# Identifying 802.11 Traffic from Passive Measurements Using Iterative Bayesian Inference

Wei Wei, Sharad Jaiswal, Jim Kurose, Don Towsley

Computer Science Department

140 Governors Drive

University of Massachusetts

Amherst, MA 01003-9264

**UMass Computer Science Technical Report 2005-47**

### Abstract

In this paper, we propose a classification scheme to differentiate Ethernet and WLAN TCP flows based on measurements collected passively at the edge of a large network. This classifier computes the fraction of wireless TCP flows, and the degree of belief that a TCP flow traverses a WLAN inside the network. The core of this classifier is an iterative Bayesian inference algorithm that we developed to obtain the maximum likelihood estimate (MLE) of these quantities. Our algorithm can handle any general two-class classification problem given the marginal distributions of these two classes. Numerical and experimental evaluations demonstrate that our classifier obtains accurate results and is insensitive to imprecise marginal distributions. We apply the classifier to various traces collected at the edge of the UMass campus network and infer that between 11-14% of all TCP flows coming into UMass campus traverse a 802.11 wireless link within the campus. We also detect wireless usage (through the use of private routers and access points) in areas not covered by the official wireless infrastructure.

## I. Introduction

The deployment and use of IEEE 802.11 wireless LANs (WLANs) has grown dramatically over the past few 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-end performance of wireless users. In this paper, we present a novel methodology to detect TCP flows that have traversed a 802.11 wireless network using measurements collected *passively* at the edge of a large (campus) network. 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 and may allow unauthorized access within the network. Finally, by monitoring the identified wireless traffic, one can infer the end-end performance of flows, thus keeping tab on the performance of the wireless network.

Identifying wireless traffic within a network is 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, since the host may be able to connect to both to a wired or a wireless network, or the host may be behind a NAT box, the IP address of a host may not provide any useful information about the type of its access network. Another approach towards estimating the extent of wireless traffic is to monitor and compute statistics from all access points in the network. However, this requires access to perhaps hundreds of such access points within a large network, many of them unknown, thus making this technique infeasible and impractical.

There has been some previous work on identifying hosts behind WLAN networks. In [1], the authors classify hosts to be behind either wired or wireless networks based on the RTTs of TCP connections established with remote hosts. However, this work relies on certain assumptions about wireless links, such as very low bandwidth and high loss rates, to differentiate wireless and wired links. These assumptions may not hold in current WLANs. More recently, in a previous work [2], we proposed a simple and efficient end-end scheme to classify the type of an access network (Ethernet, WLAN, or low bandwidth connections) of a remote host using packet pairs (e.g., two back-to-back packets). Different access networks were classified based on cut-off values (derived based on intrinsic properties of these networks) of median and entropy of the inter-arrival times of the packet pairs. The above two approaches rely on *active* measurements. In a network with hundreds or thousands of end-hosts, using active approaches to infer the extent of wireless usage requires coordination with these end hosts, and hence is impractical.

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. The classifier takes statistics computed over the inter-arrival time of so called *ACK-pairs*[1] as the input and computes the value of $\alpha$, the fraction of wireless TCP flows. In addition, for each TCP flow, this classifier determines $\beta$, the belief that this particular TCP flow traverses a WLAN inside the network. The core of this classifier is an iterative Bayesian inference algorithm that we developed to obtain the maximum likelihood estimate (MLE) of $\alpha$ and $\beta$. We prove that our iterative inference algorithm converges to the unique MLE of $\alpha$ and $\beta$. Furthermore, this inference algorithm can handle any general two-class classification problem given the marginal distributions of these two classes (i.e., Ethernet and WLAN in the context of this paper). Numerical and experimental evaluations demonstrate that our classification scheme obtains accurate results and is insensitive to imprecise marginal distributions.

We apply the classifier to various traces collected at a monitoring point placed at the gateway router of the University of Massachusetts, Amherst (UMass) campus network. We infer that between 11-14% of all TCP flows entering the UMass campus traverse a 802.11 wireless link within the campus. We also detect wireless usage (through the use of private routers and access points) in areas that are not covered by the official wireless infrastructure.

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

## A. Related work

Passive measurements have been used to infer Internet link lossiness [3], congestion sharing [4] and link capacity [5], and network performance discovery [6]. Our work is the first to identify WLAN traffic using passive measurements. The authors of [3] exploit Bayesian inference techniques to estimate the loss rates of the network links based on measurements collected at a server. Their results depend on the prior distribution. Our scheme, on the hand, uses an iterative Bayesian inference algorithm, where the prior distribution is updated during each iteration. We prove that our algorithm converges to the unique MLE regardless of the prior distribution. Packet inter-arrival times have been used in earlier works to detect a shared bottleneck [4], [7] and link capacity [8], [9], [10], [11], [5]. In [5], when TCP data packets were not available, ACK inter-arrival times were used to estimate link capacity and produces

---

[1]Informally, an ACK-pair refers to two ACKs generated in response to data packets that arrived close in time at the measurement point. A more precise definition will be given shortly.
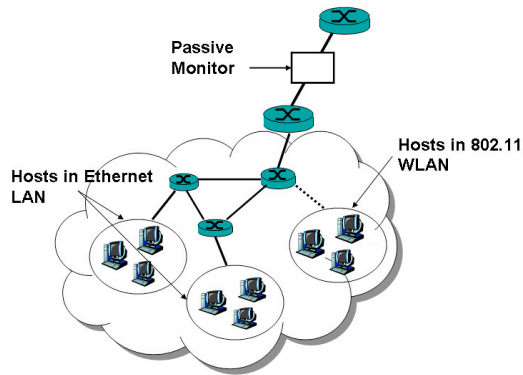
Fig. 1.   Problem setting: a monitoring point is located at the gateway router of the local network, capturing traffic coming in and going out of the network. The end hosts within this network are behind wired Ethernet or 802.11 WLAN.

lower quality results. Our approach, instead, utilizes the inter-arrival times of ACK-pairs to differentiate WLAN and Ethernet traffic based on the inherent differences of these two types of connections. A recent work estimates bottleneck bandwidth using TCP probes [12]. This approach, however, requires modifying TCP. Our approach is based on passive measurements of the ACKs, and requires no changes to TCP.

Previous work on wireless measurement has used direct measurement techniques  and focused on the performance and user behavior in wireless networks [13], [14]. We use an indirect approach to infer the percentage of the WLAN traffic and identify WLAN flows, which requires only a single monitoring point at the edge of the network and thus requires minimum cost and effort.

## II.  PROBLEM DEFINITION AND APPROACH

We now state our inference problem and describe, at a high-level, our approach towards 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 network. A monitoring point is located at the gateway router of this local network, 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 wireless 802.11 WLAN within the UMass network (2) for each TCP flow, what is the belief (probability) that this particular TCP flow traverses a 802.11 WLAN within the network.

This problem is complicated since the monitoring point is at the edge of the network, in the middle of the path between senders and receivers. Therefore, the measurements collected at the monitoring point may not provide accurate information on the characteristics of the sender and the receiver.

Our scheme 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 III, the inter-arrival times of ACK-pairs at the monitoring point differ significantly if the data packets and ACK-pairs traverse a wireless hop as compared to a wired Ethernet link. This difference is due to the intrinsic characteristics of Ethernet and WLAN. Our scheme exploits this difference to differentiate WLAN and Ethernet TCP flows.

In the next section, we present the analytical basis of our scheme, which demonstrates how the inter-ACK times will differ for TCP flows traversing a WLAN or an Ethernet. We then describe an iterative inference algorithm (the core of our classification scheme) along with the approach to identify ACK-pairs in practice.
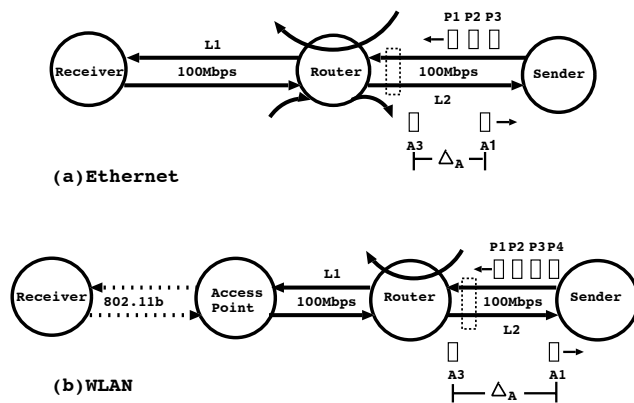
Fig. 2. Settings for the analysis: (a) Ethernet (b) 802.11b. The dashed rectangle between the sender and the router represents the monitoring point. The pair of ACKs, $A_1$ and $A_3$, forms an ACK-pair.

## III. AN ANALYTICAL BASIS

In this section, we carry out an analytical study that forms the foundation of our inference algorithm in Section IV. 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 questions, we consider an arbitrary TCP flow in which an outside server sends data 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)). The monitoring point lies between the sender and the receiver, at the edge of the local network. Let $\Delta$ denote the inter-arrival time of two data packets that arrive close in time at the monitoring point. The receiver returns a pair of ACKs corresponding to these two data packets, i.e., an ACK-pair, to the sender. Let $\Delta_A$ denote the inter-arrival time of this ACK-pair at the monitoring point, referred to as an *inter-ACK time*. We conjecture that, when the receiver uses a WLAN, $\Delta_A$ is larger than when the receiver uses Ethernet. Intuitively, this is due to two reasons. Firstly, in a WLAN, even if there is no contention in the channel, the receiver must wait for a random backoff interval after a previous successful transmission to avoid channel capture (see [2] and the references within). Therefore, this random backoff delay may be inserted between the ACK-pair, leading to a larger inter-ACK time. Secondly, 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 transmitted between the ACK-pair, and this again increases the inter-ACK time. Our analysis confirms this conjecture by demonstrating that the distributions of $\Delta_A$ under WLAN and Ethernet differ dramatically. On the other hand, our analysis also shows that these two distributions are not completely disjoint.

We now describe our analysis in more detail, starting with the assumptions and settings. We then present the analysis for wired Ethernet and wireless 802.11b channels. In the end, we briefly summarize the insights obtained from this analysis.

### A. Assumptions and settings

The settings for our analysis are shown in Fig. 2. In the figure, an outside server sends data to a receiver in the local network. In the first setting (Fig. 2(a)), the receiver uses Ethernet; in the second setting (Fig. 2(b)), the receiver uses 802.11b WLAN. In both settings, a router resides between the sender and the receiver, and connected to the sender by link $L_2$ with 100 Mbps bandwidth. The monitoring point is between the sender and the router, tapping into link $L_2$. When the receiver uses Ethernet (Fig. 2(a)), the router and the receiver are connected by link $L_1$ with 100

Mbps bandwidth (i.e., 100Mbps Ethernet). When the receiver uses 802.11b WLAN (Fig. 2(b)), an access point resides between the router and the receiver. The access point and the router are connected by link $L_1$ with bandwidth 100 Mbps; and the receiver is connected to the access point using 11 Mbps 802.11b. In both settings, the router is modeled as two $M/D/1$ queues, $Q_D$ and $Q_A$, in the direction of data packets (i.e., from the sender to the receiver) and ACKs (i.e., from the receiver to the sender) respectively. Let $\rho_D$ and $\rho_A$ denote the utilization of $Q_D$ and $Q_A$ respectively. In the Ethernet setting, cross traffic traverses both queues at the router. In the WLAN setting, cross traffic only traverses queue $Q_D$. We consider the ideal case in which no other traffic is on the path between the router and the receiver. In a WLAN, this implies that the access point is only utilized by the receiver.

TCP delayed ACK policy is commonly used in practice, implemented in both Windows and Linux [15], [16]. We therefore assume that the receiver implements this policy. To accommodate the effects of delayed ACK, we consider four packets $P_1$, $P_2$, $P_3$ and $P_4$ arriving at the monitoring point. Without loss of generality, we assume that packets $P_1$ and $P_3$ are acknowledged. Their corresponding ACKs $A_1$ and $A_3$ form an ACK-pair. Let $\Delta_A$ represent the inter-ACK time of ACKs $A_1$ and $A_3$ at the monitoring point. Let $\Delta$ denote the inter-arrival time of the data packets corresponding to $A_1$ and $A_3$ (namely, packets $P_1$ and $P_3$) at the monitoring point. We assume that all data packets are 1500 bytes and all ACKs are 40 bytes. Then $\Delta = 120 \times 2 = 240\mu s$ since the size of each data packet is 1500 bytes and the bandwidth of link $L_2$ is 100Mbps. Measurement studies show that the average packet size on the Internet is between 300 and 400 bytes [17], [18]. We assume that all packets in the cross traffic are 375 bytes for ease of calculation. Then, the transmission time of a cross traffic packet on a 100 Mbps link is $30\mu s$. For ease of exposition, we define a time unit of length $30\mu s$, and refer to it as a *packet transmission time*.

In both the Ethernet and WLAN settings, we are interested in the distribution of $\Delta_A$, i.e., the inter-ACK time of the ACK-pair $A_1$ and $A_3$. When calculating $\Delta_A$, the transmission time of an ACK is ignored since it is negligible. Suppose $m$ samples of inter-ACK time are observed in a TCP flow. Let $\xi_{.5}^{(m)}(\Delta_A)$ denote the median inter-ACK time given these $m$ samples. We are interested in the median instead of the mean since median is less sensitive outliers in the network measurements [19].

## B. Analysis of Ethernet

Let $\Delta_D$ be the inter-departure time of packet $P_1$ and $P_3$ at queue $Q_D$ (i.e., the queue in the direction of data packets at the router). Discretize $\Delta_D$ using the packet transmission time as the time unit, and denote the discretized value as $I_D$, that is, $I_D = \lfloor \Delta_D/30 \rfloor$. Discretize $\Delta_A$ in the similar manner and denote the discretized value as $I_A$, $I_A = \lfloor \Delta_A/30 \rfloor$.

*Lemma 1:* Let $Z = I_D - 8$. When $\rho_D = 1$, $Z$ follows a Poisson distribution with the parameter of 8 time units.

**Proof:** When $\rho_D = 1$, the probability of queue $Q_D$ being empty is 0. Hence, all the cross traffic arriving between packet $P_1$ and $P_3$ contribute to the inter-departure time of $P_1$ and $P_3$ at queue $Q_D$. By assumption, the transmission time of each cross traffic packet is 1 time unit at queue $Q_D$. Therefore, each cross traffic packet arriving between $P_1$ and $P_3$ increases the value of $I_D$ by 1 time unit. One component of $I_D$ is the transmission time of packet $P_1$ and $P_2$, which is $2 \times 120\mu s$ or 8 time units. The other component of $I_D$ is the transmission time of cross traffic packets between $P_1$ and $P_3$, denoted as $Z$. Then $Z = I_D - 8$. Since the transmission time of packet $P_1$ and $P_2$ is 8 time units, on average, the interval between $P_1$ and $P_3$ is 8 time units. By the assumption that queue $Q_D$ is a $M/D/1$ queue, $Z = I_D - 8$ follows a Poisson distribution with the parameter of 8 time units. ∎

We next state a lemma on the conditional distribution of the inter-ACK time given $I_D$.

*Lemma 2:* Suppose $I_D = x$ time units. When $\rho_A = 1$, the conditional distribution of $I_A$ given $I_D$ follows a Poisson distribution with the parameter of $x$ time units.

**Proof:** Since $\rho_A = 1$, the probability of queue $Q_A$ (in the direction of ACKs at the router) being empty is $0$. Hence all of the cross traffic arriving between ACK $A_1$ and $A_3$ contribute to the inter-departure time of ACK $A_1$ and $A_3$ at the router. Since we ignore the transmission time of ACK $A_1$ and no other traffic is between the router and the receiver, the spacing between ACK $A_1$ and $A_3$ is the same as the inter-departure time of packet $P_1$ and $P_3$ at queue $Q_D$, i.e., $I_D$. Therefore, the conditional distribution of $I_A$ given $I_D = x$ follows a Poisson distribution with the parameter of $x$ time units. ∎

*Lemma 3:* When $\rho_D = \rho_A = 1$,

$$P(I_A < x) = \sum_{y=8}^{\infty} \frac{8^{y-8}e^{-8}}{(y-8)!} \sum_{i=0}^{x} \frac{y^i e^{-y}}{i!}$$

**Proof:** This follows directly from Lemma 1 and Lemma 2. ∎

We now state two theorems on the distribution of the inter-ACK time and the median inter-ACK time respectively.

*Theorem 1:* When $0 < \rho_D, \rho_A \leq 1$, $P(\Delta_A > 990\mu s) < 0.0013$.

**Proof:** When $\rho_D = \rho_A = 1$, from Lemma 3, by direct calculation, we have $P(I_A > 33) < 0.0013$, which is equivalent to

$$P(\Delta_A > 990\mu s) < 0.0013$$

The above result also holds when $0 < \rho_D < 1$ and $0 < \rho_A < 1$ due to reasons below. When $0 < \rho_D < 1$, the inter-departure time of data packets at queue $Q_D$ is smaller than that when $\rho_D = 1$, since not all cross traffic contribute to the inter-departure time of packet $P_1$ and $P_3$. Similarly, when $0 < \rho_A < 1$, the inter-departure time of ACK $A_1$ and $A_3$ at the router is smaller than that when $\rho_A = 1$. ∎

*Theorem 2:* **(Ethernet median inter-ACK time)** When $0 < \rho_D, \rho_A \leq 1$, we have $P(\xi_{.5}^{(m)}(\Delta_A) > 990\mu s) \approx 0$, for $m = 5$ and $10$.

**Proof:** Let $x = P(\Delta_A > 990\mu s)$. Then

$$P(\xi_{.5}^{(m)}(\Delta_A) > 990\mu s) = \sum_{i=\lceil m/2 \rceil}^{m} \binom{m}{i} x^i (1-x)^{m-i}.$$

When $x = 0.0013$, by direct calculation, we have $P(\xi_{.5}^{(m)}(\Delta_A) > 990\mu s) \approx 0$, for $m = 5$ and $10$. We next show that $P(\xi_{.5}^{(m)}(\Delta_A) > 990\mu s)$ is an increasing function of $x$.

$$\frac{d}{dx}\left( \sum_{i=\lceil m/2 \rceil}^{m} \binom{m}{i} x^i (1-x)^{m-i} \right)$$

$$= \sum_{i=\lceil m/2 \rceil}^{m} \frac{m!}{(i-1)!(m-i)!} x^{i-1}(1-x)^{m-i} - \sum_{i=\lceil m/2 \rceil}^{m-1} \frac{m!}{i!(m-i-1)!} x^i (1-x)^{m-i-1}$$

$$= \sum_{j=\lceil m/2 \rceil - 1}^{m-1} \frac{m!}{j!(m-j-1)!} x^j (1-x)^{m-j-1} - \sum_{i=\lceil m/2 \rceil}^{m-1} \frac{m!}{i!(m-i-1)!} x^i (1-x)^{m-i-1}$$

$$= \frac{m!}{(\lceil m/2 \rceil - 1)!(m - \lceil m/2 \rceil)!} x^{\lceil m/2 \rceil - 1}(1-x)^{m-\lceil m/2 \rceil}$$

$$> 0$$

when $0 < x < 1$. That is, $P(\xi_{.5}^{(m)}(\Delta_A) > 990\mu s)$ is an increasing function of $x$. By Theorem 1, $P(\Delta_A > 990\mu s) < 0.0013$ when $0 < \rho_D, \rho_A \le 1$. Thus we have the desired results. ∎

The above result indicates that for a 100Mbps Ethernet, the median inter-ACK time is rarely over $990\mu s$. In the following, we show that for a 802.11b WLAN, the median inter-ACK time is over $1000\mu s$ in most cases. Hence, the median inter-ACK time is a useful statistic in differentiating WLAN and Ethernet flows.

### C. Analysis of 802.11b WLAN

Our analysis utilizes the following knowledge of 802.11b. In 11Mbps 802.11b, transmitting a TCP data packet requires at least $508\mu s$ [20]. Furthermore, under ideal conditions (i.e., perfect wireless channel and no contention), the receiver waits for a random amount of time uniformly distributed in $[0, 620]\mu s$ before transmitting a packet (see [2] and the reference within). Therefore, under ideal conditions, the transmission time for a data packet (1500 bytes) and an ACK (40 bytes) is uniformly distributed in $[1570, 2190]\mu s$ and $[508, 1128]\mu s$ respectively. Our analysis below assumes ideal conditions.

We first state a lemma indicating that the inter-arrival time of two consecutive data packets at the access point is less than $1570\mu s$ with probability very close to 1.

*Lemma 4:* Let $\Delta_{i,i+1}^D$ denote the inter-arrival time of $P_i$ and $P_{i+1}$ at the access point, $i = 1, 2, 3$. Then, $P(\Delta_{i,i+1}^D < 1570\mu s) \approx 1$.

**Proof:** We prove the result for $\Delta_{1,2}^D$ by considering the following two cases.

- Case 1: $\rho_D = 1$. Let $I_{1,2}^D$ denote the discretized $\Delta_{1,2}^D$ using the packet transmission time as the time unit. That is, $I_{1,2}^D = \lfloor \Delta_{1,2}^D/30 \rfloor$ time units. In this case, similar to the proof for Theorem 1, we prove that $(I_{1,2}^D - 4)$ follows a Poisson distribution with the parameter of 4 time units. Hence,

$$
\begin{aligned}
P(\Delta_{1,2}^D < 1570\mu s) &\ge P(\Delta_{1,2}^D \le 1560\mu s) \\
&= P(I_{1,2}^D \le 52) \\
&= \sum_{x=4}^{52} \frac{4^{x-4}e^{-4}}{(x-4)!} \approx 1
\end{aligned}
$$

- Case 2: $\rho_D < 1$. In this case, the value of $\Delta_{1,2}^D$ is less than that when $\rho_D = 1$. Hence, $P(\Delta_{1,2}^D < 1570\mu s) \approx 1$ also holds.

The proofs for the result on $\Delta_{2,3}^D$ and $\Delta_{3,4}^D$ are similar. ∎

Note that the above result also holds when $\Delta = 390\mu s$. These results are to be used when identifying ACK-pairs in practice. Since it takes at least $1570\mu s$ to transmit a data packet, the above lemma indicates that packet $P_{i+1}$ is ready to be transmitted after packet $P_i$ is transmitted, where $i = 1, 2, 3$. We next state a theorem on the distribution of the inter-ACK time. After receiving packet $P_1$, the receiver returns ACK $A_1$ to the sender. At this point, ACK $A_1$ and packet $P_2$ contend for the wireless channel. Let $p$ denote the probability that ACK $A_1$ obtains the channel and hence transmits earlier than packet $P_2$. We assume that $p$ can take any value in $[0, 1]$ since the contention between ACK $A_1$ and packet $P_2$ can be affected by many factors, such as when ACK $A_1$ reaches the MAC layer, when packet $P_2$ can be transmitted, etc. If $A_1$ transmits later than $P_2$, then $A_1$ contend with $P_3$ for the wireless channel. In this case, we assume that $A_1$ and $P_3$ are equal likely to obtain the channel since they both can be transmitted immediately.

*Theorem 3:* $P(\Delta_A \le 1000\mu s) < 0.1984$

**Proof:** In order for $\Delta_A \leq 1000\mu s$ to hold, no data packet can be transmitted between ACK $A_1$ and $A_3$, since the transmission time of a data packet is at least $1570\mu s$. Therefore, only two sequences of data and ACK transmission are possible: $P_2, P_3, A_1, A_3, P_4$ or $P_2, P_3, P_4, A_1, A_3$. The probability that the first sequence occurs is: $(1-p) \times 1/2 \times 1/2 \times p = p(1-p)/4$. The probability that the second sequence occurs is: $(1-p) \times 1/2 \times 1/2 = (1-p)/4$. To satisfy $\Delta_A \leq 1000\mu s$, we also require the transmission time of $A_3$ to be less than $1000\mu s$. The probability of this condition being satisfied is $(1000 - 508)/620 = 492/620$. Therefore,

$$\begin{aligned}
P(\Delta_A \leq 1000\mu s) &= [p(1-p)/4 + (1-p)/4]492/620 \\
&= \frac{1}{4}(1-p^2)\frac{492}{620} < 0.1984
\end{aligned}$$

∎

*Theorem 4:* **(802.11b median inter-ACK time)** $P(\xi_{0.5}^{(10)}(\Delta_A) \leq 1000\mu s) < 0.0062$, $P(\xi_{0.5}^{(5)}(\Delta_A) \leq 1000\mu s) < 0.0567$.

**Proof:** The proof is similar to that for Theorem 2. ∎

The above result shows that, when the receiver uses 802.11b, the median inter-ACK time is rarely lower than $1000\mu s$ under ideal conditions for relatively large sample sizes. Under more realistic conditions (e.g., noisy wireless channel and with contention), the median inter-ACK time may be even higher. However, we also observe that, for small sample sizes (e.g., $m = 5$), there is a significant probability that the median inter-ACK time is smaller than $1000\mu s$.

*D. Summary*

In the above, we analyzed the median inter-ACK time when the receiver uses 100Mbps Ethernet and 802.11b WLAN respectively and showed that they can differ significantly. However, as we have observed, the distributions of the median inter-ACK time under Ethernet and WLAN have overlap. When 10Mbps Ethernet and/or 802.11g are present, this overlap may be even more dramatic (as we shall see in Section VI). This result implies that deterministic classification of WLAN and Ethernet traffic simply by a cut-off value of the median inter-ACK time will not provide accurate results. In the next section, we propose an inference algorithm to probabilistically classify WLAN and Ethernet flows.

## IV. ITERATIVE BAYESIAN INFERENCE ALGORITHM

In this section, we design an iterative Bayesian inference algorithm to differentiate WLAN and Ethernet traffic. Analytical results in the previous section indicates that median inter-ACK time is useful to differentiate WLAN and Ethernet flows. We refer to the median inter-ACK time of a TCP flow as an *observation*. The observation of a WLAN flow is referred to as a *WLAN observation*. Similarly, the observation of an Ethernet flow is referred to as an *Ethernet observation*. Let $O$ denote a set of $n$ observations corresponding to $n$ TCP flows, $O = \{x_i\}_{i=1}^n$, where $x_i$ is the observation for the $i$th flow. A subset of the observations in $O$ are WLAN observations and the rest are Ethernet observations.

Let $\alpha$ denote the fraction of WLAN observations in all observations. Let $\beta_i$ denote the belief that the $i$th TCP flow uses a WLAN given the observation $x_i$. Let $W$ and $E$ denote the event that a TCP flow is a WLAN and an Ethernet flow respectively. Let $X$ be the random variable representing median inter-ACK time. Then, $\beta_i = P(W \mid X = x_i)$. Note that, $\beta_i$ close to 1 indicates a high belief that the $i$th TCP flow is a WLAN flow; $\beta_i$ close to 0 indicates a high belief that the $i$th TCP flow is an Ethernet flow. Let $P(X = x \mid W)$ denote the distribution of $X$ given a WLAN flow, referred to as *WLAN observation distribution*. Similarly, $P(X = x \mid E)$ denotes the distribution of $X$ given an Ethernet flow, referred to as *Ethernet observation distribution*.

In the following, we first describe how to obtain the MLE of $\alpha$ and $\{\beta_i\}$ from the set of observations, $O$. We then design an iterative Bayesian inference algorithm to solve for $\alpha$ and $\{\beta_i\}$. Our inference algorithm requires Ethernet and WLAN observation distributions. In Section V, we describe how to obtain them in practice.

*A. Maximum Likelihood Estimate of $\alpha$ and $\beta_i$*

Let $p_i := P(X = x_i \mid W)$ denote the probability of observing $x_i$ given that the $i$th TCP flow traverses a WLAN. Similarly, let $q_i := P(X = x_i \mid E)$ denote the probability of observing $x_i$ given that the $i$th TCP flow traverses an Ethernet. These two probabilities can be obtained directly from the WLAN and Ethernet observation distributions. Let $\hat{\alpha}$ and $\hat{\beta}_i$ denote the MLE of $\alpha$ and $\beta_i$ respectively. Then $\hat{\beta}_i$ can be obtained directly from $\hat{\alpha}$. This is because, using Bayesian formula, $\beta_i$ can be represented as a function of $\alpha$

$$\beta_i = \frac{\alpha p_i}{\alpha p_i + (1 - \alpha) q_i} \tag{1}$$

We next derive the MLE for $\alpha$. Assume that the observations are independent. The likelihood function of observing the set of observations $O = \{x_i\}_{i=1}^n$ is

$$
\begin{aligned}
L(\alpha \mid O) &= \prod_{i=1}^{n} P(X = x_i) \tag{2} \\
&= \prod_{i=1}^{n} (\alpha p_i + (1 - \alpha) q_i) \tag{3} \\
&= \prod_{i=1}^{n} ((p_i - q_i)\alpha + q_i) \tag{4}
\end{aligned}
$$

We assume that $\sum_{i=1}^{n}(p_i - q_i)^2 > 0$. Otherwise, it is impossible to estimate $\alpha$ since any value of $\alpha$ is equally likely. Taking log on both sides of (4), we have

$$l(\alpha \mid O) := \log L(\alpha \mid O) = \sum_{i=1}^{n} \log((p_i - q_i)\alpha + q_i) \tag{5}$$

The MLE is obtained by taking the derivative of $l(\alpha \mid O)$ and setting the derivative to 0:

$$\frac{dl(\alpha \mid O)}{d\alpha} = \sum_{i=1}^{n} \frac{p_i - q_i}{(p_i - q_i)\alpha + q_i} = 0 \tag{6}$$

Let $f(\alpha)$ denote $\sum_{i=1}^{n} \frac{p_i - q_i}{(p_i - q_i)\alpha + q_i}$. It is easily observed that $f(\alpha)$ is a decreasing function of $\alpha$. If $f(0) > 0$ and $f(1) < 0$, $f(\alpha)$ has a unique solution in $(0, 1)$. This unique solution $\hat{\alpha}$ is the MLE of $\alpha$. Otherwise, we have $f(\alpha) > 0$ or $f(\alpha) < 0$ for $\alpha \in (0, 1)$. This implies that $l(\alpha \mid O)$ is an increasing or decreasing function. Hence, the MLE is achieved either at 1 or 0. Combining these cases, the MLE of $\alpha$ is unique. Since $\beta_i$ is a function of $\alpha$, the MLE of $\beta_i$, is also unique.

Solving (6) for $\hat{\alpha}$ directly is difficult. This is because (6) is equivalent to a $(n - 1)$-th order polynomial equation, which is not solvable using only rational operations and finite root extractions when $(n - 1) \geq 5$ [21]. We therefore design an iterative algorithm to solve for $\hat{\alpha}$ and $\hat{\beta}_i$ in the following.

## B. Iterative Bayesian inference algorithm

Let $\alpha^{(k)}$ and $\beta_i^{(k)}$ denote the values of $\alpha$ and $\beta_i$ at the $k$-th iteration. The iterative Bayesian algorithm is:

$$\alpha^{(0)} = \alpha_0, 0 < \alpha_0 < 1. \tag{7}$$

$$\beta_i^{(k)} = \frac{p_i \alpha^{(k)}}{p_i \alpha^{(k)} + q_i(1 - \alpha^{(k)})}, 1 \le i \le n. \tag{8}$$

$$\alpha^{(k+1)} = \frac{1}{n} \sum_{i=1}^{n} \beta_i^{(k)}. \tag{9}$$

where $k \ge 0$, $\alpha_0$ is the initial value for $\alpha$. Equation (8) follows directly from equation (1). Equation (9) is by the definition of $\alpha$.

This is an EM (Expectation and Maximization) algorithm [22]. In the E-step, we calculate the expected number of WLAN flows. In the M-step, we calculate the MLE of $\alpha$ using (9). The convergence of this iterative Bayesian inference algorithm follows the convergence property of an EM algorithm [22]. We prove that this algorithm converges to the unique MLE of $\alpha$ and $\beta_i$, as stated in the following theorem.

*Theorem 5:* Let $\bar{\alpha} = \lim_{k \to \infty} \alpha^{(k)}$ and $\bar{\beta}_i = \lim_{k \to \infty} \beta_i^{(k)}$. The iterative Bayesian inference algorithm converges to the unique $\bar{\alpha}$ and $\bar{\beta}_i$. Furthermore, $\bar{\alpha}$ is the MLE of $\alpha$ and $\bar{\beta}_i$ is the MLE of $\beta_i$, where $i = 1, 2, \cdots, n$.

**Proof:** In (8) and (9), let $k \to \infty$. Then

$$\bar{\beta}_i = \frac{p_i \bar{\alpha}}{p_i \bar{\alpha} + q_i(1 - \bar{\alpha})} \tag{10}$$

$$\bar{\alpha} = \frac{1}{n} \sum_{i=1}^{n} \bar{\beta}_i \tag{11}$$

Combining ( 10) and ( 11), we get

$$n\bar{\alpha} = \sum_{i=1}^{n} \bar{\beta}_i = \sum_{i=1}^{n} \frac{p_i \bar{\alpha}}{p_i \bar{\alpha} + q_i(1 - \bar{\alpha})} \tag{12}$$

That is

$$n = \sum_{i=1}^{n} \frac{p_i}{(p_i - q_i)\bar{\alpha} + q_i} \tag{13}$$

$$\tag{14}$$

or

$$\sum_{i=1}^{n} \frac{(p_i - q_i)(1 - \bar{\alpha})}{(p_i - q_i)\bar{\alpha} + q_i} = 0 \tag{15}$$

Consider $0 \le \bar{\alpha} < 1$. By dividing $(1 - \bar{\alpha})$ on both sides of (15), we have

$$\sum_{i=1}^{n} \frac{p_i - q_i}{(p_i - q_i)\bar{\alpha} + q_i} = 0 \tag{16}$$

This is the same as (6), i.e., the condition satisfied by the MLE $\hat{\alpha}$. This implies that, if the inference algorithm converges to a solution in $(0, 1)$, then the solution is the unique MLE.

Combining (8) and (9), we have

$$\alpha^{(k+1)} = \frac{1}{n} \sum_{i=1}^{n} \frac{p_i \alpha^{(k)}}{p_i \alpha^{(k)} + q_i (1 - \alpha^{(k)})} \tag{17}$$

If $0 < \alpha^{(k)} < 1$, when $q_i = 0, 1 \le i \le n$, we have $\alpha^{(k+1)} = 1$. Otherwise, we have $0 < \alpha^{(k+1)} < 1$. This implies that if there is a solution in (0, 1). Then this algorithm converges to the solution if $0 < \alpha^{(0)} < 1$. Also, if the only solution is 1, then we have $p_i > 0, q_i = 0$ for $i = 1, \cdots, n$. Then this algorithm converges to 1 after the first iteration when $0 < \alpha^{(0)} < 1$. Similarly, when the only solution is 0, we have $p_i = 0, q_i > 0$ for $i = 1, \cdots, n$. Then this algorithm converges to 0 after the first iteration when $0 < \alpha^{(0)} < 1$.

By the invariance property of maximum likelihood estimators [23], $\bar{\beta}_i$ is the MLE of $\beta_i$, $i = 1, 2, \cdots, n$. ∎

## V. CLASSIFICATION SCHEME

We now design a classification scheme to determine the fraction of WLAN TCP flows and the belief that a TCP flow traverses a WLAN, for a given collection of TCP flows. The core of the classifier is the iterative Bayesian inference algorithm presented in the previous section. In the following, we first describe how to obtain ACK-pairs and observation distributions in practice. We then describe in detail how the classifier operates.

### A. Identifying ACK-pairs

We refer to two successive ACKs as an ACK-pair if the inter-arrival time of their corresponding data packets at the monitoring point is less than a threshold $T$. In our experiments, we set $T$ to $250\mu s$ or $400\mu s$ based on our analysis in Section III (Lemma 4 holds for both $\Delta = 240\mu s$ and $390\mu s$). In practice, to identify ACK-pairs, we also need to consider several practical issues (e.g., delayed ACK and packet retransmission). Suppose that the TCP receiver implements the delayed-ACK policy, i.e., it releases an ACK after receiving two packets, or if the delayed-ACK timer is triggered after the arrival of a single packet. Now consider two successive ACKs, $A_i$ and $A_j$, at the monitoring point. If $A_j$ is transmitted by the receiver because the delayed-ACK timer is triggered, then $A_j$ is not released immediately upon the arrival of its corresponding data packet. Therefore, the inter-ACK time of these two ACKs does not reflect the characteristics of the access link, and hence are not an ACK-pair. However, before we can ascertain that an ACK is triggered because the delayed-ACK timer goes off, we must first infer whether the TCP receiver in fact implements the delayed-ACK policy. For every monitored TCP connection, we initially assume that the receiver does not implement the delayed-ACK policy. If at a subsequent point, an ACK acknowledges at least two data packets, we infer that the TCP receiver implements the delayed-ACK policy. This simple inference mechanism on delayed-ACK policy can be incorrect under several scenarios. We now discuss these scenarios and their impact on the effectiveness of our inference mechanism.

- Scenario 1: The end-host does implement the delayed-ACK policy, but the monitoring point only observes ACKs that acknowledge one data packet (i.e. each of these ACKs were released because the delayed ACK timer is triggered after the arrival of a single packet). In this scenario, we prevent using these ACKs as ACK-pairs by requiring the inter-ACK time of an ACK-pair to be lower than $T_A$. In our experiments, we set $T_A$ to 200 ms, which is the typical value of the delayed-ACK timer in most operating systems.

- Scenario 2: The end-host does not implement the delayed-ACK policy, but because the previous ACK is lost before it reaches the monitoring point, the monitoring point mistakenly infers that the end-host implements the delayed-ACK policy. If the monitoring point subsequently observes the receiver actually ACKs every packet, it would infer that each of these ACKs are triggered because the delayed-ACK timer goes off and ignores these ACKs. As a result, fewer ACK-pairs are produced for our analysis. However, since most OS stacks do implement the delayed-ACK policy, and we assume that the ACK loss rates to be low, we do not expect this scenario to significantly impact our analysis.

To summarize, we eliminate ACK-pairs that only acknowledge one packet (since this indicates the trigger of the delayed-ACK timer) when a connection does implement the delayed ACK policy. As mentioned before, we also eliminate all ACK-pairs with inter-ACK interval greater than $T_A$. Finally, we exclude all ACKs whose corresponding data packets have been retransmitted or reordered. The remaining ACK-pairs are then considered suitable for analysis.

### B. Obtaining observation distributions

We refer to a set of TCP flows from which the WLAN or Ethernet observation distribution is obtained as a *training set*. A training set contains a single type of TCP flows. For instance, a training set to obtain WLAN (or Ethernet) observation distribution contains only WLAN (or Ethernet) TCP flows. For a TCP flow, if the number of ACK-pairs in the flow is no less than $N$, we refer to this TCP flow as a *qualified TCP flow*. In our experiments, we set $N$ to 1, 2 or 5. We next describe how to obtain the observation distribution from a training set. In the training set, we first identify a set of qualified TCP flows. In each qualified TCP flow, we obtain the median inter-ACK time over all ACK-pairs (there may exist larger than $N$ ACK-pairs). Suppose that a set of $n_t$ qualified TCP flow are identified in the training set. Let $x_i$ denote the set of median the inter-ACK times of the $i$th qualified TCP flows. The value of $x_i$ is discretized as follows: if $x_i$ is smaller than 1ms, it is discretized to be a multiple of $50\mu s$; otherwise, it is discretized to be a multiple of 1 ms. Then, the observation distribution is obtained from the discretized value of $x_i$, $i = 1, 2, \ldots, n_t$.

Obtaining a training set from a trace with multiple types of TCP flows requires knowledge of the local network. In general, a training set can be obtained based on a block of IP addresses that are *known* to use a specific type of access link. For instance, we obtain the training set for WLAN in the UMass campus network from a block of IP addresses reserved for WLAN (see Section VI).

In practice, multiple types of Ethernet links may be used by a local network. Ethernet links of significant different bandwidth may exhibit different observation distributions. For instance, two types of Ethernet (10Mbps half-duplex and 100Mbps Ethernet) are used in the UMass campus network and their observation distributions differs dramatically (see Section VI-A). We next describe how to obtain the Ethernet observation distribution in this case. Suppose two types of Ethernet links, $E_1$ and $E_2$, are used in the local network and the fraction of $E_1$ in all Ethernet links is $\theta$. In practice, $\theta$ can be estimated from the ratio of $E_1$ jacks in all Ethernet jacks. Then Ethernet observation distribution can be obtained as a weighted sum of $E_1$ and $E_2$ observation distributions in which the weights are $\theta$ and $1-\theta$ respectively. We investigate the impact of imprecise estimate of $\theta$ on the classification result in Section VI-B.

### C. Applying the classification scheme

Given a collection of TCP flows, our classifier operates as follows. It first identifies a group of qualified TCP flows in (based on the threshold $T$ and $N$). For each qualified TCP flow, the classifier obtains the median inter-ACK time in this flow. Finally, the set of the median inter-ACK time over all qualified TCP flows and the observation distributions are fed into the iterative Bayesian inference algorithm to obtain inference results.
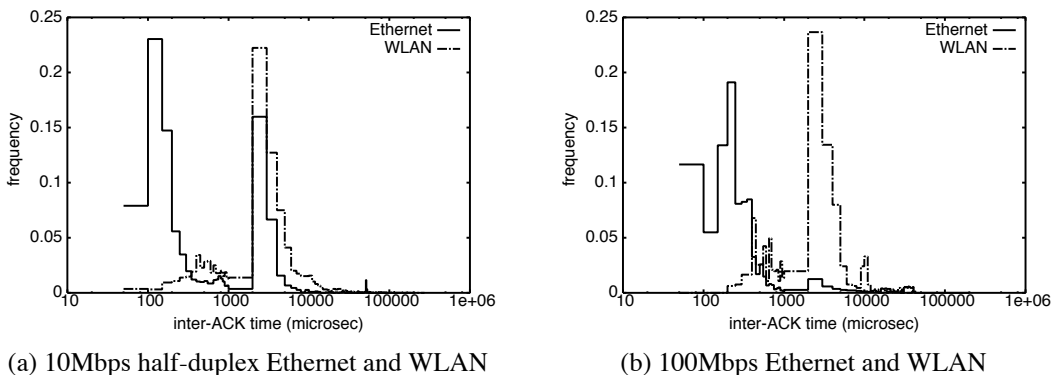
(a) 10Mbps half-duplex Ethernet and WLAN    (b) 100Mbps Ethernet and WLAN

Fig. 3.  Observation distributions, $T = 400\mu s$, $N = 1$.

## VI. PERFORMANCE EVALUATION

In this section, we evaluate the performance of our classifier numerically and experimentally. The numerical evaluation allows us to systematically investigate the impact of various parameters on the performance of our classifier. The experimental evaluation, based on real network traces, focuses on the impact of practical issues (e.g., measurement noise). All traces used in this paper are collected by the same measurement system connected via an optical splitter to the UMass commercial access link, i.e., the link connecting UMass campus network to the commercial network. All packets traversing this link are passed on to the monitoring equipment. A packet capture card (called a *DAG* card [24]) copies all packet headers to disk along with accurate time stamps.

Our evaluation is carried out on *testing sets* that are constructed to represent various scenarios. For the numerical evaluation, we construct testing sets by generated observations. For the experimental evaluation, we construct testing sets based on real network traces. The performance metric is *inference error*, defined as the difference between our inferred $\alpha$ (the fraction of WLAN flows) and the actual $\alpha$. We first describe how we construct training sets and then describe the evaluation results.

### A. Constructing training sets

Our training sets are constructed by extracting TCP flows from a group of traces (collected between February and April, 2005) based on our knowledge of the UMass campus network. The UMass campus supports a public 802.11 network that provides wireless access to campus users within certain public places, such as the libraries, the campus eateries, etc. A block of IP addresses within the UMass domain is reserved for this public WLAN. We construct the training set for WLAN by extracting TCP flows using these IP addresses from the collected traces. Two main types of Ethernet, 10Mbps half-duplex and 100Mbps Ethernet, are used in the UMass campus network. The majority of the Ethernet connections are 10Mbps half-duplex Ethernet; 100Mbps Ethernet is used in several academic departments, e.g., Computer Science Department. In the rest of this paper, all 10Mbps Ethernet refers to 10Mbps half-duplex Ethernet. We construct training sets for 10Mbps and 100Mbps Ethernet separately. The training set for 10Mbps Ethernet consists of TCP flows from a block of IP addresses belonging to two academic departments. All hosts in these two departments are on the 10Mbps half-duplex LAN and have no access to any wireless network. The training set for 100Mbps Ethernet consists of TCP flows on the 100Mbps Ethernet in the Computer Science Department.

In each training set, we vary the threshold to identify ACK-pairs, $T$, to be 250 or $400\mu s$ and the threshold to identify qualified TCP flows, $N$, to be $1, 2$ or $5$. The numbers of qualified TCP flows in the training sets under the various parameters are listed in Table I. As expected, the number of qualified TCP flows decreases as $T$ decreases and $N$ increases. For each $T$ and $N$, we obtain the observation distribution from the qualified TCP flows in each training

TABLE I

NUMBER OF QUALIFIED TCP FLOWS IN THE TRAINING SETS.

| $N$ | 10Mbps Ethernet | | 100Mbps Ethernet | | WLAN | |
|---|---|---|---|---|---|---|
| | $T = 250\mu s$ | $T = 400\mu s$ | $T = 250\mu s$ | $T = 400\mu s$ | $T = 250\mu s$ | $T = 400\mu s$ |
| 1 | 4712 | 5253 | 32514 | 35640 | 24050 | 27165 |
| 2 | 2019 | 2547 | 18527 | 21128 | 7749 | 10873 |
| 5 | 710 | 1002 | 9145 | 10943 | 2048 | 3327 |
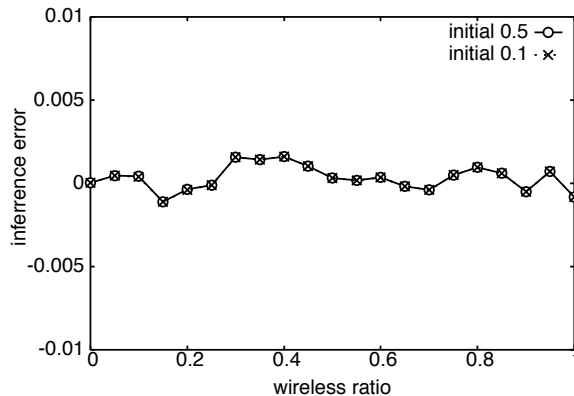


Fig. 4. Inference error in a testing set containing 10Mbps half-duplex Ethernet and WLAN, $T = 400\mu s$, $N = 1$.

set. Fig. 3(a) compares the observation distribution for 100Mbps Ethernet and WLAN. Fig. 3(b) compares the observation distribution for 100Mbps Ethernet and WLAN. In both figures, $T = 400\mu s$ and $N = 1$. As shown in Section III, although the observation distributions for WLAN and Ethernet (either 10Mbps half-duplex or 100Mbps) differ dramatically, they are not completely disjoint. The overlap between the 10Mbps-Ethernet and WLAN observation distributions ranges from 30% to 50%. Between the 100Mbps-Ethernet and WLAN observation distributions, the overlap is smaller (around 10% to 20%). The observation distribution for 10Mbps half-duplex Ethernet differs significantly from that for 100Mbps Ethernet[2].

*B. Numerical Evaluation*

Our numerical evaluation is conducted in two scenarios. In the first scenario, the Ethernet and WLAN observation distributions are known exactly. In the second scenario, the observation distributions are not known exactly and our focus is on the sensitivity of the classifier to imprecise observation distributions. In both scenarios, an observation is generated randomly according to its corresponding observation distribution. A testing set contains $n$ observations in which the fraction of WLAN observations is $\alpha$. We fix $n$ to be $100,000$, since a similar number of qualified TCP flows are identified in Section VII. We vary $\alpha$ in the range of $0$ to $1$.

In the first scenario, we construct two types of testing sets: the first type contains WLAN and 10Mbps-Ethernet observations; the second type contains WLAN and 100Mbps-Ethernet observations. Fig. 4 shows one result for a testing set containing 10Mbps-Ethernet and WLAN observations, where $T = 400\mu s$ and $N = 1$. The initial value for $\alpha$ in our iterative inference algorithm is $0.5$ or $0.1$. As shown in the figure, our classifier obtains the unique MLE of $\alpha$ regardless of the initial values for $\alpha$. The inference error is bounded by $0.002$, despite of the significant overlap of the

[2]We believe that the half-duplex nature of this medium is predominantly responsible for this difference (along with the difference in bandwidth).
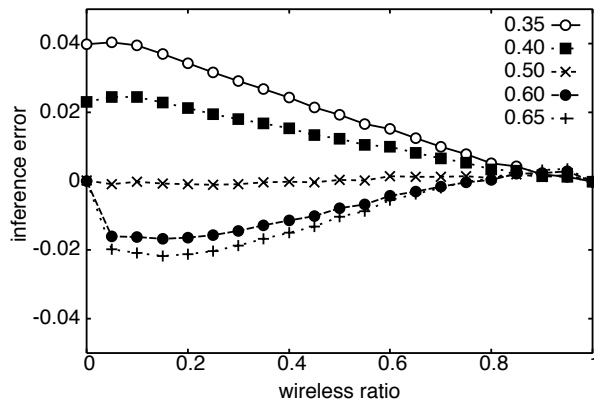
Fig. 5. Inference error for a testing set with $\theta = 0.5, T = 400\mu s, N = 1$. The estimation for $\theta$ is varied from 0.35 to 0.65.

two observation distributions (see Fig. 3(a)). Under the same $T$ and $N$, the inference error for 100Mbps Ethernet and WLAN is even smaller. The results for other values of $T$ and $N$ are similar.

In the second scenario, we construct testing sets containing 10Mbps, 100Mbps Ethernet and WLAN observations. In particular, the fraction of 10Mbps-Ethernet observations, $\theta$, is 0.5 of all Ethernet observations. That is, the number of 10Mbps-Ethernet and 100Mbps-Ethernet observations are identical. When obtaining the Ethernet observation distribution, we vary $\theta$ in the range of 0.35 to 0.65. That is, the maximum error in estimating $\theta$ is 0.15. Fig. 5 plots the inference error of our classifier when $T = 400\mu s$ and $N = 1$ under various estimates of $\theta$. We observe that, under the correct estimate of $\theta$ and hence the correct observation distribution, the inference error for $\alpha$ is close to 0. Even when the estimation error of $\theta$ is 0.15, the inference error for $\alpha$ is bounded within $\pm 0.05$ for all values of $\alpha$. This indicates that our classifier is insensitive to imprecise observation distributions. We also observe that an underestimate of $\theta$ leads to a positive inference error (i.e., an overestimate of $\alpha$). This can be explained as follows. When underestimating $\theta$, some observations corresponding to 10Mbps Ethernet are classified as WLAN (due to its relatively high median inter-ACK time compared to that of 100Mbps Ethernet) and hence leads to an overestimate of $\alpha$. Similarly, an overestimate of $\theta$ leads to a negative inference error (i.e., an underestimate of $\alpha$).

### C. Empirical Evaluation

Our empirical evaluation is based on a trace collected from 10AM to 12PM on 05/10/2005. For the purpose of evaluation, we extract three sets of TCP flows from this trace, consisting solely of 10Mbps Ethernet, 100Mbps Ethernet or WLAN flows. The extraction is based on IP addresses, using the same method as that used to obtain the training sets. For each set of TCP flows, we vary $T$ to be 250 or $400\mu s$ and $N$ to be 1, 2 or 5 to obtain qualified TCP flows. The number of qualified TCP flows in each set is shown in Table II.

In the first set of evaluation, we construct testing sets each consisting of a single type of flows, i.e., 10Mbps Ethernet, 100Mbps Ethernet or WLAN. Our classifier obtains accurate estimate of $\alpha$. Fig. 6 plots the cumulative density function (CDF) of the beliefs for all testing sets. For the testing set containing only WLAN TCP flows, we use either the 10Mbps or 100Mbps Ethernet observation distribution as the Ethernet observation distribution. In both cases, Fig. 6 shows that the beliefs of the majority of the flows are very close to 1. For the testing set containing only 100Mbps Ethernet flows, the beliefs of the majority of the flows very close to 0. For the testing set containing only 100Mbps Ethernet flows, the beliefs of almost all flows are below 0.2, indicating that, with high degree of beliefs, these TCP flows are Ethernet flows.

In the second set of evaluation, we construct testing sets by mixing two types of flows: either 10Mbps Ethernet and WLAN or 100Mbps Ethernet and WLAN. The actual value of $\alpha$ and the MLE of $\alpha$, $\hat{\alpha}$, are shown in Table III for

TABLE II

NUMBER OF QUALIFIED TCP FLOWS USED IN THE EMPIRICAL EVALUATION

| $N$ | 10Mbps Ethernet | | 100Mbps Ethernet | | WLAN | |
|---|---|---|---|---|---|---|
| | $T = 250\mu s$ | $T = 400\mu s$ | $T = 250\mu s$ | $T = 400\mu s$ | $T = 250\mu s$ | $T = 400\mu s$ |
| 1 | 4488 | 5095 | 10107 | 10906 | 8223 | 9090 |
| 2 | 1744 | 2379 | 5953 | 6568 | 3263 | 4044 |
| 5 | 608 | 948 | 3263 | 3726 | 785 | 1219 |

TABLE III

MIXED TRACE VALIDATION (ETHERNET VS WLAN, 05/10/2005)

| | 10Mbps Ethernet vs WLAN | | | | 100Mbps Ethernet vs WLAN | | | |
|---|---|---|---|---|---|---|---|---|
| | $T = 250\mu s$ | | $T = 400\mu s$ | | $T = 250\mu s$ | | $T = 400\mu s$ | |
| $N$ | $\alpha$ | $\hat{\alpha}$ | $\alpha$ | $\hat{\alpha}$ | $\alpha$ | $\hat{\alpha}$ | $\alpha$ | $\hat{\alpha}$ |
| 1 | 0.65 | 0.66 | 0.64 | 0.65 | 0.45 | 0.44 | 0.45 | 0.44 |
| 2 | 0.65 | 0.68 | 0.63 | 0.66 | 0.34 | 0.34 | 0.38 | 0.36 |
| 5 | 0.56 | 0.59 | 0.56 | 0.60 | 0.19 | 0.19 | 0.25 | 0.25 |

various testing sets. We observe that, the MLE of $\alpha$ is very accurate in all cases (the maximum inference error is $0.04$). Fig. 7(a) compares the CDFs of the beliefs for Ethernet and WLAN flows in a testing set containing 100Mbps Ethernet and WLAN flows using $T = 400\mu s$ and $N = 1$. We observe that the beliefs for most WLAN flows are in the range of $[0.7, 1]$. For Ethernet flows, the beliefs of $80\%$ of flows are below $0.2$. Fig. 7(b) plots the result for a testing set ontaining 10Mbps Ethernet and WLAN flows. The beliefs of 10Mbps Ethernet and WLAN flows are not as separated as that in Fig. 7(a). We conjecture that this is due to the similar bandwidth of WLAN and 10Mbps Ethernet.

Last, we construct a testing set consiting all three types of flows: 10Mbps, 100Mbps Ethernet and WLAN. The Ethernet observation distribution is obtained as a weighted sum of 10Mbps and 100Mbps observation distributions (see Section V). For $T = 400\mu s$ and $N = 1$, the actual fraction of WLAN flows is $0.36$ and our inference is $0.35$.

## VII. INFERENCE RESULTS ON UMASS NETWORK

In this section, we infer the extent of wireless usage in the UMass campus network. We are interested in both the entire campus network and the residential network (i.e., student dorms). All TCP flows to the residential network are identified using the block of IP addresses reserved for the residential network. Our inference results are based on two
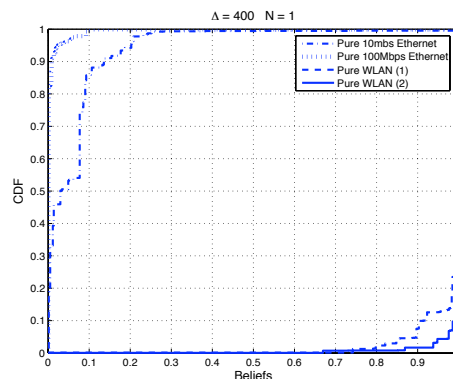


Fig. 6. The CDFs for the beliefs of the testing sets containing a single type of flow (10Mbps, 100Mbps or Ethernet), $T = 400\mu s$, $N = 1$.

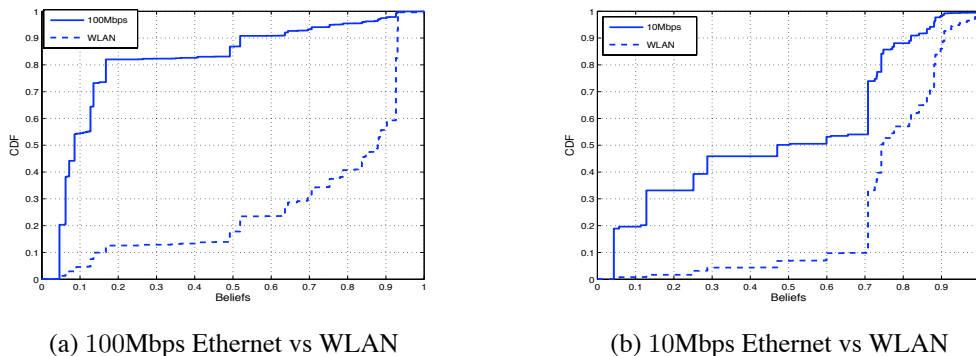(a) 100Mbps Ethernet vs WLAN



(b) 10Mbps Ethernet vs WLAN

Fig. 7.   The CDFs for the beliefs of the testing sets containing two types of flows, $T = 400\mu s$, $N = 1$.

TABLE IV

INFERRED EXTENT OF WIRELESS USAGE IN THE RESIDENTIAL AND THE ENTIRE CAMPUS NETWORK (05/10/2005, 10AM-12PM)

| | Residential | | | | All Campus | | | |
|---|---|---|---|---|---|---|---|---|
| | $T = 250\mu s$ | | $T = 400\mu s$ | | $T = 250\mu s$ | | $T = 400\mu s$ | |
| $N$ | flows | packets | flows | packets | flows | packets | flows | packets |
| 1 | 0.02 | 0.08 | 0.02 | 0.07 | 0.11 | 0.14 | 0.11 | 0.14 |
| 2 | 0.02 | 0.06 | 0.02 | 0.05 | 0.13 | 0.14 | 0.13 | 0.14 |
| 5 | 0.02 | 0.02 | 0.01 | 0.03 | 0.06 | 0.07 | 0.06 | 0.07 |

traces: collected between 11AM and 12PM on 04/04/2005, between 10AM and 12PM on 05/10/2005. The first trace contains a total of $3,309,480$ TCP flows, of which $64\%$ of the flows are from the residential network. The second trace contains a total of $6,250,306$ TCP flows, of which $60\%$ of the flows are from the residential network. In each trace, we set $T = 250\mu s$ or $400\mu s$ and $N = 1, 2,$ or $5$ when identifying qualified TCP flows. In the following, we mainly describe the results for the trace collected on $05/10/2005$; the results for the other trace are consistent.

For the trace on $05/10/2005$, when using $T = 400\mu s$ and $N = 1$, $10\%$ and $5\%$ of the TCP flows are qualified TCP flows (corresponding to $28\%$ and $19\%$ of the packets) in the entire campus network and the residential network respectively. Of all the qualified TCP flows, the CDFs of the flow size (in terms of packets) and the number of ACK-pairs are shown in Fig. 8. We observe that $90\%$ of the flows contain less than 10 ACK-pairs. The results for the trace on $04/04/2005$ are similar.

The UMass residential network uses 10Mbps half-duplex Ethernet. Therefore, for the inference in the residential network, we use the WLAN and 10Mbps Ethernet observation distributions. In the entire UMass campus network, of all Ethernet jacks, $95\%$ of the jacks use 10Mbps half-duplex Ethernet. Therefore, for the inference in the entire campus network, we use a weighted sum of 10Mbps and 100Mbps Ethernet observation distributions as the Ethernet observation distribution with the corresponding weight as $0.95$ and $0.05$ respectively (see Section V). The various observation distributions (10Mbps, 100Mbps Ethernet and WLAN) are the same as those used in Section VI.

In the following, the extent of wireless usage is measured in both the number of flows and the number of packets. Let $\gamma$ denote the extent of wireless usage in terms of packets. Then $\gamma$ can be obtained as follows. Suppose there are $n$ qualified TCP flows. Let $S_i$ denote the number of packets in the $i$-th flow, $i = 1, \cdots, n$. Then

$$\gamma = \frac{\sum_{i=1}^{n} \beta_i S_i}{\sum_{i=1}^{n} S_i}$$

We next describe the results on the extent of wireless usage and the beliefs of the TCP flows.
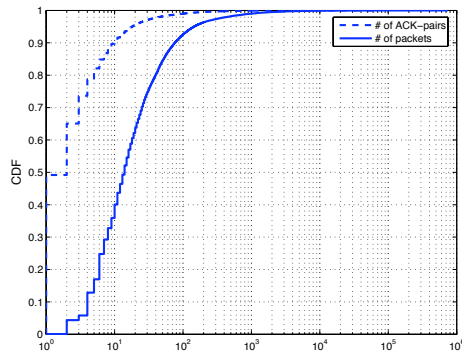
Fig. 8. The CDF for the number of ACK-pairs and the number of packets for the trace collected on $05/10/2005$.

## A. *Extent of wireless usage*

The classification results for the trace collected on $05/10/2005$ are shown in Table IV. The fraction of wireless flows in the residential network is around $2\%$. This implies the existence of wireless usage through private wireless routers and access points, since there is no official wireless coverage in the residential network. In the entire campus, the fraction of WLAN flows is $11\%$ for $T = 400\mu s$ and $N = 1$. When using the 10Mbps-Ethernet observation distribution as the Ethernet distribution, we obtain a lower estimate of the fraction of WLAN flows ($6\%$) in the entire campus (see Section VI). The extent of wireless usage in terms of packets is higher than that in terms of flows. This is because, as we shall see, WLAN flows have relatively lower ratio of ACK-pairs. For the trace on $04/04/2005$, the extent of wireless usage is $14\%$ in the entire campus and $3\%$ in the residential network for $T = 400\mu s$ and $N = 1$.

We also infer the extent of wireless usage for Computer Science Department, where all the Ethernet connections are 100Mbps. For the trace on $05/05/2005$, the fraction of wireless flows is $20\%$ for both $T = 250\mu s$ and $T = 400\mu s$ when $N = 1$. For the trace on $04/04/2005$, the fraction of wireless flows are $25\%$ and $27\%$ for $T = 250\mu s$ and $T = 400\mu s$ respectively when $N = 1$.

## B. *Beliefs of the TCP flows*

For the trace on $05/10/2005$, the CDF for the beliefs of all qualified TCP flows is shown in Fig. 9. We find that, of all the $484768$ qualified flows, $87\%$ of the TCP flows have beliefs either higher than $0.8$ or lower than $0.2$. This indicates that, for majority of the flows, our classifier provides a relatively strong belief as to whether the flows traverse WLAN or Ethernet links. In the following, we refer to a flow with belief higher than $0.8$ as a *WLAN-likely flow*; a flow with belief lower than $0.2$ as a *Ethernet-likely flow*. We now look at the characteristics of WLAN-likely and Ethernet-likely flows.

*1) The ACK-pair ratio:* We define the ACK-pair ratio of a flow as the number of ACK-pairs divided by the total number of packets in the flow. Intuitively, a WLAN flow has lower ACK-pair ratio than an Ethernet flow. This is because, as shown in Section III, the inter-ACK time of a WLAN flow tends to be larger than that of an Ethernet flow. Due to TCP's self-clocking, the dispersion between ACKs leads to dispersion of data packets and less number of ACK-pairs in WLAN flows. This is confirmed by the results from our training sets (for 10Mbps, 100Mbps Ethernet and WLAN). For each training set, we sort the flows in the decreasing order of flow length (in terms of packets). For the top $x\%$ of the flows (i.e., the longest $x\%$ of flows), we calcuate the average ACK-pair ratio of these flows. Fig. 10 plots the result for $x$ in the range of 10 to 100. The 95-th confidence intervals are very tight and hence omitted. We observe that WLAN flows have lower ACK-pair ratio than Ethernet flows. To further confirm this, we carry out controlled
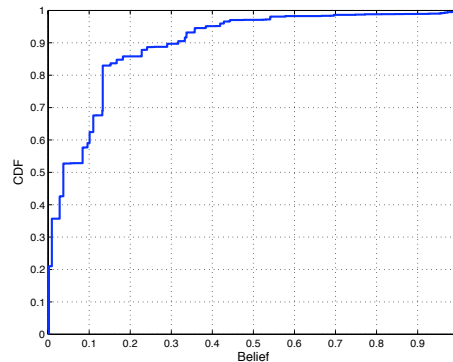
Fig. 9. The CDF for the beliefs of the qualified TCP flows for the trace collected on 05/10/2005.
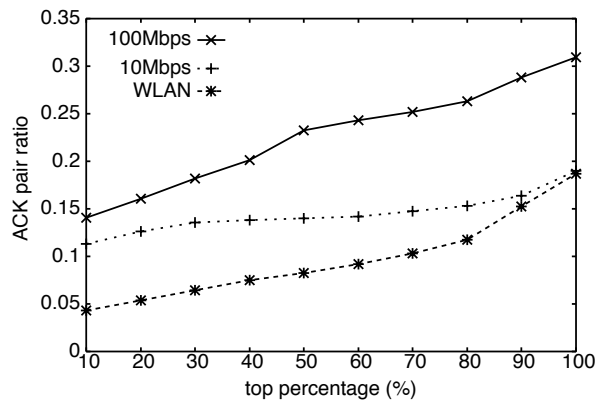


Fig. 10. Comparison of ACK-pair ratio in the training sets for 100Mbps-Ethernet, 10Mbps-Ethernet and WLAN.

experiments consisting of downloads of a 2Mbyte file using either Ethernet or WLAN access link and traced at the UMass gateway. The mean ACK-pair ratio is 0.19 and 0.02 over 20 and 15 experiments using Ethernet and WLAN respectively. This again confirms that WLAN flows have relatively lower ACK-pair ratio than Ethernet flows.

For the trace on 05/10/2005, we sort the flows in the decreasing order of flow length (in terms of packets). Fig. 11 plots the average ACK-pair ratio for the top $x\%$ of the flows for $x$ from 5 to 100. The confidence intervals are again not significant and hence omitted. We observe that the ACK-pair ratio is relatively stable for Ethernet-likely flows for all values of $x$. For WLAN-likely flows, the ACK-pair ratio is close to 0 for long flows, higher for short flows and lower than that for Ethernet-likely flows for all values of $x$. The lower ACK-pair ratio for WLAN-likely flows compared to Ethernet-likely flows is consistent with our observation that WLAN flows have relatively lower ACK-pair ratios.

We also calculate the correlation coefficient between the belief and the ACK-pair ratio. For the trace on 05/10/05, the correlation coefficient is $-0.15$ for $T = 400\mu s$ and $N = 2$, indicating a significant negative correlation between the belief and the ACK-pair ratio. That is consistent with our observation that a WLAN connection (i.e., with a higher belief) has lower ACK-pair ratio. For the top $5\%$ of the longest flows, the correlation coefficient is $-0.26$.

*2) Sources of WLAN flows:* We further validate the result of our inference based on known IP addresses. For the trace on 05/10/2005, of all the flows with beliefs higher than 0.99, $5.6\%$ of the flows are from the 802.11 public WLAN, confirming that the classification for this group of IP addresses is correct; we also find TCP flows from the residential network, confirming the usage of private wireless routers in the student dorms. Of all the flows with beliefs lower than 0.01, only $0.08\%$ of the flows are from the 802.11 public WLAN, indicating a very low inference error.
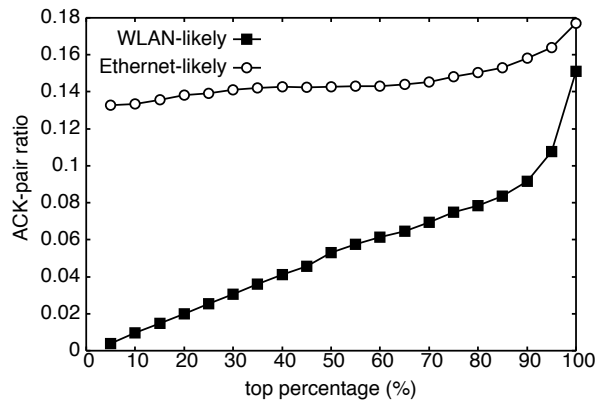
Fig. 11. Comparison of the ACK-pair ratio between Ethernet-likely and WLAN-likely flows for the trace collected on 05/10/2005.

## VIII. CONCLUSION AND FUTURE WORK

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 classifier computes, for $n$ TCP flows, the fraction of wireless TCP flows, $\alpha$; and the belief that the $i$th TCP flow traverses a WLAN inside the network, $\beta_i$, $i = 1, 2, \ldots, n$. The core of this classifier is an iterative Bayesian inference algorithm that we developed to obtain the maximum likelihood estimate (MLE) of $\alpha$ and $\{\beta_i\}$. We prove that our iterative inference algorithm converges to the unique MLE of $\alpha$ and $\{\beta_i\}$. Our algorithm can handle any general two-class classification problems given the marginal distributions of these two classes. Numerical and experimental evaluations demonstrate that our classifier obtains accurate results and is insensitive to imprecise marginal distributions. We apply the classifier to various traces collected at the edge of UMass campus network and infer that between 11-14% of all TCP flows going in UMass campus traverse a 802.11 wireless link within the campus. We also detect wireless usage (through the use of private routers and access points) in areas not covered by the official wireless infrastructure.

As future work, we are exploring the application our techniques to measurements collected passively at a server, in order to infer the types of client connections. In our measurements, 20% to 30% of the collected packets are used for the inference. We plan to desgin schemes that use information more efficiently.

## IX. ACKNOWLEDGMENT

## REFERENCES

[1] L. Cheng and I. Marsic, "Fuzzy reasoning for wireless awareness," *International Journal of Wireless Information Networks*, vol. 8, no. 1, 2001.

[2] W. Wei, B. Wang, C. Zhang, J. Kurose, and D. Towsley, "Classification of access network types: Ethernet, wireless LAN, ADSL, cable modem or dialup?," in *Proceedings of IEEE Infocom 2005*, March 2005.

[3] V. Padmanabhan, L. Qiu, and H. Wang, "Server-based inference of internet link lossiness," 2003.

[4] D. Katabi, I. Bazzi, and X. Yang, "A passive approach for detecting shared bottlencks," in *IEEE International Conference on Computer Communications and Networks*, 2001.

[5] S. Katti, D. Katabi, C. Blake, E. Kohler, and J. Strauss, "Multiq: Automated detection of multiple bottleneck capacities along a path," in *Proceedings of the 2004 ACM Sigcomm Internet Measurment Conference*, 2004.

[6]  S. Seshan, M. Stemm, and R. H. Katz, "SPAND: Shared passive network performance discovery," in *USENIX Symposium on Internet Technologies and Systems*, 1997.

[7]  D. Rubenstein, J. Kurose, and D. Towsley, "Detecting shared congestion of flows via end-to-end measurement," in *Proc. ACM SIGMETRICS*, June 2000.

[8]  R. Carter and M. Crovella, "Measuring bottleneck link speed in packet-switched networks," *Performance evaluation*, pp. 297–318, 1996.

[9]  C. Dovrolis, P.Ramanathan, and D. Moore, "What do packet dispersion techniques measure?," in *Proc. IEEE INFOCOM*, 2001.

[10]  V. Jacobson, "pathchar - a tool to infer characteristics of internet paths." ftp://ftp.ee.lbl.gov/pathchar, April 1997.

[11]  A. B. Downey, "Using pathchar to estimate Internet link characteristics," in *Measurement and Modeling of Computer Systems*, pp. 222–223, 1999.

[12]  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," in *Proceedings of the 8th IEEE Global Internet Symposium*, March 2005.

[13]  D. Kotz and K. Essien, "Analysis of a campus-wide wireless network," *Mobile Networks and Applications*, 2003.

[14]  A. Balachandran, G. Voelker, P. Bahl, and P. Rangan, "Characterizing user behavior and network performance in a public wireless LAN," 2002.

[15]  P. Sarolahti and A. Kuznetsov, "Congestion control in linux TCP," in *Proc. of USENIX'02*, June 2002.

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

[17]  K. Thompson, G. Miller, and R. Wilder, "Wide-area Internet traffic patterns and characteristics," *IEEE Network*, vol. 11, pp. 10–23, Nov./Dec. 1997.

[18]  "Packet trace analysis."
http://ipmon.sprintlabs.com/packstat/packetoverview.php.

[19]  K. Papagiannaki, S. Moon, C. Fraleigh, P. Thiran, and C. Diot, "Measurement and analysis of single-hop delay on an IP backbone network," *IEEE JSAC Special Issue on Internet and WWW Measurement, Mapping, and Modeling*, vol. 21, no. 6, 2003.

[20]  S. Garg, M. Kappes, and A. S. Krishnakumar, "On the effect of contention-window sizes in IEEE 802.11b networks," Tech. Rep. ALR-2002-024, Avaya Labs Research, 2002.

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

[22]  A. Dempster, L. N.M., and R. D.B., "Maximum-likelihood from incomplete data via the EM algorithm," *J. Royal Statistical Soc. Ser. B(methodological)*, vol. 39, pp. 1–38, 1977.

[23]  G. Casella and R. L. Berger, *Statistical Inference*, ch. 7, p. 320. Duxbury Thomson Learning, 2002.

[24]  "http://www.endace.com."