Classification of Access Network Types: Ethernet, Wireless LAN, ADSL, Cable Modem or Dialup?

Wei Wei, Bing Wang, Chun Zhang, Jim Kurose, Don Towsley
Department of Computer Science
University of Massachusetts, Amherst, MA 01003

***Abstract*—Ethernet, wireless LAN, ADSL, cable modem and dialup are common access networks, but have dramatically different characteristics. Fast and accurate classification of access network type can improve protocol or application performance significantly. In this paper, we propose a simple and efficient end-end scheme to classify the type of an access network using packet pairs. Our scheme is based on the intrinsic characteristics of the various access networks and utilizes the median and entropy of the packet pair inter-arrival times. Extensive experiments show that our scheme obtains accurate classification results in a very short time (10 to 100 seconds).**

I. Introduction

Access networks, consisting of the links connecting end systems to edge routers, are of dramatically different types. Dialup, ADSL (asymmetric digital subscriber line) and HFC (hybrid fiber coaxial cable) are three prevalent types in residential area. Ethernet and switched Ethernet are common LAN technologies in business and educational institutions. In addition, wireless LANs (WLAN) using the IEEE 802.11b standard are becoming more and more popular in the office environment. We refer to the above access networks as dialup, ADSL, cable modem, Ethernet and WLAN connections. The various types of connections have dramatically different characteristics in terms of physical media, capacity, and symmetry of the upload and download bandwidths. Furthermore, user behavior can be very different when using different access networks. For instance, studies show that wireless sessions are usually short (within several to tens of minutes) [1], [2]. Dialup connections may also have short durations, while cable modem and ADSL users tend to be on line for a longer time. Ethernet connections might be the most stable among all types of connections. In this paper, we do not differentiate between Ethernet and switched Ethernet and refer to them loosely as a high-bandwidth wired Ethernet connection or simply a wired connection.

Classifying connection types using end-end approaches is very useful in a wide range of scenarios for network protocols and applications. For instance, there is a large amount of literature on improving the performance of TCP when the last hop is a wireless link (e.g. [3], [4]) or cable connection (e.g. [5]), assuming that it is *known* that the last hop is a wireless link or cable connection. In peer-to-peer applications, given a group of neighbors, a peer may well choose a neighbor using an Ethernet connection over those using other connection types. Similarly, when constructing an application-level overlay [6], it is often desirable to choose overlay nodes with Ethernet connections over those with other connection types. In application layer multicast [7], it is desirable to select nodes at higher levels of the multicast tree to have wired high-bandwidth connections. Being able to determine the type of access network can also be useful for applications with bandwidth and delay requirements. For example, for streaming multimedia applications, a server may want to adapt the rate of the media to the access network bandwidth.

However, accurate classification of connection types is not an easy task. It is often not possible for the end system to reliably report the connection type for the following reasons. First, the end system may not have knowledge of the connection type. A laptop connected to a cable or ADSL modem using a wireless connection would report WLAN as its connection type instead of cable or ADSL. Also, an end system may have an incentive to conceal its connection type, and a compromised machine may also report its connection type inaccurately to degrade the performance of an overlay network.

In this paper, we are interested in end-end approaches for determining the access network type. We propose a simple and efficient end-end scheme to classify the type of an access network using packet pairs (a packet pair contains two back-to-back packets). Our algorithm is based on the intrinsic characteristics of the various connection types and roughly works as follows. If node A needs to determine the connection type of node B, A asks B to send a sequence of packet pairs to A. A determines B's connection type based on median and entropy of the inter-arrival times of the packet pairs from B (see Section III). Extensive experiments show that our scheme obtains accurate classification results in a very short time (10 to 100 seconds).

Packet-pair or packet-train approaches have been previously used in the literature to determine the capacity or available bandwidth of an end-end path [8], [9]. Our work differs from the above in that we use packet pairs to determine the access network type based on characteristics of the different access network types. It turns out that the median and entropy of the inter-arrival time of packet pairs demonstrates very different characteristics for different access network types. In [10], the authors distinguish congestion losses from wireless losses using packet inter-arrival times at the receiver *assuming* that the last hop is wireless and the wireless link is the only bottleneck. Our scheme can differentiate access network types in both lossy and un-lossy environments. Furthermore, we do not assume the wireless link to be a bottleneck. The only work that we are aware of that considers differentiating connection types is [11]. However, [11] only differentiates between wireless and wired connections and assumes very low bandwidth and lossy wireless links. The method in [11] is based on the observation that the high loss rate of wireless links leads to a wider RTT spread. Our work, in contrast, is based on the fundamental characteristics of the various access networks and therefore provides accurate classification regardless of the loss rate at the access network.

The rest of the paper is organized as follows. In Section II, we provide some background for the access networks considered in this paper. In Section III, we present our scheme to classify the connection types. Section IV presents an analytical foundation for our approach. Section V describes the experimental results. Finally, Section VI concludes the paper and describes future work.

II. Background

In this section, we provide background for all the access networks considered in this paper. Our goal here is to describe those mechanisms in the access protocol that will allow us to distinguish one type of access network from another. We first describe the IEEE 802.11 standard used in WLAN and its main difference from Ethernet. We then describe the Data Over Cable Service Interface Specification (DOCSIS) in cable networks. Last, we briefly describe Ethernet, ADSL and dialup connections.

A. IEEE 802.11 standard

IEEE 802.11 standard [12] defines the physical layer and media access control (MAC) layer for WLAN. In IEEE 802.11 MAC layer, CSMA/CA (carrier sense multiple access with collision avoidance) is implemented in all wireless stations and base stations in order to coordinate the access of the shared media by multiple stations. A wireless station accesses the channel using a *basic access method* or an *optional four-way handshaking access method*. When the packet size is smaller than an *RTS Threshold*, the basic access method is used in order to improve efficiency. Since the packet pairs we use are of very small packet size, they are sent using the basic access method, and thus we only describe the basic access method here. When using the basic access method, if a station has a packet to send, it may transmit when the media is free for greater than or equal to a DIFS (Distributed Interframe Space) time. If the media is busy, a station sets a random backoff timer following a binary exponential backoff procedure. The wireless station chooses the initial backoff timer value for a packet to be a random number uniformly distributed in the range of 0 and the contention window, CW. The contention window is set to CW_{min} for each new data transmission and doubles each time a transmission is unsuccessful until it reaches the maximum contention window, CW_{max}. The backoff timer decreases by one when the media is idle for a slot time and is frozen when the channel is sensed busy. When the backoff timer reaches zero, the station sends the data frame.

In 802.11, unlike in Ethernet, the destination needs to send an explicit ACK to the sender since a wireless sender cannot determine whether or not its transmission has been successful. Since IEEE 802.11b is used most widely in a WLAN, we focus on IEEE 802.11b in this paper. Table I [13] summarizes the transmission overhead per packet in 11 Mbps 802.11b [13], [14]. The slot time is 20 μs and the minimum Contention Window, CW_{min}, is 31. Therefore, assuming no collisions, the average backoff duration is 310 μs, leading to an average transmission overhead of around 810 μs per packet. Some wireless hosts and base stations are 802.11b compatible and support bandwidth up to 22 Mbps. When using a bandwidth of 22 Mbps, the total average transmission overhead per packet is halved to around 400 μs, assuming no collisions.

Last, we emphasize that, a 802.11b wireless station must wait for a random backoff time after a successful transmission, even if no other station is transmitting, in order to avoid channel capture. This backoff mechanism is not used only when the station decides to transmit a new packet and the medium has been free for more than DIFS. This implies that random backoff will occur between two back-to-back packets. Hence, when two back-to-back packets are sent on a perfect wireless channel, the inter-departure time of the packet pair is uniformly distributed between 500 μs and 1130 μs, with the median of 810 μs. We will take advantage of the distribution and median of the packet-pair inter-departure time in our classification scheme.

TABLE I

BREAKDOWN OF TRANSMISSION OVERHEAD PER FRAME IN 802.11B (USING 11 MBPS).

Overhead Type	Time (μs)	Comments
PHY	192	Includes PLCP header and the physical layer preamble
MAC	24.7	Time to transmit 34 bytes of MAC header at 11 Mbps
IP & UDP	20.4	Transmission time for 28 bytes of IP and UDP headers
ACK	202.2	ACK transmission time including associated PHY overhead
SIFS	10	After frame is received but before ACK is sent
DIFS	50	Minimum idle time to be observed before back-off starts
Back-off	310	Average value of back-off
Total	809.3	

B. DOCSIS

In cable networks, the downstream channel from the Cable Modem Termination System (CTMS) at the head-end to a Cable Modem (CM) at home is a broadcast channel shared by many different homes. The upstream channel from the CMs to the CTMS is a random access channel. DOCSIS specifies the MAC and physical layer protocols for cable networks and is the *de facto* standard in the cable industry. We briefly describe the specification of the upstream channel in DOCSIS, with a focus on contention resolution. The upstream channel is divided into units of $6.25\ \mu s$, referred to as *mini-slots*. The CTMS periodically broadcasts a downstream management message, referred to as MAP, to all the CMs. Each MAP contains timing information regarding *request mini-slots* and *data mini-slots*. When a CM has data to send, it must first send a request message using request mini-slots to the CMTS and wait for a *data grant*. After receiving a data grant message from the CMTS, a CM uses the assigned data mini-slots to transmit data on the upstream channel. The CMs contend for use of request mini-slots following a contention resolution method based on binary exponential backoff [15]. The CTMS assigns the size of the initial backoff window and the maximum backoff window in the MAP. When a CM needs to send a request message, it randomly chooses a backoff value in its window. This backoff value indicates the number of contention mini-slots that the CM must wait before it can transmit the request. If a collision occurs, which is indicated by the absence of a data grant or a data pending indication in the next MAP from the CTMS to the CM, the CM doubles its window size, until the maximum window size is reached. A request message is discarded by the CM after 16 retries.

The contention occurring on the upstream channel of Cable modem provides us the opportunity to distinguish cable from ADSL (which provides a dedicated collision free connection), even though they have similar bandwidth. Because of contention, the inter-departure time of a cable modem has more randomness than that of an ADSL connection, leading to larger entropy in the inter-arrival time of packet pairs, an intuition confirmed by our experimental results.

C. Ethernet, ADSL and dialup

Switched Ethernet has dedicated access and high bandwidth. Although an Ethernet connection uses shared media, the randomness caused by the shared media in Ethernet is negligible compared to a WLAN and cable modem because of its high bandwidth and ability to detect collisions. ADSL has dedicated access with low bandwidth. Dialup has dedicated access and very low bandwidth.

III. CLASSIFICATION SCHEME

In this section, we describe our classification scheme for access network types. Roughly, the scheme operates as follows. When node A needs to determine the connection type of node B, A asks B to send a sequence of packet pairs to A. For each packet pair from B, A records the inter-arrival time of the two packets in the pair. Then A determines B's connection type based on the median and entropy of the sequence of inter-arrival times. In the rest of the paper, we refer to B as the *sender* and A as the *receiver*. We assume that the receiver is well-connected (i.e., it has a wired connection with reasonable bandwidth). This is reasonable in the settings where a server classifies client connection types. In a peer-to-peer or overlay network, we can assume that one or several well-connected nodes are in charge of determining the connection types of the other nodes.

Our scheme is motivated by the fact that the inter-arrival time at the receiver between two back-to-back packets will be different dramatically depending on the connecting type. As shown in Section II, in a WLAN, two back-to-back packets from a wireless station are separated by a random backoff duration, even when the channel has no contention or transmission errors. In the upstream channel of cable network, two back-to-back packets from a CM are also separated by a random backoff duration due to the backoff mechanism for resolving contention among the CMs. Ethernet also exploits a binary

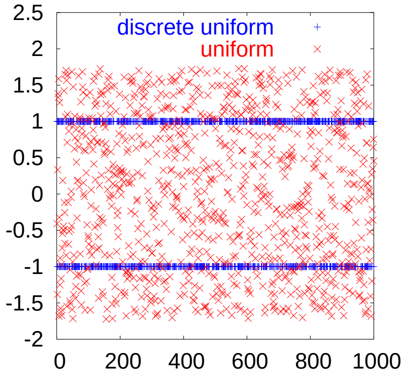


Fig. 1. Two distributions with the same mean, variance but very different entropy.

backoff mechanism, but the interval between two back-to-back packets from an end system in a wired Ethernet is much shorter than in a WLAN and cable modem connections due to the wired, high bandwidth characteristics of the Ethernet (10 Mbps, 100 Mbps or higher). In all of the other types of connections (i.e., switched Ethernet, ADSL and dialup), no random backoff is introduced between two back-to-back packets. We therefore use packet pairs to track the different amount of randomness inherently induced by the access protocols used in different access networks and further exploit this randomness to distinguish the connection types. We measure this randomness using entropy.

The reason to use entropy rather than variance is that entropy is a much better metric to capture the randomness of a random variable than variance. A simple example illustrates this advantage. Suppose V_1 and V_2 are two random variables and suppose we generate 1000 instances of V_1 and V_2. As shown in Fig. 1, V_1 is a discrete random variable taking values on $+1$ and -1 with equal probability, and V_2 is a continuous random variable follows a uniform distribution on the interval of $[-\sqrt{3}, \sqrt{3}]$. The variance of V_1 and V_2 are both $1/12$. These two random variables therefore cannot be distinguished using variance. However, they can be easily distinguished using entropy. For a discrete random variable X taking value in $\mathcal{X}$ and having a probability mass function of $p(x) = P(X = x), x \in \mathcal{X}$, the entropy of X, $H(X)$, is defined as

$$H(X) = -\sum_{x\in\mathcal{X}} p(x) \log p(x).$$

When the logarithm is to the base of 2, $H(X)$ is denoted as $H_2(X)$ and expressed in bits. To differentiate V_1 and V_2 using entropy, we discretize the interval $[-\sqrt{3}, \sqrt{3}]$ into 1024 bins. Then $H_2(V_1) = 1$ bit and $H_2(V_2) = \log_2(1024) = 10$ bits. Intuitively, entropy performs better because entropy captures the randomness of a random variable over entire $\mathcal{X}$ while variance only describes variations of a random variable around its mean.

The sender sends 500 packet pairs to the receiver;
The receiver obtains the sequence of the inter-arrival times of the packet pairs as $\{I_i\}_{i=1}^n$;

Rule 1:
if ($\xi^n_{.5}(I) \leq 600\ \mu s$ **and** $H_2^{300}(I) \leq 2.0$ bits
and $H_2^{900}(I) \leq 0.6$ bit)
then the sender uses Ethernet (high-bandwidth wired);

Rule 2:
else if ($600\ \mu s < \xi^n_{.5}(I) \leq 2$ ms **and** $H_2^{300}(I) \geq 0.8$ bit)
then the sender uses WLAN;

else the sender uses ADSL, cable or dialup connection
(low-bandwidth connection).

Fig. 2. Classification scheme: the receiver classifies the connection type of the sender based on a sequence of packet pairs from the sender.

In addition to entropy, our scheme also utilizes the median of the packet-pair inter-arrival times. We use the median instead of the mean to reduce the effect of outliers in the network measurements [16]. The median captures the capacities of the connections and intrinsic connection characteristics other than randomness. For instance, a low bandwidth connection tends to result in a larger median value than a high bandwidth connection. As another example, the slower carrier sensing and explicit ACK in WLAN leads to a much higher median value of the inter-arrival time of a packet-pair than in Ethernet. We now describe our classification scheme in detail.

A. Classification scheme

For ease of exposition, we first define our notation. Let the random variable I denote the inter-arrival time of a packet pair from the sender to the receiver. Let $\xi_{.5}(I)$ and $H(I)$ denote the (population) median and entropy of I respectively. In practice, we obtain estimates of $\xi_{.5}(I)$ and $H(I)$ through a sequence of samples. Let I_i denote the inter-arrival time of the two packets in the ith packet pair from the sender to the receiver. Suppose the receiver receives n packet pairs. Then $\{I_i\}_{i=1}^n$ represents the sequence of inter-arrival times of the packet pairs. Let $\xi^n_{.5}(I)$ denote the sample median of $\{I_i\}_{i=1}^n$. We estimate $H(I)$ from the samples at the time scale of $300\ \mu s$ and $900\ \mu s$. That is, we discretize $\{I_i\}_{i=1}^n$ using a bin size of $300\ \mu s$ or $900\ \mu s$ and calculate the entropy of the discretized values. For convenience, we use the logarithm of base 2 and denote the entropies obtained at the various time scales as $H_2^{300}(I)$ and $H_2^{900}(I)$ respectively.

Our classification scheme is summarized in Fig. 2. First, the sender sends 500 packet pairs to the receiver (our experiments in Section V demonstrate that 500 packet pairs are sufficient to classify the connections). The receiver records the sequence of the inter-arrival times of the packet pairs, $\{I_i\}_{i=1}^{n}$, where $n \leq 500$ since packets might be lost on the path from the sender to the receiver. *Rule 1* states the criteria that differentiates Ethernet and non-Ethernet connections. That is, if $\xi_{.5}^{n}(I) \leq 600$ μs, $H_2^{300}(I) \leq 2.0$ bits and $H_2^{900}(I) \leq 0.6$ bit, then the scheme classifies the sender's connection as Ethernet. Otherwise, the connection type is non-Ethernet. This rule is based mainly on our analytical results in Section IV. We use a precision of 0.1 for the entropies to reduce the effect of outliers. This is because, even in the presence of one outlier, the resultant entropy can be off by approximately 0.02 [17]. In Section V, we show that all the three conditions in *Rule 1* (i.e., median, entropy at the time scales of 300 μs and 900 μs) are required to determine that a connection is Ethernet.

Rule 2 states that if $\xi_{.5}^{n}(I)$ is between 600 μs and 2 ms and $H_2^{300}(I) \geq 0.8$ bit, then the sender's connection is a WLAN; connections not satisfying the criteria of Ethernet and WLAN are cable, ADSL or dialup connections. In this rule, 600 μs is the upper bound of $\xi_{.5}^{n}(I)$ for Ethernet and the 2 ms is from empirical results. Our empirical results show that it is difficult to obtain a clear cut distinction among the three low-bandwidth connections using median and entropy. The entropy of a cable connection can be much larger than that of ADSL due to the contention in the cable networks. However, in general, it is difficult to differentiate cable and ADSL completely. This is because the upstream cable and ADSL connection are of similar bandwidth. Moreover, the entropies of cable and ADSL connections are similar when there is little sharing or contention among the cable modems. Dialup may have a much larger median value than cable and ADSL due to its low bandwidth. Roughly, we determine the connection to be dialup when $\xi_{.5}^{n}(I) \geq 10$ ms. However, we also observe very low values of $\xi_{.5}^{n}(I)$ for some dialup connections, which is likely due to traffic shaping, as observed in [18].

IV. An Analytical Basis for Classification

In this section, we present analytical models that form the foundation for our classification's differentiation between Ethernet and non-Ethernet connections. Even though the models are idealized, they provide insight into defining appropriate classification rules that work extremely well in practice. In the following, Section IV-A describes the assumptions for the analysis. Section IV-B states the results on the median and entropy of the packet-pair inter-arrival times for Ethernet connections, which leads to *Rule 1* in the classification scheme. Section IV-C presents results for non-Ethernet connections, which partially justifies *Rule 2* in the classification scheme. Finally, Section IV-D use *ns* simulation to validate the assumptions used in the models.

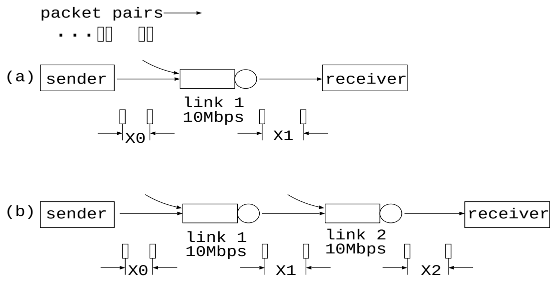


Fig. 3. Settings for the analysis: one and two 10 Mbps links connect the sender and the receiver in Setting (a) and (b) respectively. $X_k = \lceil \Delta_k/(300\mu s) \rceil, k = 0, 1, 2.$

A. Assumptions

Consider a sender sending packet pairs to a receiver. We assume that on the route from the sender to the receiver, there are at most two intermediate low-bandwidth links with a bandwidth of B Mbps. All the other intermediate links are assumed to have a capacity much higher than B Mbps (based on the fact that backbone is usually very well provisioned). Hence their effect on the inter-arrival time of a packet pair at the destination is negligible compared to that of the intermediate low-bandwidth links. We therefore ignore these high bandwidth links and consider only the two settings illustrated in Fig. 3.

In Setting (a), the sender is connected to the receiver by one link of bandwidth B Mbps. In Setting (b), the sender is connected to the receiver by two links, each with a bandwidth of B Mbps. In both settings, we refer to the access link as link L_0, the first link as link L_1. In Setting (b), we refer to the second link as link L_2. Assume that packet arrivals to link L_1 and L_2 form a Poisson process, and that packet arrivals at link L_1 and L_2 are independent. Links L_1 and L_2 are modeled as $M/D/1$ queues. Let ρ_1 and ρ_2 represent the utilization of link L_1 and L_2 respectively, $0 \leq \rho_1, \rho_2 \leq 1$. Let Δ_0 denote the inter-departure time of a packet pair at the access link L_0, which then becomes the inter-arrival time at link L_1. Let Δ_1 and Δ_2 denote the inter-departure time of the packet pair at link L_1 and L_2 respectively. For ease of analysis, we assume that all the packets are of equal size of S bytes. We later show using *ns* simulation that this assumption does not have a critical effect on the results.

To make the discussion concrete, we assume $B = 10$ Mbps. Measurement studies show that the average packet size in the Internet is between 300 and 400 bytes [19], [20]. We use $S = 375$ bytes for ease of computation. Let μ denote the processing rate of link L_1 and L_2 in packets per second. Then $1/\mu$ denotes the processing time at link L_1 and L_2. For a packet of 375 bytes, $1/\mu = 300$ μs. For ease of exposition, we will define a time unit of length $1/\mu = 300\ \mu s$, and refer to this as the *packet transmission time*. We discretize the inter-departure time of a packet pair at link L_k using the packet transmission time as the time unit, and denote the discretized inter-departure time as X_k, $k = 0, 1, 2$. That is, $X_k = \lceil \Delta_k \mu \rceil = \lceil \Delta_k/(300 \mu s) \rceil$. We calculate entropy using time scales of 300 μs and 900 μs. Let $H_2^{300}(\Delta_k)$ and $H_2^{900}(\Delta_k)$ denote the entropy of Δ_k using the respective time scale and base 2 logarithm, $k = 0, 1, 2$. It is easy to see that $H_2^{300}(\Delta_k) = H_2(X_k)$. Let $\xi^n_{.5}(\Delta_k)$ denote the median of n samples of inter-departure times at link L_k, $k = 1, 2$. In our experiments, n is in the range of 400 to 500.

We are interested in the median and entropy of packet pair inter-arrival times at the receiver, that is, $H_2(I)$ and $\xi^n_{.5}(I)$. In Setting (a), since the inter-arrival time of a packet pair at the receiver is the same as the inter-departure time of the packet pair at link L_1, we have $I = \Delta_1$. Similarly, in Setting (b), we have $I = \Delta_2$. We first present two lemmas on the distribution of the packet-pair inter-departure time at an $M/D/1$ queue. In the interest of space, derivations and proofs are omitted and can be found in [17].

Lemma 1: Consider an $M/D/1$ queue with processing rate μ and utilization ρ. Suppose that the inter-arrival time of a packet pair at the queue is $\Delta_a = X_a/\mu$, $X_a > 0$. Let Δ_d denote the inter-departure time of the packet pair after the queue and $\Delta_d = X_d/\mu$, $X_d > 0$. Then if $x_a \leq 1$, we have

$$P(X_d = x_d \mid X_a = x_a, \rho) = \frac{e^{-x_a \rho}(x_a \rho)^{x_d - 1}}{(x_d - 1)!}$$

where $x_d = 1, 2, \ldots$

Lemma 2: Under the conditions of Lemma 1, if $x_a > 1$ and $\rho = 1$, we have

$$P(X_d = x_d \mid X_a = x_a, \rho = 1) = \frac{e^{-x_a} x_a^{x_d - 1}}{(x_d - 1)!}$$

where $x_d = 1, 2, \ldots$

B. Ethernet connections

We now state two theorems concerning the median and entropy of packet-pair inter-arrival times for a 100 Mbps Ethernet connection. These two theorems form the foundation of *Rule 1* in our classification scheme to differentiate between Ethernet and non-Ethernet connections.

Theorem 1: **(Median for Ethernet connection)** In Setting (a) and (b), for sample size n between 400 and 500, we have $P(\xi^n_{.5}(I) \leq 600\ \mu s) \approx 1$. That is, the median inter-arrival time of the packet pairs at the receiver is below 600 μs with probability close to 1.

Proof: Under the assumption that the bandwidth of the sender is 100 Mbps and the first low-bandwidth intermediate link is 10 Mbps, we have $\Delta_0 = 30 \mu s = 0.1/\mu$. That is, $X_0 = 0.1$. By Lemma 1,

$$\begin{aligned} P(X_1 \leq 2 \mid X_0 = 0.1, \rho_1) &= e^{-0.1\rho_1} + 0.1\rho_1 e^{-0.1\rho_1} \\ &\geq e^{-0.1} + 0.1 e^{-0.1} = 0.9953 \end{aligned}$$

Since $I = \Delta_1$ in Setting (a), we prove the above theorem by showing that $P(\xi^n_{.5}(\Delta_1) \leq 600\ \mu s) \approx 1$. Let $p = P(X_1 \leq 2 \mid \rho_1) = P(\Delta_1 \leq 600\ \mu s)$. Then

$$P(\xi^n_{.5}(\Delta_1) \leq 600\ \mu s) = \sum_{i=n/2}^{n} \binom{n}{i} p^i (1-p)^{n-i}.$$

When n is between 400 and 500, we prove that $P(\xi^n_{.5}(\Delta_1) \leq 600\ \mu s)$ is an increasing function of p [17]. When $p = 0.9953$, $\sum_{i=n/2}^{n} \binom{n}{i} p^i (1-p)^{n-i} \approx 1$. Since $p \geq 0.9953$, we have $P(\xi^n_{.5}(\Delta_1) \leq 600\ \mu s) \approx 1$. The proof for setting (b) is in [17]. ■

The above theorem provides an upper bound on the median packet pair inter-arrival time at the receiver and is used in *Rule 1* of our classification scheme.

Theorem 2: **(Entropy for Ethernet connection)** In Setting (a), $H_2^{300}(I) \leq 0.49$ bit, $H_2^{900}(I) \leq 0.071$ bit. In Setting (b), when $\rho_1 = 1$ and $\rho_2 = 1$, $H_2^{300}(I) = 1.989$ bits, $H_2^{900}(I) = 0.574$ bit.

Proof: Since $I = \Delta_1$ in Setting (a), we prove the above theorem by showing that $H_2^{300}(\Delta_1) \leq 0.49$ bit, $H_2^{900}(\Delta_1) \leq 0.071$ bit. By Lemma 1,

$$P(X_1 = x_1 \mid X_0 = 0.1, \rho_1) = \frac{e^{-0.1\rho_1}(0.1\rho_1)^{x_1 - 1}}{(x_1 - 1)!}$$

Let $p(x_1 \mid 0.1, \rho_1) = P(X_1 = x_1 \mid X_0 = 0.1, \rho_1)$. We have

$$H_2(X_1 \mid \rho_1) = \frac{1}{\log 2} \sum_{x_1 = 1}^{\infty} p(x_1 \mid 0.1, \rho_1) \log(p(x_1 \mid 0.1, \rho_1))$$

In [17], we prove that $H_2(X_1 \mid \rho_1)$ is an increasing function of ρ_1, and therefore is maximized at $\rho_1 = 1$, and $H_2(X_1 \mid \rho_1 = 1) \leq 0.49$ bit. Hence, $H_2^{300}(\Delta_1) = H_2(X_1 \mid \rho_1) \leq 0.49$ bit. The results for the time scale of 900 μs are obtained by a change of variable [17]. For the proof of setting(b), see [17]. ■

In Setting (a), the upper bound of the entropy is achieved when $\rho_1 = 1$ [17]. Intuitively, this is because the high utilization at link L_1 makes it more likely for other packets to be inserted between a packet pair, which leads to more randomness and hence a higher entropy value.

In Setting (b), Theorem 2 provides the entropy values when $\rho_1 = 1$ and $\rho_2 = 1$, derived from $H(X_2)$. Numerical results indicate that $H(X_2 \mid \rho_2 = 1)$ is an increasing function of ρ_1 and hence $H(X_2 \mid \rho_2 = 1)$ obtains its maximum value when $\rho_1 = 1$ [17]. The intuition is that, for $\rho_2 = 1$, a higher value of ρ_1 can lead to higher randomness and hence higher entropy value. However, a rigorous proof of this result is left as future work. We speculate that $H(X_2)$ is an increasing function of both ρ_1 and ρ_2, which is confirmed by our simulation results [17]. Then the various entropy values in Theorem 2 are the upper bounds of $H_2(I)$. These values are used in *Rule 1* of our classification scheme. Our experimental results confirm that the entropies of all of the experiments are bounded within these obtained values (see Section V).

C. Non-Ethernet connections

We now analyze the case where the access network is not Ethernet. We first present a result on 802.11b.

Theorem 3: **(Median and entropy for 802.11b)** When using 11 Mbps 802.11b, under ideal conditions (with no contention, no retransmissions and perfect channel conditions), the median of the inter-departure times at the sender is above $800\ \mu s$ and $H_2^{300}(\Delta_0) > 1$ bit.

The above results are from the binary backoff mechanism in 802.11b (see Table I) and for the packet pair inter-departure time at the sender. The same results hold for the packet pair inter-arrival times at the receiver when all intermediate links from the sender to the receiver have utilization close to 0. Note that when contention and retransmission occur, these values will only increase.

Theorem 3, Theorem 1 and empirical observations form the foundation for *Rule 2* of our classification scheme. That is, if the median of packet-pair inter-arrival time is between $600\ \mu s$ and 2 ms, and the entropy of packet-pair inter-arrival time is at least 0.8 bit at the time scale of $300\ \mu s$, we classify the access network as WLAN. Here the median value of $600\ \mu s$ is the upper-bound for Ethernet connection; the median value of 2 ms comes from the empirical observations of cable, ADSL and dialup connections (see Section V); the entropy value of 0.8 bit comes from Theorem 3 and allows for sampling errors.

We next calculate the entropy of the packet pair inter-arrival times at the receiver in Setting (a) and (b) and show that the entropy for non-Ethernet connections can be much larger than that of Ethernet connection. Our results are for $\rho_1 = 1$ and $\rho_2 = 1$. This is because, for non-Ethernet links, the inter-arrival time of the packet pair at link L_1 and L_2 are not necessarily lower than the processing unit. Therefore, we use Lemma 2 to obtain the distribution of the inter-departure time at a queue, which requires full utilization at the queue. The entropy values for Setting (a) and (b) are listed in Table II [17]. In the table, x_0 is 3 or 4, corresponding approximately to the range of x_0 in a WLAN. For cable, ADSL and dialup, x_0 values are much larger, and the corresponding entropies are also much larger.

TABLE II
ENTROPY WHEN $\rho_1 = 1$, $\rho_2 = 1$.

x_0	Setting (a)		Setting (b)	
	$H_2^{300}(I)$	$H_2^{900}(I)$	$H_2^{300}(I)$	$H_2^{900}(I)$
(unit)	(bit)	(bit)	(bit)	(bit)
3	2.787	1.045	3.332	1.605
4	3.010	1.390	3.539	1.868

D. Validation using ns

In the above analysis, we considered links L_1 and L_2 as $M/D/1$ queues and assumed that all packets are of equal size. We now use *ns* simulation to show that these assumptions are not crucial to our analytical results.

We use setting (a) and (b) in *ns*. The sender is a wired link with the bandwidth of 100 Mbps or a wireless station connected to a base station using 802.11b. A sequence of UDP packet pairs is sent from the sender to the receiver. The interval between two packet pairs is 20 ms. We refer to each packet in the sequence of packet pairs as a probe packet. The size of the probe packet is varied to be 36, 70 or 100 bytes. HTTP, on/off UDP and TCP traffics are added to links L_1 and L_2. The HTTP traffic is generated using empirical data provided by *ns*. The on/off UDP traffic follows a Pareto distribution with a shape parameter of 1.5. The average rate during the on periods is 1 Mbps. The number of TCP flows is varied to be 0, 5, 10, 20. The number of HTTP flows is varied to be 10, 30, 50. The resultant link utilization for the various settings is from 0.4 to 1.0.

We observe that the entropy and median generally increase with the utilization and the burstiness of the traffic. When the sender is a wireless station, the entropy of packet pair inter-departure time at the sender is around 1.4 bits at the time scale of $300\ \mu s$ and the median is around 1.5 ms, conforming the results in Theorem 3. Table III lists the maximum entropy and median of packet-pair inter-arrival times in all configurations when a probe packet is 100 bytes (the corresponding entropy and median are smaller for smaller probe packets). For an Ethernet connection, the maximum values of the entropy and median are well within the bounds derived in Section IV-B. For a WLAN, the various entropies are consistent with the results in Table II. The maximum median value is around 1.6 ms and 1.8 ms in Setting (a) and (b) respectively for a WLAN and is only $80\ \mu s$ for Ethernet.

V. EXPERIMENTAL RESULTS

In this section, we describe our experiments over the Internet. This section serves two purposes. First, we use the experiments to validate the analytical results derived in Section IV. Second, we obtain some empirical results from the experiments for the classification of connection

TABLE III

VALIDATION USING *ns*: THE MAXIMUM ENTROPY AND MEDIAN VALUES IN ALL CONFIGURATIONS.

Sender	Setting (a)			Setting (b)		
	$H_2^{300}(I)$ (bit)	$H_2^{900}(I)$ (bit)	median (μs)	$H_2^{300}(I)$ (bit)	$H_2^{900}(I)$ (bit)	median (μs)
Ethernet (100 Mbps)	0.08	0.01	80	0.50	0.12	80
WLAN (11 Mbps)	1.53	0.83	1600	3.46	1.72	1800

TABLE IV

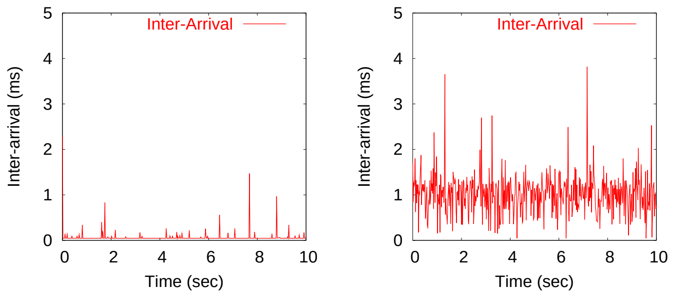
SMALL-SCALE CONTROLLED EXPERIMENTS: BOTH THE SENDER AND THE RECEIVER ARE AT UMASS.

Sender	Receiver	Sample size (pairs)	min (μs)	max (μs)	median (μs)	$H_2^{300}(I)$ (bit)	$H_2^{900}(I)$ (bit)
UMass-2	UMass-1	500	1	190	16	0.0	0.0
UMass-w1	UMass-2	500	79	8246	1099	0.8	0.3
UMass-w1	UMass-2	500	873	3959	1138	1.2	0.2
UMass-w1	UMass-1	500	693	8644	1117	1.4	0.6
UMass-w1	UMass-1	500	845	5700	1115	1.7	0.9
UMass-w1 (40 ms)	UMass-1	500	716	3428	1147	1.2	0.3
UMass-w2	UMass-1	500	193	7567	1096	1.9	1.1

TABLE V

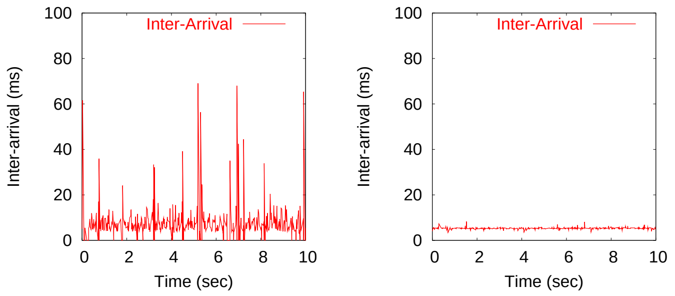
SMALL-SCALE CONTROLLED EXPERIMENTS: USC ACTS AS THE RECEIVER.

Sender	Sample size (pairs)	min (μs)	max (μs)	median (μs)	$H_2^{300}(I)$ (bit)	$H_2^{900}(I)$ (bit)
UMass-1 (Ethernet)	500	45	2293	46	0.2	0.1
UMass-2 (Ethernet)	499	44	4225	46	0.2	0.1
UPenn (Ethernet)	500	44	524	46	0.1	0.0
UMN (Ethernet)	497	45	2457	384	0.6	0.2
Brazil (Ethernet)	469	45	21036	46	0.5	0.2
Taiwan (Ethernet)	496	44	665	46	0.1	0.0
Italy (Ethernet)	498	55	8523	68	0.7	0.5
UMass-w2 (WLAN)	497	46	29341	961	2.8	1.6
UMass-w1 (WLAN)	499	46	3817	1056	2.4	1.1
UMass-w1 (WLAN, 40ms)	500	44	4380	1021	2.7	1.3
Home-1 (ADSL)	500	3440	8270	5281	2.1	1.2
Home-2 (WLAN+ADSL)	500	3343	13507	5286	2.4	1.4
Home-1 (cable)	500	43	69053	5954	4.9	3.7
Home-2 (WLAN+cable)	498	43	167707	5653	4.6	3.8



(a) (b)

Fig. 4. Inter-arrival time of the packet pairs at the receiver: (a) UMass (Ethernet) to USC. (b) UMass (WLAN) to USC.



(a) (b)

Fig. 5. Inter-arrival time of the packet pairs at the receiver: (a) Home (cable) to USC. (b) Home (ADSL) to USC.

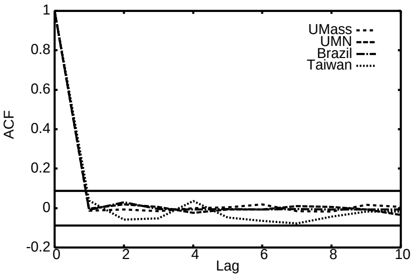


Fig. 6. Autocorrelation function of the sequence of packet pair inter-arrival times with USC as the receiver.

types. Our experiments are in two sets. The first set of experiments involves 12 machines located in 4 continents. We have accounts on all of the machines and are able to run experiments between each pair of machines. We refer to this set of experiments as *small-scale controlled experiments*. In the second set of experiments, experimenters in 10 countries ran a sender program which sent packet pairs to two machines in the University of Massachusetts (UMass). We refer to this set of experiments as *large-scale uncontrolled experiments*. We next describe these two sets of experiments in detail. At the end, we summarize the key results from both sets of experiments.

A. Small-scale controlled experiments

This set of experiments involves 12 machines, all Linux based, and with the connection of Ethernet, WLAN, cable or ADSL connections. We name the machines by their locations. Four machines are in UMass, where UMass-1 and UMass-2 are two machines with Ethernet connections and UMass-w1 and UMass-w2 are two machines with WLAN connections. The Ethernet card for UMass-1 is configured to be 10 Mbps while the Ethernet card for UMass-2 is at the default 100 Mbps. Both UMass-w1 and UMass-w2 are 802.11b compatible. The bandwidth of UMass-w1 and UMass-w2 is 11 Mbps and 22 Mbps respectively. Two machines, referred to as Home-1 and Home-2, are at a resident home in Amherst, MA, where both cable and ADSL connections are installed. Home-1 connects to the Internet through a router at the resident home. Home-2 has a wireless card which is 802.11b compatible and connects to the Internet through a wireless access point at the resident home at a bandwidth of 22 Mbps. All the other machines have Ethernet connections and are located at university sites in the east coast (UPenn), middle west (UMN), west coast (USC) of the US, Brazil, Taiwan and Italy.

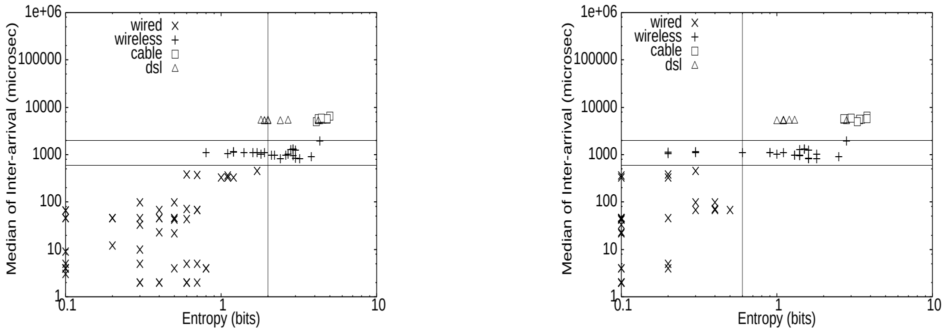
Machines with Ethernet connections act as receivers. For each receiver, any other machine can act as a sender of packet pairs. In each experiment, the sender sends a packet pair every 20 ms or 40 ms with a total of 500 packet pairs. Therefore, each experiment lasts for 10 or 20 seconds. The receiver records the arrival time of each packet using *tcpdump* [21] and calculates the inter-arrival time of the two packets in each packet pair. The Linux version at two machines (Brazil and UPenn) are too low to capture timestamps accurately. We therefore use 6 machines (UMass-1, UMass-2, USC, UMN, Taiwan and Italy) as receivers.

Let $\{I_i\}_{i=1}^n$ denote the sequence of inter-arrival times of packet pairs at the receiver. Before looking at the experimental results, we first validate that the I_i's can be regarded as independent random variables. We first plot the autocorrelation function of $\{I_i\}_{i=1}^n$ for various senders and the same receiver USC in Fig. 6. We observe that, the autocorrelation function of $\{I_i\}_{i=1}^n$ falls into the confidence interval for the various lags, indicating that the I_i's are not correlated. Thus we regard the I_i's as independent random variables.

We now consider some experimental results where both the sender and the receiver are at UMass, as listed in Table IV. In the first experiment (the first row in the table), the sender has an Ethernet connection; in all the other experiments, the sender has a WLAN connection. It is interesting to note from Table IV that, even for the sender and receiver located in the same domain, the median of inter-arrival time for a WLAN is beyond 600 μs, which easily differentiates Ethernet and WLAN connections. Furthermore, when the sender uses a WLAN connection, the entropy at the time scale of 300 μs is generally above 1 bit, except for one case (with the value of 0.8 bit). We speculate that this exception is due to sampling error.

We next consider experiments where the sender and the receiver are at geographically different locations. In particular, we consider the case that USC is the receiver. We first visually examine the inter-arrival times at the receiver from different types of connections. Fig. 4 plots the inter-arrival times at USC when the sender's connection is Ethernet or WLAN. For the Ethernet connection, the inter-arrival times are much lower and more regular than those for the WLAN connection, indicating a lower median inter-arrival time and entropy. Fig. 5 depicts the inter-arrival times at USC when the connection of the sender is cable or ADSL. For the cable connection, the inter-arrival times are scattered in a wide range of 70 ms, while for the ADSL connection, the inter-arrival times are concentrated around 5 ms. This indicates that the entropy of the cable connection tends to be larger than that of the ADSL connection.

We next consider the quantitative results shown in Table V. The sender includes all the other machines with all the possible connection types. We observe that the minimum and maximum of the inter-arrival times are not as stable as the median. The median inter-arrival time from an Ethernet connection is generally tens of microseconds (except from UMN), all within 600 μs and the entropy is no more than 0.7 bit at the timescale of 300 μs. When



(a) Time scale of 300 μs.

(b) Time scale of 900 μs.

Fig. 7. Small-scale controlled experiments: classification results.

the sender uses a WLAN, the median inter-arrival time is around 1 ms; the entropy at the timescale of 300 μs in the range of 2 to 3 bits, consistent with the results in Table II. For ADSL and cable connections, the median inter-arrival time is over 5 ms, much larger than that under Ethernet and WLAN. When the sender uses cable, the entropy is much larger than the other connections, due to the relatively low bandwidth and the randomness in the upstream channel. Note that, in Table V, the number of samples ranges from 469 to 500 packet pairs (with a loss rate from 0 to 6%), demonstrating that our classification scheme is not sensitive to the loss rate.

Fig. 7 summarizes the classification results for the small-scale controlled experiments using the time scales of 300 μs and 900 μs. In total 98 experiments are plotted in the figure. In the figure, the horizontal lines represent the median of 600 μs and 2 ms; the vertical lines correspond to the upper bounds of the entropies for Ethernet connections. We first observe that the median and the entropy of the packet pair inter-arrival times for Ethernet connections satisfy the criteria in the classification scheme (see Section III). That is, the medians are all within 600 μs (the maximum median of all the experiments is 453 μs) and the entropy is below 2.0 bits and 0.6 bit at the time scales of 300 μs and 900 μs respectively. The median inter-arrival times is between 600 μs and 2 ms for a WLAN, while it is larger than 2 ms for ADSL and cable connections. The entropies for WLAN connections are larger than 1 bit except one case (value of 0.8 bit, where the sender and the receiver are in the same domain). The cable connections exhibit large entropies, indicating a larger amount of randomness due to contention in cable networks.

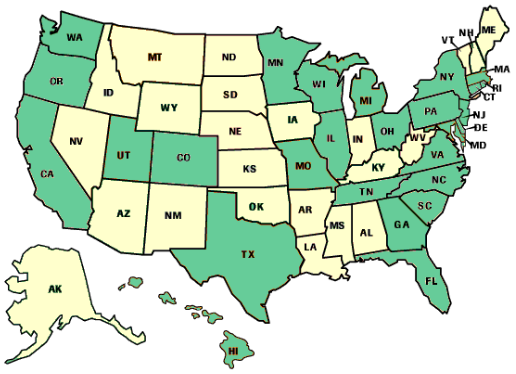


Fig. 8. Large-scale uncontrolled experiments: trace coverage in the US.

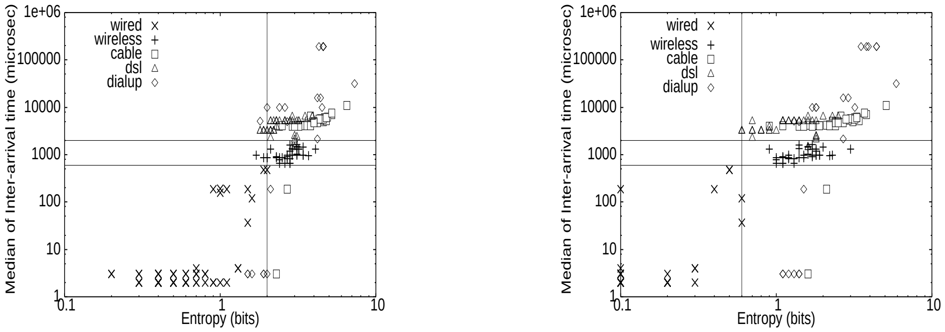
B. Large-scale uncontrolled experiments

In order to increase the scale of the experiments, we create a Windows program which sends UDP packet pairs to two receiver machines in UMass (i.e., UMass-1 and UMass-2). The program sends one packet pair every 200 ms, much coarser than the 20 or 40 ms used in Section V-A, in order to accommodate very low bandwidth dialup connections. The receivers run *tcpdump* [21] to collect the arrival time of each UDP packet and calculate the inter-arrival time of each packet pair. We distributed the sender program to friends at different locations and asked them to run it on their local machines. We collected 421 traces in total from March 9 to April 1, 2004. The senders machines are located in 28 states in the US (illustrated by the shaded regions in Fig. 8) and 9 other countries (Brazil, Canada, China, Germany, Israel, Japan, Korea, Norway, United Kingdom). The breakdown of the traces is listed in Table VI. For each receiver, we have around 70 traces from Ethernet connections; 40 from WLAN and cable connections; 50 from ADSL connections and 20 from dialup connections.

TABLE VI
LARGE-SCALE UNCONTROLLED EXPERIMENTS: BREAKDOWN OF THE TRACES.

Receiver	Ethernet	WLAN	cable	ADSL	Dialup
UMass-1	70	33	35	46	19
UMass-2	74	37	39	46	22

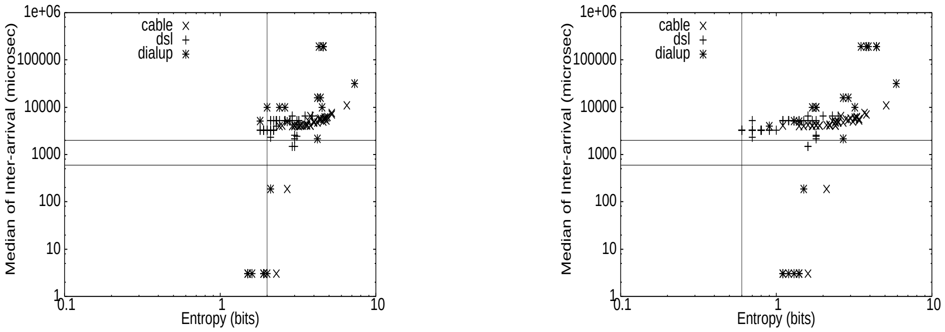
Fig. 9 plots the median and entropy of the inter-arrival



(a) Time scale of 300 μs.

(b) Time scale of 900 μs.

Fig. 9. Large-scale uncontrolled experiments: classification results for all connection types (UMass-1 as the receiver).



(a) Time scale of 300 μs.

(b) Time scale of 900 μs.

Fig. 10. Large-scale uncontrolled experiments: classification results for connection types of cable, ADSL and dialup (UMass-1 as the receiver).

times for all the experiments where UMass-1 is the receiver. The entropy is at the time scales of 300 μs and 900 μs. We observe similar results when UMass-2 acts as the receiver. This indicates that the results are not sensitive to the bandwidth of the receiver. In Fig. 9, the median and entropy of the inter-arrival times for Ethernet connections satisfy the three criteria in the classification scheme. Furthermore, no other connection type satisfies the three criteria simultaneously. Therefore, our classification method accurately differentiates between the Ethernet and non-Ethernet connections. According to Theorem 2 (see Section IV), when there is only one intermediate low-bandwidth link, the upper bound of entropy is 0.5 and 0.1 bit at the time scale of 300 μs and 900 μs respectively when using a precision of 0.1. In Fig. 9, approximately half of the Ethernet connections are within these upper bounds. For WLAN connections, the medians are in the range of 600 μs and 2 ms and the entropy at the time scale of 300 μs is larger than 1 bit, satisfying the criteria for WLAN connection in the classification scheme. All the traces from cable, ADSL and dialup connections have a median outside the range of 600 μs to 2 ms except for two traces from an ADSL connection, which are mis-classified as WLAN.

We now examine the results of cable, ADSL and dialup connections more closely. Fig. 10 plots the results for cable, ADSL and dialup connections. We observe that it is difficult to obtain a clear cut differentiation among the three low-bandwidth connections using median and entropy. The median and entropy from various traces are close for ADSL connections, while they are in a much wider range for cable and dialup connections. The majority of the traces have median above 2 ms. However, for two traces from cable connections and five traces from dialup connections, the median is as low as a few microseconds. We speculate that this is due to traffic shaping, as observed in [18]. These traces are nevertheless correctly classified as non-Ethernet connections since their entropy under the time scale of 900 μs is beyond 0.6 bit, which disqualifies them as Ethernet connections. One example of traffic shaping is shown in Fig. 11, which is from a dialup connection. We observe that although majority of the inter-arrival times are lower than 10 μs, a significant fraction of the inter-arrival times are in a wide range of 1000 to 6000 μs. The wide spread of inter-arrival times results in a large entropy value, which differentiates it from an Ethernet connection.

C. Summary

We next summarize the results from the small-scale controlled and the large-scale uncontrolled experiments. The key observations are:

- It is important to combine median and entropy together for the classification. For instance, the entropy

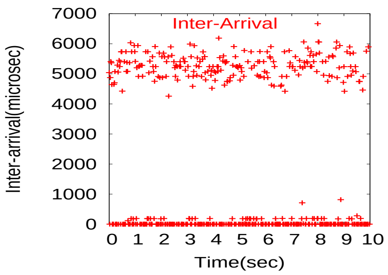


Fig. 11. An example of the packet-pair inter-arrival time of a dialup connection: the small median value might be due to traffic shaping.

of Ethernet and WLAN can overlap; the median of Ethernet and some cable and dialup connections can also overlap.

- Our classification scheme distinguishes between Ethernet and non-Ethernet connections accurately in all of the experiments (totally 509 experiments).
- Our classification scheme identifies all of the 93 traces from a WLAN connection accurately. However, 5 of the 100 traces from ADSL connections are mis-classified as WLAN connections. All of the 5 ADSL traces are from Calgary, Canada.
- Connections with a large median usually have a large entropy. For instance, no trace with a median above $600\ \mu s$ has an entropy below 0.8 bit. This might be because, when the distance between two packets in a packet pair is large, more cross traffic packets can be inserted between the two packets.

VI. Conclusion

Ethernet, wireless LAN, ADSL, cable modem and dialup are common types of connections with dramatically different characteristics. Fast and accurate classification of connection types can improve the performance of network protocol and applications dramatically. In this paper, we proposed a simple and efficient end-end scheme to classify the type of an access link using packet pairs. Our scheme utilized both the median and the entropy of the packet pair inter-arrival times, based on the intrinsic characteristics of the various connection types. Extensive experiments showed that our scheme obtains accurate classification results in a very short time (10 to 100 seconds). As future work, we are pursuing in the following directions: (i) Investigating using passive measurements (e.g., packets in TCP) to classify the connection types. (ii) Classification of cable and ADSL connections based on more specific characteristics of these two kinds of connections.

References

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

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

[3] M. Gerla, R. Bagrodia, L. Zhang, K. Tang, and L. Wang, "TCP over wireless multihop protocols: Simulation and experiments," 1999.

[4] B. S. Bakshi, P. Krishna, N. H. Vaidya, and D. K. Pradhan, "Improving performance of TCP over wireless networks," in *International Conference on Distributed Computing Systems*, 1997.

[5] R. Cohen and S. Ramanathan, "TCP for high performance in hybrid fiber coaxial broad-band access networks," *IEEE/ACM Trans. on Networking*, vol. 6, no. 1, pp. 15–29, 1998.

[6] D. G. Andersen, H. Balakrishnan, M. F. Kaashoek, and R. Morris, "Resilient overlay networks," in *Proc. 18th ACM SOSP*, October 2001.

[7] S. Banerjee, B. Bhattacharjee, and C. Kommareddy, "Scalable application layer multicast," in *Proceedings of ACM Sigcomm 2002*, August 2002.

[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] S. Biaz and N. Vaidya, "Discriminating congestion losses from wireless losses using inter-arrival times at the receiver," *IEEE Symposium ASSET'99, Richardson, TX, USA*, 1999.

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

[12] "LAN/MAN standards of the IEEE Computer Society. Wireless LAN medium access control (MAC) and physical layer (PHY) specification. IEEE standard 802.11," 1997.

[13] 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.

[14] "IEEE 802.11, 802.11a, 802.11b standards for wireless local area netowrks." http://standards.ieee.org/getieee802/802.11.html.

[15] "CableLabs, Data-over-cable service interface specifications - radio frequency interface specification, MCNS consortium, 2000, SP-RFIv1.1-106-001215."

[16] 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.

[17] W. Wei, B. Wang, C. Zhang, J. Kurose, and D. Towsley, "Classification of access network types: Ethernet, wireless LAN, ADSL, cable or dialup?," tech. rep., Department of Computer Science, University of Massachusetts, Amherst, 2004.

[18] K. Lakshminarayanan and V. N. Padmanabhan, "Some findings on the network performance of broadband hosts," in *ACM SIGCOMM Conference on Internet Measurement*, (Miami Beach, FL), 2003.

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

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

[21] "tcpdump." http://www.tcpdump.org/.