, and WLAN observations, $T = 400\ \mu\text{s}$, $M = 2$. Fraction of WLAN flows: (a) 0.05, (b) 0.34, and (c) 0.95.

the beliefs show similar trends as those in two-class and three-class scenarios (figures omitted).

C. Adjusted Fractions

The testing sets above only contain qualified TCP flows. We next construct testing sets with both qualified and disqualified TCP flows and obtain the adjusted fraction of each class over all the flows. In particular, we construct two testing sets: One is the union of the three training sets for the UMass network, and the other is the union of the four training sets for the UConn network. For both testing sets, we first obtain a set of qualified flows and apply our classification algorithm to infer the fraction of each class among the qualified flows. We then use (8) to obtain the adjusted fraction for each class (we use the percentages of qualified flows listed in Tables I and II as $\phi_k,\ k = 1, \ldots, K$ for the two testing sets, respectively). The errors of the adjusted fraction are very small: Most errors are within 0.001, and the maximum error is within 0.003. For these two testing sets, the percentages of qualified flows (i.e., $\phi_k,\ k = 1, \ldots, K$) are accurate. In practice, since the estimate of ϕ_k may not be perfect, the errors in the adjusted fractions can be larger.

VIII. Inference Results in Campus Networks

In this section, we apply our classification scheme to traces collected from two campus networks. In particular, we use two traces collected from the UMass network: one collected between 11 AM and 12 PM on April 4, 2005, containing 3 309 480 TCP flows, and the other collected between 10 AM and 12 PM on May 10, 2005, containing 6 250 306 TCP flows. We also use two traces from the UConn network: one collected on December 13, 2010 between 1:30–2:30 PM, containing 3 444 009 TCP flows, and the other collected on December 16, 2010 from 2:30–3:30 PM, containing 2 711 419 TCP flows. For the two traces from the UMass network, we use the classification scheme with $K = 3$ to classify the TCP flows into three classes: 10-Mb/s and 100-Mb/s Ethernet, and WLAN flows. For the two traces from the UConn network, we use the classification scheme with $K = 4$ to classify the TCP flows into four classes: 10-Mb/s, 100-Mb/s, and 1-Gb/s Ethernet, and WLAN flows. The training sets for these two networks are as described in Section VII-A. In each trace, we set $T = 250\ \mu\text{s}$ or $400\ \mu\text{s}$ and $M = 2$ or 5 when identifying qualified TCP flows. Lastly, we use the percentages of qualified TCP flows listed in Tables I and II to obtain the adjusted fraction of each connection type in these two networks, respectively.

A. UMass Network

In the UMass network, we know the IP address block that is reserved for the residential network (i.e., student dorms), which allows us to identify all TCP flows to that network. We next obtain the inference results for both the residential network and the entire campus network. The reason for the former is twofold: 1) a large fraction of the campus traffic is from the residential network (in the two traces, 64% and 60% of the flows are from

TABLE III
INFERENCE RESULTS OF THE RESIDENTIAL AND THE ENTIRE CAMPUS NETWORK, UMASS (MAY 10, 2005, 10 AM–12 PM)

Connection type	M	Residential				Entire Campus			
		$T = 250\mu s$		$T = 400\mu s$		$T = 250\mu s$		$T = 400\mu s$	
		inferred	adjusted	inferred	adjusted	inferred	adjusted	inferred	adjusted
WLAN	2	0.02	0.02	0.02	0.02	0.04	0.05	0.04	0.04
	5	0.02	0.03	0.02	0.03	0.04	0.07	0.04	0.06
10Mbps Ethernet	2	0.94	0.94	0.95	0.95	0.78	0.79	0.79	0.79
	5	0.93	0.94	0.95	0.95	0.77	0.80	0.80	0.81
100Mbps Ethernet	2	0.04	0.04	0.03	0.03	0.18	0.16	0.17	0.17
	5	0.05	0.03	0.03	0.02	0.20	0.13	0.17	0.13

the residential network, respectively); and 2) we are interested in finding out whether WLAN traffic is present in the residential network—since it has no official provision of WLAN, the presence of WLAN traffic implies provision through private routers or access points.

Table III presents the inferred fraction and the adjusted fraction of each connection type for the trace collected on May 10, 2005 (results for the other trace are consistent). In the table, the results for different combinations of T and M values are similar, indicating that our scheme is not sensitive to the choice of the parameters. Furthermore, in both the residential network and the entire network, for WLAN, the adjusted fraction tends to be larger than the inferred fraction since WLAN tends to have a lower percentage of qualified flows compared to the various Ethernet connections (see Tables I and II).

In the residential network, the inferred fractions of WLAN, 10-Mb/s Ethernet, and 100-Mb/s Ethernet flows are 0.02, 0.93–0.95, and 0.03–0.05, respectively; the adjusted fractions are 0.02–0.03, 0.94–0.95, and 0.02–0.04, respectively. The fraction of 10-Mb/s Ethernet flows is close to one, which is consistent with our knowledge that the official connection type of the residential network is 10-Mb/s Ethernet. However, the inferred fraction of WLAN flows is nonzero, and furthermore, our inference shows that some flows have very high beliefs of using WLAN (exceeding 0.99). We therefore infer that wireless traffic is present in the residential network (through private wireless routers and access points since there is no official wireless coverage). The inferred fraction of 100-Mb/s Ethernet flows is nonzero either. This estimation error might be due to traffic limiters in the residential network, which queue packets once the traffic rate is above 10 Mb/s (and hence can queue ACKs together and lead to small inter-ACK times as in the 100-Mb/s Ethernet case). Traffic limiters are not used in the two academic departments whose TCP flows are used to obtain the training set as described in Section VII-A. Indeed, the median inter-ACK time distribution of the residential network has a higher density for small inter-ACK times (due to ACK queuing by the traffic limiters) compared to the marginal distribution from the training set.[6]

In the entire campus, the inferred fraction of WLAN flows is 0.04, and the adjusted fraction is 0.04–0.07; the inferred fraction of 10-Mb/s Ethernet flows is still large, 0.77–0.80, with the adjusted fraction of 0.79–0.81, which is consistent with our knowledge that a large fraction of UMass traffic is from the residential network (which uses 10-Mb/s Ethernet). Most flows have low beliefs of using WLAN (97% of the flows have beliefs of using WLAN below 0.2) and a significant number of flows have high beliefs of using 10-Mb/s Ethernet (64% of the flows have beliefs of using 10-Mb/s Ethernet above 0.8). As explained in Section VII, this is because the inferred fraction of WLAN flows is small and the inferred fraction of 10-Mb/s Ethernet flows is large.

Let us refer to a flow that has the belief of using WLAN larger than 0.8 as a *WLAN-likely flow*, and a flow that has the belief of using Ethernet (10 or 100 Mb/s) larger than 0.8 as an *Ethernet-likely flow*. We next investigate the differences between WLAN-likely and Ethernet-likely flows to shed more insights into our inference results. In particular, the property we investigate is *ACK-pair ratio*, the number of ACK-pairs in a flow divided by the total number of packets in the flow. Intuitively, a WLAN flow has a smaller ACK-pair ratio than an Ethernet flow. This is because, as shown in Section IV, the inter-ACK times in a WLAN flow tend to be larger than those in an Ethernet flow, which leads to dispersion of data packets (due to TCP's self-clocking), and hence less ACK-pairs. We confirm the above intuition using results from our training sets. More specifically, we sort the flows in the training set in decreasing order of flow length (in packets) and obtain the average ACK-pair ratios for the top xth percentile of the Ethernet (10 or 100 Mb/s) and WLAN flows, respectively. Fig. 8(a) plots the results for x from 10 to 100. The 95th confidence intervals are very tight and hence omitted. We observe that WLAN flows indeed have smaller ACK-pair ratios than Ethernet flows. Fig. 8(b) plots the average ACK-pair ratios for the top xth percentile of the WLAN-likely and Ethernet-likely flows (confidence intervals are again tight and omitted). We observe that WLAN-likely flows have smaller ACK-pair ratios than Ethernet-likely flows for all values of x. This is consistent with the results from the training data, indicating that our classification results accurately reflect the relationship between WLAN and Ethernet ACK-pair ratios.

B. UConn Network

Table IV lists the inference results for the trace collected on December 13, 2010 from the UConn network (the results for the other trace are consistent). The inferred fractions of

[6] While the queuing by the traffic limiters may also separate some ACKs apart, leading to large inter-ACK times, we do not believe the nonzero percentage of WLAN traffic is an estimation error caused by the traffic limiters. This is because the queuing should only separate a small fraction of the ACK-pairs apart (i.e., when there is an excessive delay from draining one ACK to draining the other ACK of an ACK-pair from the buffer), and hence should not affect our observations (an observation is the *median* inter-ACK time of a qualified TCP flow; see Section VI). Furthermore, it is unlikely that traffic limiters can cause random delays as those caused by WLAN protocols.

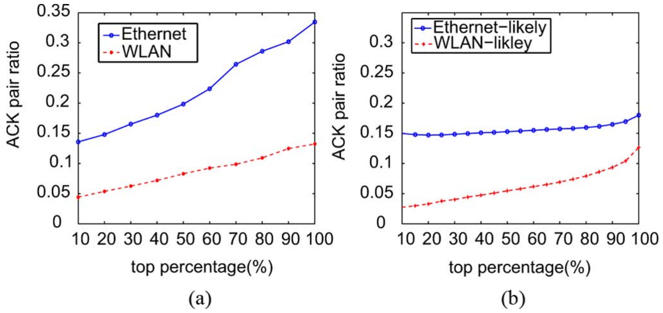


Fig. 8. ACK-pair ratios (a) from the training sets for the UMass network and (b) from Ethernet-likely and WLAN-likely flows for the trace collected on May 10, 2005 from the UMass network, $T = 400\ \mu\text{s}$, $M = 2$.

TABLE IV
INFERENCE RESULTS OF THE UCONN NETWORK (DECEMBER 13, 2010, 1:30–2:30 PM)

Connection type	M	$T = 250\mu s$		$T = 400\mu s$	
		inferred	adjusted	inferred	adjusted
WLAN	2	0.51	0.55	0.51	0.55
	5	0.46	0.52	0.46	0.54
10Mbps Ethernet	2	0.17	0.21	0.16	0.20
	5	0.18	0.22	0.17	0.21
100Mbps Ethernet	2	0.19	0.12	0.19	0.12
	5	0.21	0.14	0.21	0.13
1Gbps Ethernet	2	0.13	0.12	0.14	0.13
	5	0.17	0.12	0.16	0.12

WLAN, 10-Mb/s, 100-Mb/s, and 1-Gb/s Ethernet flows are 0.46–0.51, 0.16–0.18, 0.19–0.21, and 0.13–0.17, respectively; the adjusted fractions are 0.52–0.55, 0.20–0.22, 0.12–0.14, and 0.12–0.13, respectively. Observe that WLAN is the dominant connection type: Around half of the traffic traverses a wireless link. The fraction of WLAN traffic is much more than that in the UMass network. In addition, compared to the belief results for the UMass network, the belief of using WLAN has increased dramatically: Around 40% of the flows have beliefs of using WLAN above 0.8, which is consistent with the much larger fraction of traffic that traverses a WLAN in the UConn network.

IX. DISCUSSION

A natural question is whether our classification scheme is applicable to future networks where the bit rates of Ethernet and WLAN increase, and WLAN might provide similar or even higher bit rate than Ethernet. Since our scheme relies on the marginal distributions of the various connection types, it is applicable as long as these marginal distributions are different. Our experimental results support the above statement. Specifically, we construct training sets for 802.11b, 802.11g, and 802.11n traffic using controlled experiments in the UConn campus network (as described in Section IV) to obtain the marginal distribution for each type of WLAN connection. We then construct testing sets by mixing one type of WLAN traffic (802.11b, 802.11g, or 802.11n) and one type of Ethernet traffic (10-Mb/s, 100-Mb/s, or 1-Gb/s Ethernet), where the fraction of WLAN traffic is varied from 0 to 1 (as in Section VII-B-II). Applying our scheme to these testing sets obtains accurate inference results: The inference error of the fraction of WLAN traffic is below 0.01 for all the cases except when mixing 802.11n and 1-Gb/s Ethernet traffic where the error is within 0.02. These results demonstrate that our scheme is applicable to scenarios where WLAN can provide comparable or even higher bit rate than Ethernet (e.g., 802.11g or 802.11n versus 10-Mb/s Ethernet) as long as their marginal distributions are different.

Another question is whether our scheme is applicable to networks that use traffic shapers. In the two campus networks that we study, the traffic shapers in the UMass network are simply traffic limiters, which do not affect the classification of WLAN traffic (although they led to an overestimate of 100-Mb/s Ethernet traffic) as explained in Section VIII-A; the traffic shapers in the UConn network are at the edge of the network, mainly blocking undesirable traffic, and hence do not affect our classification results. In other networks, the traffic shapers can work in different ways. Our scheme is applicable as long as the traffic shapers do not affect the observations (i.e., median inter-ACK times of the qualified TCP flows).

X. CONCLUSION

In this paper, we have proposed a classification scheme to differentiate Ethernet and WLAN TCP flows based on measurements collected passively at the edge of a network. This scheme computes the fraction of wireless TCP flows and the belief that a TCP flow traverses a WLAN inside the network. The core of the scheme is an iterative Bayesian inference algorithm that we developed to obtain the MLE of these quantities. Numerical and experimental evaluations demonstrated that our classification scheme obtains accurate results. We applied the scheme to various traces collected from two campus networks: One set of traces was collected from the UMass network in 2005, and the other set was collected from the UConn network in 2010. Inference in the UMass network indicates that 4%–7% of all incoming TCP flows traverse an 802.11 wireless link within the campus. It also detects wireless usage (through the use of private routers and access points) in areas not covered by the official wireless infrastructure. Inference in the UConn network shows a much more prevalent amount (52%–55%) of wireless traffic.

APPENDIX I
PROOF OF THEOREMS 1–3

Theorem 1 is for 100-Mb/s Ethernet [see Fig. 2(a)]. Theorems 2 and 3 are for 11-Mb/s 802.11b and 54-Mb/s 802.11g WLAN, respectively [see Fig. 2(b)]. The proof of Theorem 1 utilizes the following two lemmas.

Lemma 1: Let $x = P(\Delta_{\text{A}} > 600\ \mu\text{s})$. Then, $P(\xi_{.5}^{(m)}(\Delta_{\text{A}}) > 600\ \mu\text{s})$ is an increasing function of x.

Proof: By the definition of median, we have

$$P(\xi_{.5}^{(m)}(\Delta_{\text{A}}) > 600\ \mu\text{s}) = \sum_{\lceil m/2 \rceil}^{m} \binom{m}{i} x^i (1-x)^{m-i}.$$

We show that $P(\xi_{.5}^{(m)}(\Delta_{\text{A}}) > 600\ \mu\text{s})$ is an increasing function of x by showing that its derivative is positive (details omitted). ■

Lemma 2: In the Ethernet setting, when $0 < \rho_{\mathrm{D}}, \rho_{\mathrm{A}} \leq 1$, $P(\Delta_{\mathrm{A}} > 600\ \mu\mathrm{s}) < 0.18$.

Proof: The proof is found in [35, Appendix A]. ■

We now prove Theorem 1 using Lemmas 1 and 2.

Proof: Let $x = P(\Delta_{\mathrm{A}} > 600\ \mu\mathrm{s})$. Then

$$P(\xi_{.5}^{(m)}(\Delta_{\mathrm{A}}) > 600\ \mu\mathrm{s}) = \sum_{i=\lceil m/2 \rceil}^{m} \binom{m}{i} x^i (1-x)^{m-i}.$$

When $x = 0.18$, by direct calculation, we have $P(\xi_{.5}^{(5)}(\Delta_{\mathrm{A}}) > 600\ \mu\mathrm{s}) < 0.04$. Since $P(\xi_{.5}^{(m)}(\Delta_{\mathrm{A}}) > 600\ \mu\mathrm{s})$ is an increasing function of x (Lemma 1) and $x < 0.18$ (Lemma 2), we have the desired results. ■

The proof of Theorem 2 is similar to that of Theorem 1. It utilizes Lemma 1 and the following lemma (proof found in [35, Appendix C]).

Lemma 3: In the 802.11b WLAN setting, under idealized conditions (i.e., the channel between the access point and the receiver is perfect and is only used by the access point and the receiver), $P(\Delta_{\mathrm{A}} > 600\ \mu\mathrm{s}) > 0.96$.

The proof of Theorem 3 is similar to that of Theorem 1. It utilizes Lemma 1 and the following lemma (proof found in [35, Appendix D]).

Lemma 4: In the 802.11g WLAN setting, under idealized conditions (i.e., the channel between the access point and the receiver is perfect and is only used by the access point and the receiver), $P(\Delta_{\mathrm{A}} > 600\ \mu\mathrm{s}) > 0.45$.

Appendix II
Proof of Theorem 4

Proof: As shown in Section V, the MLE of α_k is obtained by solving (7). We prove that the MLE of α_k is unique by looking at the second-order conditions and considering two cases: $K = 2$ and $K > 2$.

When $K = 2$, the MLE of is α_k obtained from

$$f'(\alpha_1) = \sum_{i=1}^{N} \frac{p_{1i} - p_{2i}}{g_i} = 0. \tag{9}$$

The second derivative of $f(\alpha_1)$ is

$$f''(\alpha_1) = -\sum_{i=1}^{N} \frac{(p_{1i} - p_{2i})^2}{g_i^2} \leq 0. \tag{10}$$

We assume that $\sum_{i=1}^{N}(p_{1i} - p_{2i})^2 > 0$ (otherwise, the two classes are not differentiable from the observations). Under this assumption, $f''(\alpha_1) < 0$. That is, $f'(\alpha_1)$ is a strictly decreasing function of α_1. If $f'(0) > 0$ and $f'(1) < 0$, then $f'(\alpha_1)$ has a unique solution in (0, 1). This unique solution is the MLE of α_1. Otherwise, we have $f'(\alpha_1) > 0$ or $f'(\alpha_1) < 0$ for $\alpha_1 \in (0, 1)$. This implies that the likelihood function, $f(\alpha_1)$, is an increasing or decreasing function of α_1. Hence, the MLE is achieved at either 1 or 0. Combining all the above cases, the MLE of α_1 is unique, and hence the MLE of α_2 is unique. Since β_{ki} is a function of α_k, the MLE of β_{ki} is also unique.

When $K > 2$, the second-order conditions are expressed in the form of Hessian matrix

$$H = \begin{pmatrix} f_{1,1} & f_{1,2} & \cdots & f_{1,K-1} \\ \vdots & \vdots & \vdots & \vdots \\ f_{K-1,1} & f_{K-1,2} & \cdots & f_{K-1,K-1} \end{pmatrix}$$

where

$$\begin{aligned} f_{k,j} &= \frac{\partial^2 f(\alpha_1, \ldots, \alpha_{K-1})}{\partial \alpha_k \partial \alpha_j} \\ &= -\sum_{i=1}^{N} \frac{(p_{ki} - p_{Ki})(p_{ji} - p_{Ki})}{g_i^2}. \end{aligned} \tag{11}$$

We next show that the Hessian matrix is negative semidefinite. Consider a vector $\mathbf{h} = (h_1, \ldots, h_{K-1})$. By direct calculation, we have

$$\mathbf{h}H\mathbf{h}^t = -\sum_{i=1}^{N} \frac{\left(\sum_{k=1}^{K-1} h_k (p_{ki} - p_{Ki})\right)^2}{g_i^2} \geq 0. \tag{12}$$

Therefore, the likelihood function has a maximum. Furthermore, $\mathbf{h}H\mathbf{h}^t$ is zero iff for each $i = 1, \ldots, N$, p_{ki}'s are the same for all $k = 1, \ldots, K$. We assume this is not the case since it renders the multiple classes not differentiable from the observations. Therefore, $\mathbf{h}H\mathbf{h}^t > 0$, and hence the likelihood function has a unique maximum.

We next consider the scenario where (7) does not have a solution for any $\alpha_k \in [0, 1]$. That is, the derivative $\frac{\partial f(\alpha_1, \ldots, \alpha_{K-1})}{\partial \alpha_k}$ is larger or smaller than zero for all $\alpha_k \in [0, 1]$. When the derivative is larger than zero, the likelihood function $f(\alpha_1, \ldots, \alpha_{K-1})$ is an increasing function of α_k and achieves the maximum at $\alpha_k = 1$. When the derivative is smaller than zero, $f(\alpha_1, \ldots, \alpha_{K-1})$, is a decreasing function of α_k and achieves the maximum at $\alpha_k = 0$. In either case, the likelihood function has a unique maximum.

Summarizing the cases of $K = 2$ and $K \geq 2$, we have proven the theorem. ■

Appendix III
Proof of Theorem 5

Proof: Let $\bar{\alpha}_k = \lim_{m \to \infty} \alpha_k^{(m)}$ and $\bar{\beta}_{ki} = \lim_{m \to \infty} \beta_{ki}^{(m)}$. Then

$$\bar{\beta}_{ki} = \frac{\bar{\alpha}_k p_{ki}}{\sum_{j=1}^{K} \bar{\alpha}_j p_{ji}}, \qquad \text{for } k = 1, \ldots, K, i = 1, \ldots, N \tag{13}$$

$$\bar{\alpha}_k = \frac{1}{N} \sum_{i=1}^{N} \bar{\beta}_{ki}, \qquad \text{for } k = 1, \ldots, K. \tag{14}$$

Combining (13) and (14), we have

$$N\bar{\alpha}_k = \sum_{i=1}^{N} \bar{\beta}_{ki} = \sum_{i=1}^{N} \frac{\bar{\alpha}_k p_{ki}}{\sum_{j=1}^{K} \bar{\alpha}_j p_{ji}}. \tag{15}$$

Consider $\bar{\alpha}_k \in (0,1)$. Dividing $\bar{\alpha}_k$ on both sides of (15), we have

$$\sum_{i=1}^{N}\left(\frac{p_{ki}}{\sum\limits_{j=1}^{K}\bar{\alpha}_j p_{ji}} - 1\right) = 0. \qquad (16)$$

Substituting $\bar{\alpha}_K = 1 - \sum_{j=1}^{K-1}\bar{\alpha}_j$ into (16), we have

$$\sum_{i=1}^{N}\left(\frac{p_{ki}-g_i}{g_i}\right) = 0, \qquad \text{for } k = 1,\ldots,K-1 \qquad (17)$$

where $g_i = \sum_{j=1}^{K-1}\bar{\alpha}_j p_{ji} + (1 - \sum_{k=1}^{K-1}\bar{\alpha}_k)p_{Ki}$. By direct calculation, (17) is equivalent to

$$\sum_{i=1}^{N}\frac{(\bar{\alpha}_k - 1)(p_{Ki} - p_{ki}) + \sum\limits_{j=1, j\neq k}^{K}\bar{\alpha}_j(p_{Ki} - p_{ji})}{g_i} = 0, \qquad \text{for } k = 1,\ldots,K-1. \qquad (18)$$

This is equivalent to (7), i.e., the condition satisfied by the MLE $\hat{\alpha}_k$. This implies that, if the inference algorithm converges to a solution of $\hat{\alpha}_k$ in (0, 1), then the solution is the unique MLE.

In the above, we consider $\bar{\alpha}_k \in (0,1)$. We next consider the cases where $\bar{\alpha}_k = 1$ or 0. From Algorithm 1, we have

$$\alpha_k^{(m+1)} = \frac{1}{N}\sum_{i=1}^{N}\frac{\alpha_k^{(m)}p_{ki}}{\sum\limits_{j=1}^{K}\alpha_j^{(m)}p_{ji}}. \qquad (19)$$

When $\bar{\alpha}_k = 1$, we have $p_{ki} > 0$ and $p_{ji} = 0$ for $j = 1,\ldots,K$, $j \neq k$. In this case, our algorithm converges to obtain $\bar{\alpha}_k = 1$ after the first iteration when $0 < \alpha_k^{(0)} < 1$. When $\bar{\alpha}_k = 0$, we have $p_{ki} = 0$ and there exists j such that $p_{ji} > 0$ for $j = 1,\ldots,K, j \neq k$. In this case, our algorithm converges to obtain $\bar{\alpha}_k = 0$ after the first iteration when $0 < \alpha_k^{(0)} < 1$.

In the above, we have proven that $\bar{\alpha}_k$ converges to the unique MLE of α_k. By the invariance property of maximum likelihood estimators [11], $\bar{\beta}_{ki}$ converges to the MLE of β_{ki}, $k = 1,\ldots,K$, $i = 1,\ldots,N$. ■

Acknowledgment

The authors wish to thank R. Tuthill (UMass), R. Kocsondy, J. Pufahl, and S. Maresca (UConn) for their help with the monitoring equipment and understanding the network architecture of the two campus networks, and X. Chen for collecting traces in the UConn network. They also thank the anonymous reviewers for their insightful comments, and M. Allman for handling the paper. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding agencies.

References

[1] "Endace," Endace, Auckland, New Zealand [Online]. Available: http://www.endace.com

[2] "Microsoft Windows 2000 TCP/IP implementation details," [Online]. Available: http://www.microsoft.com/technet/itsolutions/network/deploy/depovg/tcpip2k.mspx

[3] A. Adya, V. Bahl, R. Chandra, and L. Qiu, "Architecture and techniques for diagnosing faults in IEEE 802.11 infrastructure networks," in *Proc. ACM MobiCom*, Sep. 2004, pp. 30–44.

[4] P. Bahl, R. Chandra, J. Padhye, L. Ravindranath, M. Singh, A. Wolman, and B. Zill, "Enhancing the security of corporate Wi-Fi networks using DAIR," in *Proc. ACM MobiSys*, Jun. 2006, pp. 1–14.

[5] V. Baiamonte, K. Papagiannaki, and G. Iannaccone, "Detecting 802.11 wireless hosts from remote passive observations," in *Proc. IFIP/TC6 Netw.*, Atlanta, GA, May 2007, pp. 356–367.

[6] A. Balachandran, G. Voelker, P. Bahl, and V. Rangan, "Characterizing user behavior and network performance in a public wireless LAN," in *Proc. ACM SIGMETRICS*, Marina Del Rey, CA, Jun. 2002, pp. 195–205.

[7] M. Balazinska and P. Castro, "Characterizing mobility and network usage in a corporate wireless local-area network," in *Proc. ACM MobiSys*, May 2003, pp. 303–316.

[8] R. Beyah, S. Kangude, G. Yu, B. Strickland, and J. Copeland, "Rogue access point detection using temporal traffic characteristics," in *Proc. IEEE GLOBECOM*, Dec. 2004, vol. 4, pp. 2271–2275.

[9] G. Birkhoff and S. L. Mac, *Insolvability of Quintic Equations*, 5th ed. New York: Macmillan, 1996, ch. 15.8, pp. 418–421.

[10] R. Carter and M. Crovella, "Measuring bottleneck link speed in packet-switched networks," *Perform. Eval.*, vol. 27–28, no. 4, pp. 297–318, 1996.

[11] G. Casella and R. L. Berger, *Statistical Inference*. Belmont, CA: Duxbury, 2002, ch. 7, p. 320.

[12] L. Cheng and I. Marsic, "Fuzzy reasoning for wireless awareness," *Int. J.Wireless Inf. Netw.*, vol. 8, no. 1, pp. 15–26, 2001.

[13] Y.-C. Cheng, J. Bellardo, P. Benko, A. C. Snoeren, G. M. Voelker, and S. Savage, "Jigsaw: Solving the puzzle of enterprise 802.11 analysis," in *Proc. ACM SIGCOMM*, Pisa, Italy, Sep. 2006, pp. 39–50.

[14] A. Dempster, N. Laird, and D. Rubin, "Maximum-likelihood from incomplete data via the EM algorithm," *J. Roy. Stat. Soc. Ser. B, Meth.*, vol. 39, no. 1, pp. 39:1–38, 1977.

[15] C. Dovrolis, P. Ramanathan, and D. Moore, "What do packet dispersion techniques measure?," in *Proc. IEEE INFOCOM*, Apr. 2001, vol. 2, pp. 905–914.

[16] A. B. Downey, "Using pathchar to estimate Internet link characteristics," *Proc. ACM SIGCOMM*, pp. 241–250, Aug. 1999.

[17] T. Henderson, D. Kotz, and I. Abyzov, "The changing usage of a mature campus-wide wireless network," in *Proc. ACM MobiCom*, Sep. 2004, pp. 187–201.

[18] V. Jacobson, "Pathchar—A tool to infer characteristics of Internet paths," Apr. 1997 [Online]. Available: ftp://ftp.ee.lbl.gov/pathchar

[19] S. Jaiswal, G. Iannaccone, C. Diot, J. Kurose, and D. Towsley, "Measurement and classification of out-of-sequence packets in a tier-1 IP backbone," in *Proc. IEEE INFOCOM*, Mar. 2003, vol. 2, pp. 1199–1209.

[20] S. Jaiswal, G. Iannaccone, C. Diot, and J. K. Towsley, "Inferring TCP connection characteristics through passive measurements," in *Proc. IEEE INFOCOM*, Mar. 2004, vol. 3, pp. 1582–1592.

[21] D. Katabi, I. Bazzi, and X. Yang, "A passive approach for detecting shared bottlenecks," in *Proc. IEEE Int. Conf. Comput. Commun. Netw.*, 2001, pp. 174–181.

[22] S. Katti, D. Katabi, C. Blake, E. Kohler, and J. Strauss, "MultiQ: Automated detection of multiple bottleneck capacities along a path," in *Proc. ACM SIGCOMM IMC*, Oct. 2004, pp. 245–250.

[23] R. Mahajan, M. Rodrig, D. Wetherall, and J. Zahorjan, "Analyzing the MAC-level behavior of wireless networks in the wild," in *Proc. ACM SIGCOMM*, Pisa, Italy, Sep. 2006, pp. 75–86.

[24] C. Mano, A. Blaich, Q. Liao, Y. Jiang, D. Salyers, D. Cieslak, and A. Striegel, "RIPPS: Rogue identifying packet payload slicer detecting unauthorized wireless hosts through network traffic conditioning," *Trans. Inf. Syst. Security*, vol. 11, no. 2, Mar. 2008, Article no. 2.

[25] K. Papagiannaki, S. Moon, C. Fraleigh, P. Thiran, and C. Diot, "Measurement and analysis of single-hop delay on an IP backbone network," *IEEE J. Sel. Areas Commun.*, vol. 21, no. 6, pp. 908–921, Aug. 2003.

[26] A. Persson, C. A. C. Marcondes, L.-J. Chen, M. Y. Sanadidi, and M. Gerla, "TCP probe: A TCP with built-in path capacity estimation," presented at the 8th IEEE Global Internet Symp., Mar. 2005.

[27] D. Rubenstein, J. Kurose, and D. Towsley, "Detecting shared congestion of flows via end-to-end measurement," in *Proc. ACM SIGMETRICS*, Jun. 2000, pp. 145–155.

[28] P. Sarolahti and A. Kuznetsov, "Congestion control in Linux TCP," in *Proc. USENIX*, Jun. 2002, pp. 49–62.

[29] D. Schwab and R. Bunt, "Characterising the use of a campus wireless network," in *Proc. IEEE INFOCOM*, Mar. 2004, vol. 2, pp. 862–870.

[30] A. Sheth, C. Doerr, D. Grunwald, R. Han, and D. C. Sicker, "MOJO: A distributed physical layer anomaly detection system for 802.11 WLANs," *Proc. ACM MobiSys*, pp. 191–204, Jun. 2006.

[31] V. Shrivastava, S. Rayanchu, J. Yoon, and S. Banerjee, "802.11n under the microscope," in *Proc. ACM SIGCOMM IMC*, Oct. 2008, pp. 105–110.

[32] D. Tang and M. Baker, "Analysis of a local-area wireless network," in *Proc. ACM MobiCom*, Jun. 2000, pp. 1–10.

[33] W. Wei, S. Jaiswal, J. Kurose, and D. Towsley, "Identifying 802.11 traffic from passive measurements using iterative Bayesian inference," Department of Computer Science, University of Massachusetts, Amherst, MA, 2005.

[34] W. Wei, S. Jaiswal, J. Kurose, and D. Towsley, "Identifying 802.11 traffic from passive measurements using iterative Bayesian inference," in *Proc. IEEE INFOCOM*, Apr. 2006, pp. 1–12.

[35] W. Wei, K. Suh, B. Wang, Y. Gu, J. Kurose, D. Towsley, and S. Jaiswal, "Passive online detection of 802.11 traffic using sequential hypothesis testing with TCP ACK-pairs," *IEEE Trans. Mobile Comput.*, vol. 8, no. 3, pp. 398–412, Mar. 2009.

[36] W. Wei, B. Wang, C. Zhang, J. Kurose, and D. Towsley, "Classification of access network types: Ethernet, wireless LAN, ADSL, cable modem or dialup?," *Comput. Netw.*, vol. 52, no. 17, pp. 3205–3217, Dec. 2008.

[37] J. Yeo, M. Youssef, and A. Agrawala, "A framework for wireless LAN monitoring and its applications," in *Proc. ACM WiSe*, Oct. 2004, pp. 70–79.

[38] J. Yeo, M. Youssef, T. Henderson, and A. Agrawala, "An accurate technique for measuring the wireless side of wireless networks," in *Proc. USENIX/ACM WiTMeMo*, Jun. 2005, pp. 13–18.

Wei Wei (S'02–M'06) received the B.S. degree in applied mathematics from Beijing University, Beijing, China, in 1992, the M.S. degree in statistics from Texas A&M University, College Station, in 2000, and the M.S. degrees in computer science and applied mathematics and Ph.D. degree computer science from the University of Massachusetts, Amherst, in 2004, 2004, and 2006, respectively.

He is currently a Senior Postdoctoral Research Associate with the Computer Science Department, University of Massachusetts, Amherst. His research interests are in the areas of computer networks, distributed embedded systems, and performance modeling.

Sharad Jaiswal received the B.E. degree in computer science and engineering from the Regional Engineering College, Trichy, India, in 1998, the M.S. degree in computer systems engineering from Boston University, Boston, MA, in 2000, and the Ph.D. degree in computer science from the University of Massachusetts, Amherst, in 2005.

He is a Member of Technical Staff with Alcatel-Lucent Bell Labs, Bangalore, India. His current interests include network traffic monitoring and modeling and wireless network architecture and design.

Jim Kurose (S'81–M'84–SM'91–F'97) received the Ph.D. degree in computer science from Columbia University, New York, NY, in 1984.

He is currently a Distinguished University Professor with the Department of Computer Science, University of Massachusetts, Amherst. He has been a Visiting Scientist with IBM Research, INRIA, Institut EURECOM, the University of Paris, LIP6, and Thomson Research Labs. With Keith Ross, he is the coauthor of the textbook *Computer Networking: A Top–Down Approach* (5th ed., Addison-Wesley, 2009). His research interests include network protocols and architecture, network measurement, sensor networks, multimedia communication, and modeling and performance evaluation.

Prof. Kurose has served as Editor-in-Chief of the IEEE TRANSACTIONS ON COMMUNICATIONS and was the founding Editor-in-Chief of the IEEE/ACM TRANSACTIONS ON NETWORKING. He has been active in the program committees for IEEE INFOCOM, ACM SIGCOMM, and ACM SIGMETRICS for a number of years, and has served as Technical Program Co-Chair for these conferences. He has received a number of awards for his educational activities, including the IEEE Taylor Booth Education Medal.

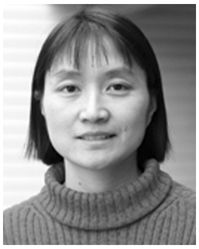
Don Towsley (M'78–SM'93–F'95) received the B.A. degree in physics and Ph.D. degree in computer science from the University of Texas at Austin in 1971 and 1975, respectively.

From 1976 to 1985, he was a member of the faculty of the Department of Electrical and Computer Engineering, University of Massachusetts, Amherst. He is currently a Distinguished Professor with the Department of Computer Science, University of Massachusetts, Amherst. He has held visiting positions at the IBM T. J. Watson Research Center, Yorktown Heights, NY; Laboratoire MASI, Paris, France; INRIA, Sophia-Antipolis, France; AT&T Labs—Research, Florham Park, NJ; and Microsoft Research Lab, Cambridge, U.K. His research interests include networks and performance evaluation.

Prof. Towsley is a Fellow of the Association for Computing Machinery (ACM), a member of ORSA, and Chair of IFIP Working Group 7.3. He has served as Editor-in-Chief of the IEEE/ACM TRANSACTIONS ON NETWORKING and on the Editorial Boards of the *Journal of the ACM* and the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, and has previously served on numerous other Editorial Boards. He was Program Co-Chair of the joint ACM SIGMETRICS and PERFORMANCE 1992 conference and the Performance 2002 conference. He has received the 2007 IEEE Koji Kobayashi Award, the 2007 ACM SIGMETRICS Achievement Award, the 2008 ACM SIGCOMM Achievement Award, the 1998 IEEE Communications Society William Bennett Best Paper Award, and numerous best conference/workshop paper awards.

Kyoungwon Suh (M'07) received the B.S. and M.S. degrees in computer engineering from Seoul National University, Seoul, Korea, in 1991 and 1993, respectively, the M.S. degree in computer science from Rutgers University, New Brunswick, NJ, in 2000, and the Ph.D. degree in computer science from the University of Massachusetts, Amherst, in 2007.

In 2008, he served as a Technical Consultant to NHN Corporation, Seoul, Korea, in the area of network security and content distribution network. He is currently an Assistant Professor with Illinois State University, Normal. His research interests include peer-to-peer and overlay networks, network measurement and inference, network security, and multimedia content distribution.



Bing Wang (M'02) received the B.S. degree in computer science from Nanjing University of Science & Technology, Nanjing, China, in 1994, the M.S. degree in computer engineering from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 1997, and the M.S. degrees in computer science and applied mathematics and Ph.D. degree in computer science from the University of Massachusetts, Amherst, in 2000, 2004, and 2005, respectively.

Afterwards, she joined the Computer Science and Engineering Department, University of Connecticut, Storrs, as an Assistant Professor. Her research interests are in computer networks, multimedia, and distributed systems.

Dr. Wang received the NSF CAREER Award in 2008.